Clustering and Classification of Cases Using Learned Global Feature Weights

Eric C.C. Tsang, Simon C.K. Shiu, X.Z. Wang, and Martin Lam
Department of Computing, The Hong Kong Polytechnic University
Emails: csetsang, csckshiu, csxzwang@comp.polyu.edu.hk

Abstract
Case-base Reasoning (CBR) systems have attracted great attention in the last few years. It is a system that allows user to store, share and reuse what has been stored inside the system (i.e., the previous problem solving experiences/cases). It is similar to the way we solve unknown problems by using our experience and knowledge. When the case-base size increases, it is very difficulty to maintain the case-base, for example when similar cases have accumulated, anomalies such as redundant cases, conflicting cases, ambiguous cases, subsumed cases and unreachable cases may exist in the case-base. This is the so called case-base maintenance (CBM) problem. In this paper we propose a method to improve the performance of clustering and classification of cases in a large-scale case-base by using a learned global feature weight methodology. This methodology is based on the idea that we could use similarity measure to find several concepts (clusters) in the problem-domain such that those cases in a cluster are closely related among themselves while among different clusters those cases are farther apart. It has been demonstrated in the experiment that the performance of clustering with learned global feature weights is much better than the performance without global feature weights in terms of the retrieval efficiency and accuracy of solution provided by the system.

1. Introduction

We solve a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation. This is the core of CBR. CBR is a plausible model of human cognition. We naturally do CBR all the time, i.e., we draw on experience. CBR is a problem-solving paradigm that is fundamentally different from other major AI approaches. CBR not only relies on general knowledge of a problem domain, or making associations along generalized relationships between problem descriptors and conclusions, but also utilizes the *specific* knowledge of previously experienced, concrete problem situations (cases). A second important difference is that CBR is also an approach to incremental and sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems.

The growing use of CBR applications has increased the awareness of the importance of Case-Base Maintenance (CBM). CBM is the process of refining a CBR system's case-base to improve its performance, i.e., CBM implements policies for revising the organization or contents of the case-base in order to facilitate future reasoning for a particular set of performance objectives [2]. Nowadays many large-scale CBR systems with case-base sizes ranging from thousands to millions have been developed [4,5,6]. Large case-base size raises the problem of case retrieval, and various case deletion policies have been proposed to control the case-base growth [1,8]. Many anomalies such as redundant cases, conflicting cases, ambiguous cases, subsumed cases and unreachable cases may exist in the case-base [3,7]. Techniques that can automatically detect problematic cases in the case-base are therefore crucial to the future success of CBR technologies.

This paper presents a CBM methodology for clustering and classification of cases in case-base systems which use little or no domain-specific knowledge. It consists of two phases. The first phase is to evaluate feature importance, i.e., learning global feature weights from the case-base. This phase can recognize salient features and eliminate irrelevant features from the case-base. Its results are helpful in providing a reasonable clustering of the case-base. The second phase is to partition the case-base; the clusters identified convey different concepts within the case-base. Experimental analyses of our methodology show promising results, i.e., the performance of clustering with learned global feature weights is much better than the performance without global feature weights. This paper is organized into different sections. In section 2, global weight concept is presented and a method to learn these weights is provided. Section 3 presents a technique to partition a case-base into several clusters while in section 4 an experiment to justify our method is given. The experimental results and its analyses are mentioned in section 5. Finally a conclusion is wrapped up in section 6.

2. Global Feature Weight Concept And Its Learning Algorithm

The structure of a case in a case library is determined by problem domain and solution. The solution is considered as an action but the problem domain is usually characterized by a collection of features. Suppose that the number of features is *n*, then the global feature weight refers to a *vector* $(w_1, w_2, \ldots\ldots, w_n)$ each of its components is a real number in [0,1]. It can be interpreted that, for the entire case library, different features have different degrees of importance to the solution. It is different from a local feature weight concept. A global feature weight is assigned to each feature in a case and for the same feature in all cases they have the same global weight. A local feature weight is assigned to each feature in a case and for the same feature in all cases, they have different local weights.

2.1 Learning Global Feature Weights

Let $CL = \{e_1, e_2, \cdots, e_N\}$ be the case library. Each case in the library is identified by an index of corresponding features. We use a collection of features $\{F_j \ (j = 1, \cdots, n)\}$ to index the cases and a variable *v* to denote the action. The *i-th* case e_i in the library is represented as a *n*+1-dimensional vector, *i.e.* $e_i = (x_{i1}, x_{i2}, \cdots, x_{in}, v_i)$ where x_{ij} corresponds to the value of feature $F_j (1 \le j \le n)$ and v_i corresponds to the action $(i = 1, \cdots, N)$.

Suppose that for each $j \ (1 \le j \le n)$ a weight $w_j \left(w_j \in [0,1]\right)$ has been assigned to the *j-th* feature to indicate the degree of importance of the feature. Then, for any pair of cases e_p and e_q in the library, a weighted distance metric can be defined as

$$d_{pq}^{(w)} = d^{(w)}(e_p, e_q) = \left(\sum_{j=1}^{n} w_j^2 (x_{pj} - x_{qj})^2 \right)^{1/2} \qquad (1)$$

When all the weights are equal to 1 the distance metric defined above degenerates to Euclidean measure, denoted by $d_{pq}^{(1)}$, in short, denoted by d_{pq}.

$$d_{pq}^{(1)} = d^{(1)}(e_p, e_q) = \left(\sum_{j=1}^{n} (x_{pj} - x_{qj})^2 \right)^{1/2} \qquad (2)$$

By using the weighted distance, a similarity measure between two cases, $SM_{pq}^{(w)}$, can be defined as follows:

$$SM_{pq}^{(w)} = \frac{1}{1 + \alpha \cdot d_{pq}^{(w)}} \qquad (3)$$

where α is a positive parameter. When all weights take value 1 the similarity measure is denoted by $SM_{pq}^{(1)}$.

$$SM_{pq}^{(1)} = \frac{1}{1 + \alpha \cdot d_{pq}^{(1)}} \qquad (4)$$

In this section, a feature evaluation function (in which the feature weights are regarded as the variables) is defined. The smaller the evaluation value, the better the corresponding features. Thus we would like to find the weights such that the evaluation function attains its minimum. The task of minimizing the evaluation function with respect to weights is performed using a gradient-decent technique.

We formulate this optimization problem as follows. For a given collection of feature weights $w_j \left(w_j \in [0,1],\ j = 1, \cdots n\right)$ and a pair of cases e_p and e_q, equation (1) defines a weighted distance measure $d_{pq}^{(w)}$ and equation (3) defines a similarity measure $SM_{pq}^{(w)}$. When all weights take value 1, $d_{pq}^{(w)}$ and $SM_{pq}^{(w)}$ will degenerate to the Euclidean distance $d_{pq}^{(1)}$ and $SM_{pq}^{(1)}$ respectively.

A feature evaluation index function *E* is defined as:

$$E(w) = \sum_{p} \sum_{q(q \ne p)} \left(SM_{pq}^{(w)} (1 - SM_{pq}^{(1)}) + SM_{pq}^{(1)} (1 - SM_{pq}^{(w)}) \right) \qquad (5)$$

One may notice that the feature evaluation index function *E*(*w*) will gradually become zero when $SM_{pq}^{(w)} \to 0$ or 1, we hope to find a collection of weights such that the feature evaluation function attains its minimum. To minimize equation (5), we use a gradient-decent technique. The change in w_j (denoted by Δw_j) is computed as

$$\Delta w_j = -\eta \frac{\partial E}{\partial w_j}, \qquad (6)$$

for $j = 1, \cdots, n$, where η is the learning rate.

For the computation of $\frac{\partial E}{\partial w_j}$, the following expressions are used:

$$\frac{\partial E(w)}{\partial w_j} = \sum_{p} \sum_{q(q<p)} \left(1 - 2 \cdot SM_{pq}^{(w)}\right) \cdot \frac{\partial SM_{pq}^{(w)}}{\partial d_{pq}^{(w)}} \cdot \frac{\partial d_{pq}^{(w)}}{\partial w_j} \qquad (7)$$

$$\frac{\partial SM_{pq}^{(w)}}{\partial d_{pq}^{(w)}} = \frac{-\alpha}{(1+\alpha \cdot d_{pq}^{(w)})^2} \tag{8}$$

$$\frac{\partial d_{pq}^{(w)}}{\partial w_j} = w_j \chi_j^2 \Bigg/ \left(\sum_{j=1}^{n} w_j^2 \chi_j^2 \right)^{1/2} \tag{9}$$

where $\chi_j^2 = (x_{pj} - x_{qj})^2$.

The Global Feature Weights Training Algorithm
To obtain the optimal global feature weights, we need to minimize *E* and the training algorithm is described as follows.

Step 1. Select the parameter α and the learning rate η.

Step 2. Initialize w_j with random values in [0, 1].

Step 3. Compute Δw_j for each j using equation (6).

Step 4. Update w_j with $w_j + \Delta w_j$ for each j.

Step 5. Repeat step 3 and step 4 until convergence, i.e., until the value of *E* becomes less than or equal to a given threshold, or until the number of iterations exceeds a certain predefined number.

After training, the function $E(w)$ attains a local minimum. We expect that, in average, the similarity values $\{SM_{pq}^{(w)}, p, q = 1, \cdots N, q < p\}$ with trained weights are closer to 0 or 1 than that without trained weights such as $\{SM_{pq}^{(1)}, p, q = 1, \cdots N, q < p\}$.

3. Partition Case-Base Into Several Clusters

In this section we attempts to partition the case library into several clusters by using the weighted distance metric with the weights learned in phase 1.

Since the considered features are considered to be real-valued, many methods such as C-Mean clustering [9] and Kohonen's self-organized network [11] can be used to partition the case library. However, this paper adopts a typical approach of clustering, i.e., similarity matrix [10] which uses only the information of similarity between cases. This approach first transforms the similarity matrix into an equivalent matrix and then considers the cases being equivalent to each other as one cluster. The procedure is as follows:

Give a significant level (threshold) $\beta \in [0,1]$

Step 1. Determine the similarity matrix $SM = (SM_{pq}^{(w)})$ according to equations (1) and (3).

Step 2. Compute $SM1 = SM \circ SM = (s_{pq})$ where $s_{pq} = \max_k (\min(SM_{pk}^{(w)}, SM_{kq}^{(w)}))$.

Step 3. If $SM1 \subset SM$ then go to Step 5, else replace *SM* with *SM*1 and go to Step 3.

Step 4. Determine several clusters based on the rule "case p and case q belong to the same cluster if and only if $s_{pq} \geq \beta$ "

The clusters identified are considered to have different concepts within the case-base, and each concept identifies a subset of the problem domain that differs characteristically from the rest of the problem domain. Since the clustering result of phase 2 is crisp (not fuzzy), we require the clustering with *m* clusters $\{L_1, L_2, \cdots, L_m\}$ to satisfy the following property:

$$Min_{q \in L_i} SM_{pq}^{(w)} > Max_{1 \leq j \leq m, j \neq i} (Max_{q \in L_j} SM_{pq}^{(w)}) \tag{10}$$

for any $i (1 \leq i \leq m)$ and any case $p \in L_i$.

Equation (10) describes such a situation that the similarity between a case and the cluster to which the case belongs is bigger than the similarity between the case and any other cluster.

3.1 Clustering Evaluation

It is worth noting that the result of clustering depends strongly on the feature weights that are used in computing the similarity between two cases. We evaluate the performance of clustering by the following three indexes, namely, the intra-similarity, the inter-similarity and the number of clusters. These indexes measure the quality of a partition by ensuring that it will not favor large number of clusters. They are based on the idea that for a partition to be a good cluster it has small intra-cluster distances and large inter-cluster distances. Therefore they can be used as a criterion to optimize the clustering result.

Intra-similarity
For a given cluster L, the intra-similarity of L is defined as

$$SM_L^{(w)} = \frac{2}{r(r-1)} \sum_{p,q \in L(p<q)} SM_{pq}^{(w)} \tag{11}$$

where *r* is the number of cases in the cluster *L*. For a clustering with *m* clusters $\{L_1, L_2, \cdots, L_m\}$, the intra-similarity is defined as the average of all its cluster

intra-similarities, i.e.,

$$SM_{\text{int}ra}^{(w)} = \frac{1}{m}\sum_{j=1}^{m} SM_{L_j}^{(w)} \quad (12)$$

It is clear that the value of $SM_{\text{int}ra}^{(w)}$ is in [0, 1]. The bigger the value of $SM_{\text{int}ra}^{(w)}$, the better the performance of the clustering.

Inter-similarity

For a pair of clusters L_1 and L_2, the inter-similarity is defined as

$$SM_{L_1,L_2}^{(w)} = \frac{1}{r_1 \cdot r_2}\sum_{p\in L_1, q\in L_2} SM_{pq}^{(w)} \quad (13)$$

where r_1 and r_2 are numbers of cases in L_1 and L_2 respectively. For a clustering with m clusters $\{L_1, L_2, \cdots, L_m\}$, the inter-similarity is defined as the average of all pairs of inter-similarities, i.e.,

$$SM_{\text{int}er}^{(w)} = \frac{2}{m(m-1)}\sum_{1\le i<j\le m} SM_{L_i,L_j}^{(w)} \quad (14)$$

Obviously, the value of $SM_{\text{int}er}^{(w)}$ is in [0, 1]. The smaller the value of $SM_{\text{int}er}^{(w)}$, the better the performance of the clustering.

Number of clusters

To have an acceptable accuracy, i.e., the intra-similarity of clustering is bigger than or equal to a threshold (>0.5) and the inter-similarity of clustering is smaller than or equal to a threshold (<0.5), one can interpret that the smaller the number of clusters, the better the performance of the clustering.

4. An Experiment

Boston Housing Data is used as the testing data set in this experiment. This data set is taken from the StatLib library that is maintained at Carnegie Mellon University. Boston Housing Data concerns about the housing values in the suburbs of Boston; it contains 506 records and 14 attributes, including 13 continuous attributes and one binary-valued attribute. Attribute information is shown in Table 1.

In this experiment, the raw data testing file was converted into a case-base that our algorithm can handle by converting all rows into cases and all columns into features. To test the scale up ability of the methodology, we used Boston Housing Data as the testing case-base. There is a recursive loop in the experiment for recording the results for each significance level (β) values beginning from 0.65. We investigate the influence of β on the quality of resulting partitions. This experiment is designed to generate the global feature weights of the case-base, to partition the case-base into several clusters and to find out the three clusters evaluation indexes against the case-base with learned global feature weights and without it.

Attribute Name	**Meaning**
CRIM	Per capita crime rate by town
ZN	Proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	Proportion of non-retail business acres per town
CHAS	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX	Nitric oxides concentration (parts per 10 million)
RM	Average number of rooms per dwelling
AGE	Proportion of owner-occupied units built prior to 1940
DIS	Weighted distances to five Boston employment centres
RAD	Index of accessibility to radial highways
TAX	Full-value property-tax rate per $10,000
PTRATIO	Pupil-teacher ratio by town
B	$1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in $1000's

Table 1: Boston Housing Data Attribute Information.

Feature Name	Feature
CRIM	0.668691
ZN	0.438765
INDUS	0.577563
CHAS	0
NOX	0.992728
RM	0
AGE	0.766656
DIS	0.607898
RAD	0.887387
TAX	0
PTRATIO	0.616901
B	0.356120
LSTAT	0
MEDV	0

Table 2: Boston Housing Data Feature Weight Information.

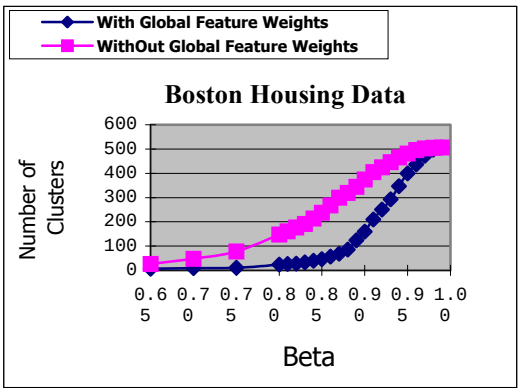

Fig. 1: Experimental result of number of classes

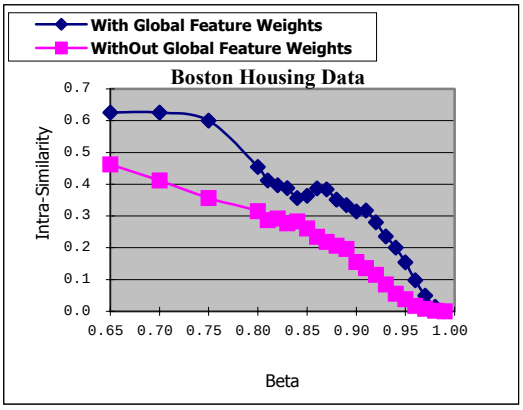

Fig. 2: Experimental result of Intra-Similarity

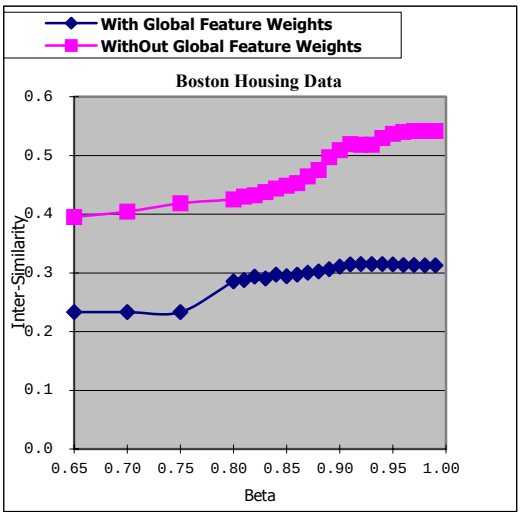

Fig. 3: Experimental result of Inter-Similarity

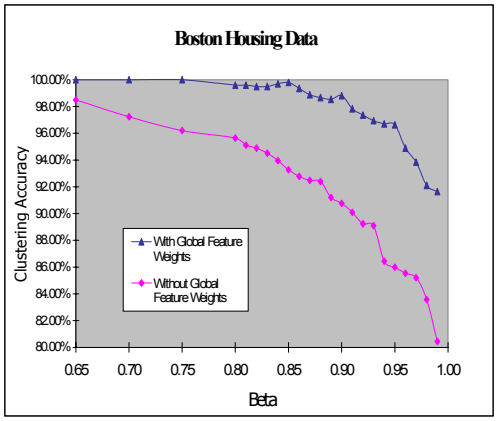

Fig. 4: Experimental result of clustering accuracy

5. Experimental Results And Its Analyses

After applying the learning feature weights algorithm mentioned in section 2, the feature weights results shown in Table 2 were obtained (training time: 568minutes). Those features with zero feature weight are taken as irrelevant features and can be deleted from the target case-base. The experimental parameter settings are: alpha (α) = 0.015, learning Rate (η) = 0.150, threshold = 0.015.

Figure 1 reveals two important observations. Firstly, the curve for the testing case-base with global feature weights is *uniformly better* (i.e. smaller number of clusters) than those without it. It is expected since the global feature weights do take into consideration the degree of importance of the feature weights of the attributes in the case-base. The algorithm eliminates those irrelevant features if its learned feature weights approach zero.

Secondly, as the value of β increases, the number of clusters produced increases. It implies that the cluster size drop as β increases. Initially, when the value of β is 0.65, the number of clusters is pretty small. After β increases to over 0.8, the number of clusters approaches to the maximum which is equal to the number of cases in the case-base. This phenomenon explains the fact that the variation of β is closely related to the number of clusters.

A large intra-similarity implies a good quality of the clusters. Therefore, we focus on finding the maximum value of intra-similarity. One may observe from Figure 2 that the curve for the testing case-base with global feature weights is ***uniformly better*** (i.e. higher) than those without it. This result shows that our methodology does improve the quality of the clusters.

Moreover, there is a decreasing trend of the intra-similarity as β increases. This phenomenon can be

explained by the fact that when we partition the case-base into larger number of clusters, the size of the clusters becomes smaller and hence smaller intra-similarity. The intra-similarity attains minimum value when β is 1. Consequently, when only intra-similarity is considered, the optimal β that maximizes the intra-similarity should be less than or equal to 0.75.

Contrary to intra-similarity, we look for β that minimizes inter-similarity. Inter-similarity shows the measurement of the cluster density between clusters; a smaller value corresponds to good quality of the clusters. One may observe from Figure 3 that there is an increasing trend of the inter-similarity as β increases. This phenomenon is due to the result of "*Number of clusters*" showing that the number of clusters is small when β become small and vice versa. Therefore, this implies that the inter-similarity attains maximum value when β is 1.

Figure 4 also shows that as the value of β increases, the clustering accuracy decreases gradually. Initially, when the value of β is less than or equal to 0.75, the graph attains maximum clustering accuracy. After β increases to over 0.8, the clustering accuracy drops gradually. This phenomenon can be explained by the fact that the clustering accuracy is closely related to the number of clusters.

6. Conclusion

In this paper a systematic methodology for clustering and classification of cases from scratch with minimal user intervention has been developed and implemented. The proposed methodology is efficient and effective in dealing with large case-bases and is necessary for maintaining a case-base. The main idea of this methodology is to partition case-bases into clusters where the cases within the cluster are more similar than cases in other clusters. A method to learn global feature weights is used for weighting features and selecting salient features. The global feature weight information can help eliminate irrelevant features from the case-base, thereby improving retrieval efficiency and helping to get reasonable clusters. The learned global feature weights with Euclidean metric are then used for discovering the clusters (concepts) of the case-base.

After applying the proposed methodology, we obtained a new partitioned case-base in which each cluster contains cases that are closely related to each other, while between different clusters the cases are farther apart from each other. The results obtained are encouraging: the dimension of the *Boston Housing Data* was reduced from 14 to 9, and when using global feature weights approach, the performance of clustering is much better than without the learned global feature weights.

Acknowledgement

This work is supported by the Hong Kong Polytechnic University Central Research Grant A-PB88.

7. References

[1] B. Smyth, and M.T. Keane, "Remembering to Forget: A Competence-Preserving Case Deletion Policy for Case-based Reasoning systems," *Proceedings of the fourteenth international Joint Conference on Artificial Intelligence*, 1995, pp 377-382.

[2] D.B. Leake, and D.C. Wilson, "Categorizing Case-Base Maintenance: Dimensions and Directions, Advances in Case-Based Reasoning," *4th European Workshop, EWCBR-98* Dublin, Ireland, 1998, pp 196-207.

[3] O'Leary, D.E., "Verification and Validation of Case-Based Systems," *Expert Systems With Applications*, Vol. 6, 1993, pp 57-66.

[4] H. Kitano, and H. Shimazu, "The Experience Sharing Architecture: A Case Study in Corporate-wide Case-based Software Quality Control," In D. Leake, ed., *Case-Based Reasoning: Experiences, Lessons, and Future Directions,* Menlo Park, CA:AAAI Press, 1996, pp 235-268.

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

[6] J. Deangdej, D. Lukose, E. Tshui, P. Beinat, and L. Prophet, "Dynamically Creating Indices for Two Million Cases: A Real World Problem," In I. Smith, and B. Faltings, eds., *Advances in Case-Based Reasoning*, Springer Verlag, 1996, pp 105-119.

[7] Simon C.K. Shiu, *Formal Description Techniques for the Verification of Expert Systems*, PhD Thesis, Department of Computing, HK Polytechnic University, 1997.

[8] S.S. Anand, D. Patterson, J.G. Hughes, and D.A. Bell, "Discovering Case Knowledge using Data Mining," *Second Pacific Asia Conference, PAKDD-98*, Australia, 1998, pp 25-35.

[9] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, NewYork, 1981.

[10] G. Fu, "An algorithm for Computing the Transitive Closure of a Similarity Matrix," *Fuzzy Sets and Systems*, vol. 51, 1992, pp 189-194.

[11] T. Kohonen, *Self-Organization and Associate Memory,* Springer, Berlin, 1988.