A Data-Driven Approach for Determining Weights in Global Similarity Functions

Amar Jaiswal[0000−0002−5992−7439] and Kerstin Bach[0000−0002−4256−7676]

Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
https://www.ntnu.edu/idi

Abstract. This paper presents a method to discover initial global similarity weights while developing a case-based reasoning (CBR) system. The approach is based on multiple feature relevance scoring methods and the relevance of features within each scoring method. The objective of this work is to utilize the characteristics of a dataset when creating similarity measures. The primary advantage of this method lies in its data-driven approach in the absence of domain knowledge in the early phase of a CBR system development. The results obtained based on the experiments on multiple public datasets show that the method improves the performance of similarity measures for a CBR system in discriminating relevant similar cases. Evaluation of the results is based on the method suitable for unbalanced datasets.

Keywords: Global Similarity Weights · Feature Weights · CBR · Case-Based Reasoning

1 Introduction

Case-based reasoning [1] (CBR) is a problem solving methodology based on past experiences. It is based on the assumption that similar problems have similar solutions. With this assumption, a CBR system is designed to retrieve similar cases for a new problem. The solution of the retrieved cases are used to solve the new problem. Hence, it becomes a key to retrieve the correct and relevant cases. This paper proposes a data-driven approach to address this issue in the early phases of a CBR system development, where the domain knowledge might not be initially available.

When developing CBR systems today, we often have access to datasets containing experiences. Those experiences are often structured for various purposes and not necessarily all information are relevant to represent a case. The relevant attributes can often be determined in collaboration with experts or using data driven approaches, while the definition of initial similarity measures are more challenging. This task has been addressed by researchers before, and learning or deriving similarity measures is an active field in CBR research [11,27].

In this paper we will investigate whether we can derive global similarity measures from a given dataset. In paper [22] similarity measures have been learned using feedback and similarity teacher. Local similarity measures as well as the learning of comprehend similarity measures have been obtained using Artificial Neural Networks is presented in [12,2].

However, deriving the weights for global similarity measures from a given dataset is a novel approach and has the potential to improve building initial CBR systems. In this paper, we will address how those similarity measures can be automatically defined and we show how the proposed methods works on open datasets.

The hypothesis of the paper is using an ensemble of feature relevance scoring methods to discover initial feature weights for a CBR system. This can be used in early phases of a CBR system development, where the researcher has little or no guidance for the domain knowledge.

The paper is organised as follows: section 2 discusses the related work about finding feature weights, section 3 presents the core of the paper, how to discover feature weights using a data-driven approach. Section 4 provides the details of the experiments' setup, datasets used, and evaluation process. Section 5 presents the experimental results. Section 6 is dedicated to the interpretation of the results and its relevance to our hypothesis. The last section concludes the paper and projects the future work.

2 Related Work

Extracting feature weights is a well known research problem area since multiple decades [4]. Multiple methods and references are mentioned in the paper that are used in feature weight extraction. In another paper [5] Aha and Goldstone demonstrate that the feature weights in the similarity setting are context dependent, with the help of 40 human subjects in their experiment. Thus, a universal algorithm for feature weight extraction might not be possible in this context.

The work in [8] describes the challenges involved with the symbolic features to be used for k-NN, and claims that the weighted k-NN is advantageous in simplicity, training speed, and perspicuity. The paper [21] is focused on learning a non-symmetric local similarity metrics, which is based on the learning approach.

Stahl and Gabel [23] discusses the challenges involved in developing a CBR system and points out that the required knowledge, many a times, is unavailable during the developmental phase. He also describes optimising the similarity measures with the help of a *similarity teacher*, which might not be available in the initial phases of the development.

Cost and Salzberg [8] discusses the importance of k-NN for classification tasks, where features have symbolic values. It also presents the experimental results based on three techniques PEBLS, back propagation, and ID3 for the comparison. Novakovic et al. [17] compare six feature ranking methods and their experimental results, which shows that different ranking methods assigns different ranks to the features. This supports our hypothesis that using ensemble of multiple methods could provide improvement in discovering the correct feature weights.

Prati's paper [19] investigates and proposes a general framework for ensemble feature ranking based on different ranking aggregation methods. It suggests that the ensemble feature ranking improves the quality of feature ranks. It also elucidates on the merits and demerits of using score aggregation versus rank aggregation for discovering the feature weights.

Multiple papers have presented the evaluation of feature weights based on mean absolute error, mean squared error, or accuracy [25,21,23]. However, when datasets are unbalanced, these evaluation methods might not be suitable, due to the well known issues of class-imbalance and accuracy paradox [26]. Thus, for the evaluation of the results, we have used the F1-scores and 10-fold cross-validation, based on the confusion matrix.

3 Relevance-based feature weights

This section presents our approach for discovering global similarity weights for a given classification dataset. It is primarily based on the scores from multiple feature relevance scoring methods.

The global similarity function is the weighted sum of all the local similarity scores. The global similarity function used in this paper is shown in equation 1, where w_i is the weight of the feature i. The $sim(Q, C)$ describes the global similarity function between a query Q and a case C. Further, for each attribute i a local similarity function is defined as $sim_i(q, c)$, where q is the attribute value from the query and c is the respective attribute value from the case. The result of this global similarity function is a similarity score in the range [0,1]. The paper is focused on data-driven approach to discover the value of w_i for the feature i.

$$\mathbf{sim}(\mathbf{Q},\mathbf{C}) = \frac{1}{\sum w_i} . \sum_{i=1}^{n} w_i . sim_i(\mathbf{q},\mathbf{c}) \tag{1}$$

3.1 Proposed Method

We will refer to "feature relevance scoring methods" as "**scoring methods**", and "feature relevance scores" as "**scores**" going forward.

The entire method is described as a flowchart shown in figure 1. The process of discovering feature weights starts by selecting a classification dataset, a set of scoring methods, and percentage of features to be used. The percentage of features, *percent*, defines the proportion of features that are considered in the feature weight computation. Thus, a $percent = 100$ refers to all the features, while a $percent = 25$ refers to 25% of features with highest ranks, by each scoring method.

The scores are computed for all the features over each scoring method. However, only the *percent* of features with highest scores in each scoring method are considered. It is represented as $max\_number\_of\_features$ shown in equation 2.

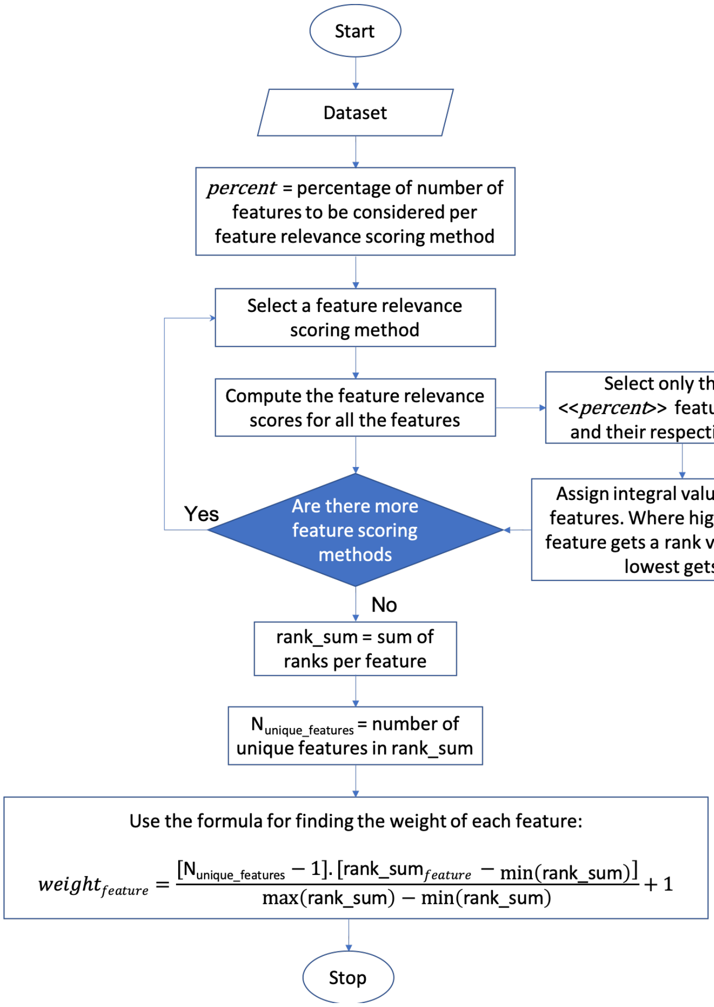
Start
Dataset
percent = percentage of number of features to be considered per feature relevance scoring method
Select a feature relevance scoring method
Compute the feature relevance scores for all the features
Select only the top <<percent>> features i.e. N_f, and their respective scores
Assign integral values to these features. Where highest scoring feature gets a rank value N_f, and lowest gets 1.
Are there more feature scoring methods
Yes
No
rank_sum = sum of ranks per feature
$N_{unique\_features}$ = number of unique features in rank_sum
Use the formula for finding the weight of each feature:
$weight_{feature} = \frac{[N_{unique\_features} - 1].[rank\_sum_{feature} - min(rank\_sum)]}{max(rank\_sum) - min(rank\_sum)} + 1$
Stop

Fig. 1: Flow chart of the proposed method for discovering the subset features and their weights for a given dataset and featuring scoring methods.

The top $max\_number\_of\_features$ are sorted in descending order with respect to their scores.

The following procedure is executed for each scoring method. A rank, as per equation 2, is assigned to each feature. A feature with a highest score receives the highest rank which is equal to the value of $max\_number\_of\_features$. The rank of the lowest scoring feature is assigned to 1. Additionally, in case of a collision with equal scores, the rank of the previous feature (in descending order) is assigned to the colliding feature. While the succeeding non-colliding feature receives a rank with respect to its position in descending order.

$$rank = \{x \in \mathbb{N} \mid 1 \leq x \leq max\_number\_of\_features\} \tag{2}$$

Once the ranks are computed for all the scoring methods, they are summed up with respect to each feature and stored in a $rank_{sum}$ vector. The size of $rank_{sum}$ vector is assigned to the variable N, in equation 3.

Finally, the computation of the feature weight is performed as shown in equation 3. The $Weight(\mathbf{f})$ represents the weight of the feature f. These feature weights are used as global similarity weights for the respective CBR system.

$$Weight(\mathbf{f}) = \left[\Big(N - 1\Big) \cdot \Big(\frac{rank_{sum}(\mathbf{f}) - min(\mathbf{rank_{sum}})}{max(\mathbf{rank_{sum}}) - min(\mathbf{rank_{sum}})} \Big) \right] + 1 \tag{3}$$

3.2 Relevance-based feature weighting algorithm

In this section we present the relevance-based feature weighting algorithm. The algorithm 1 requires three parameters: the target classification dataset, a list of scoring methods, and the percentage of features to be considered. These parameters are defined as arguments for the function $computeFeatureWeights$. This function returns a map of feature weights where the feature names are the keys.

The ranks for the features are computed for every scoring method, and are stored in the variable $feature_{rank}$. The ranks are assigned in the descending order per scoring method, thus the most relevant feature gets the highest rank. If multiple features possess the same score then all of them are assigned with the same rank whereas the subsequent feature gets a rank with respect to its position in the descending order. The algorithm computes the sum of all the ranks with respect to each feature and stores it in the variable $rank_{sum}$.

In the last step of the algorithm, the value for N is the number of unique features in the $rank_{sum}$. And, the $min()$ and $max()$ functions provide the minimum and maximum values. Once the feature weights are successfully computed, they are used as feature weights in modeling the global similarity function.

The size of the final feature list and hence the attributes that receive a global weight depends on multiple factors, such as:

- the value of the *percent* variable,
- the relevance of a feature for the classification,
- the number of scoring methods used,
- the scores from various scoring methods

One of the inherent properties of this algorithm is that it also performs feature selection, which could be influenced in multiple ways. Two of the primary ways are by changing the value of *percent* or by varying the number of scoring methods to be used in the algorithm.

Algorithm 1: Relevance-based feature weighting algorithm

```
Input: Dataset
Input: $methods \leftarrow feature\ relevance\ scoring\ methods$
Input: $percent \leftarrow percentage\ of\ features\ to\ be\ considered\ per\ method$
Output: $Weights$
$Weights \leftarrow computeFeatureWeights(Dataset, methods, percent)$
Function computeFeatureWeights($Dataset, methods, percent$):
    $feature_{rank} \leftarrow getRanks(Dataset, methods, percent)$
    $rank_{sum} \leftarrow sum(feature_{rank})$                // per feature
    $Weights \leftarrow 0$
    $N \leftarrow size(rank_{sum})$                // $N$ scaling factor
    for $feature$ do
        $Weights(feature) \leftarrow [N-1] \cdot [\frac{rank_{sum}(feature) - \mathbf{min}(rank_{sum})}{\mathbf{max}(rank_{sum}) - \mathbf{min}(rank_{sum})}] + 1$
    end
return $Weights$
```

4 Experiments

In this section we present a set of experiments where we used openly available datasets to evaluate our method. The criteria for the considered datasets were that they fit a classification task, have different numbers of features as well as a variation of cases vs. the number of features.

The experimental setup uses myCBR tool[24] including its workbench and REST API module. The myCBR tool is used for modeling similarity, generating ephemeral case bases, and performing retrievals. The evaluation of the experimental results are based on 10-fold cross-validation.

The following subsections briefly describe datasets, feature relevance scoring methods, and confusion matrices used in our experiments.

4.1 Datasets

Table 1 lists four public datasets used in our experiments. They are available on "UCI Machine Learning Repository"[1]. The chosen datasets are for multivariate classification tasks and consist of features of type categorical, numerical, or a combination of both. One can see that the number of cases and target classes vary, and therewith pose different challenges for a CBR classifier.

4.2 Feature Relevance Scoring Methods

To get the feature relevance scores we used Orange [9], an open source tool. With the help of Rank widget, Orange version 3.20.1, the scores from the six scoring methods are obtained at the default settings for each dataset. As described in section 3, one can use multiple scoring methods, for the experiments we used the

[1] https://archive.ics.uci.edu/ml/index.php

Table 1: Description of the datasets used in the experiment

Sln.	Dataset	Task Type	Data Types	Samples	Features	Missing Values	Target Classes
1	Car Evaluation [6]	Multivariate Classification	Categorical	1728	6	0	4
2	Pima Indians Diabetes [28]	Multivariate Classification	Float, Integer	768	8	0	2
3	Tic-Tac-Toe Endgame [3]	Multivariate Classification	Categorical	958	9	0	2
4	Zoo [10]	Multivariate Classification	Categorical, Integer	101	17	0	7

default six scoring methods from the tool. A brief description of these scoring methods are as follows:

- $Information\ Gain$ [16]: measures the gain in information entropy by using a feature with respect to the class.
- *Gain Ratio* [20]: a ratio of the information gain and the attribute's intrinsic information, which reduces the bias towards multi-valued features that occurs in the information gain.
- *Gini* [7]: is a measure commonly used in decision trees to decide what is the best attribute to split the current node for an efficient decision tree construction. It is a measure of statistical dispersion and can be interpreted as a measure of impurity for a feature or the inequality among values of a frequency distribution.
- $Chi2$ [18]: this method evaluates each feature individually by measuring the chi-squared statistic with respect to the class.
- $ReliefF$ [15]: this method uses the ability of an attribute to distinguish between classes on similar data instances.
- $FCBF$ [29]: (Fast Correlation Based Filter) entropy-based measure, which also identifies redundancy due to pairwise correlations between features.

4.3 Confusion Matrix

The results of the retrievals are represented as a confusion matrix (CM). For instance, a retrieval result for 26 classes of a dataset can be represented using a CM as shown in figure 4, where $A, B, ..., and\ Z$ are the class labels. An element of this matrix, Φ (a positive integer value $(\mathbb{Z}^{\geq 0})$), is the number of times a query resulted in a class pair. A class pair represents the location of an element in a CM, and is represented by lower subscripts of Φ as Φ_{tp}. Where, t represents the true class and p represents the predicted class. Additionally, the t and p represents the rows and columns of the CM, respectively.

$$CM = \begin{bmatrix} \Phi_{AA} \ \Phi_{AB} \ ... \ \Phi_{AZ} \\ \Phi_{BA} \ \Phi_{BB} \ ... \ \Phi_{BZ} \\ . \quad . \quad . \quad . \\ \Phi_{ZA} \ \Phi_{ZB} \ ... \ \Phi_{ZZ} \end{bmatrix} \tag{4}$$

4.4 Confusion Matrix for k-Fold Cross-Validation

A CM is constructed with respect to each dataset before the evaluation process begins and is initialized to 0 (element-wise).

When a query is executed, the CM_{init} is updated as shown in equation 5 with respect to the true and predicted class labels.

$$CM_{query} = CM_{init}[true\_class][predicted\_class] + 1 \tag{5}$$

Thereafter, according to the equation 6, the confusion matrix CM_k for k^{th} iteration of the k-fold cross-validation is computed. In this equation, the variable M represents the total number of query cases in the k^{th} iteration.

$$CM_k = \sum_{m=1}^{M} CM_{query_m} \tag{6}$$

Finally, the confusion matrix for the entire k-fold cross-validation is computed as shown in equation 7. Thus, at the end of all k iterations the CM_{k-fold} contains all predictions with respect to the entire case base.

$$CM_{k-fold} = \sum_{i=1}^{k} CM_k \tag{7}$$

The experiments performed in this paper are with *percent* values equal to 50, 75, and 100.

4.5 Evaluation

For the process of evaluation, we create a case base for each dataset where all the features are included. The local similarity measures are modeled using the interquartile ranges for a numerical feature (see [27] for details), and pair-wise similarity for a categorical feature. As a baseline system each dataset has been provided to a CBR engineer to model the global and local similarities manually. Additionally, a equal weighted global similarity function is implemented for each these datasets. The basis for the evaluation of the experimental results is a confusion matrix generated from 10-fold cross-validation as per equation 7 and the F1-scores.

The datasets selected for the experiments are unbalanced, thus we use F1-score as an evaluation measure. The F1-scores are computed for 10 runs over the 10-fold cross-validation confusion matrices. The computation of F1-scores are based on the equation 8.

$$F1\_score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{8}$$

5 Results

This section presents the results obtained from the experiments described in the previous sections.

The naming convention used for representing a global similarity function is $<$ **name** $>$_$<$ **percentage** $>$. Where the $<$ **name** $>$ describes the type of similarity function, explained as below:

- **manual_***: global similarity function with manually modeled feature weights, based on domain knowledge.
- **eq_***: global similarity function with equal feature weights.
- **rank_***: global similarity function with discovered feature weights, which uses sum of ranks for weight computation.
- **score_***: global similarity function with discovered feature weights, which uses sum of scores for weight computation.
- **info_gain_*, gain_ratio_*, gini_*, chi_sq_*, relief_f_*, and fcbf_***: global similarity functions for the individual scoring methods, described in section 4.2.

The $<$ **percentage** $>$ or ***** is a place holder for percentage of features selected, per scoring method, that was used for weight computation. The percentages considered for this paper are **50%**, **75%**, and **100%** (**all**). All the features with respect to various global similarity functions are same for a given percentage value.

Figure 2 presents the confusion matrices for of the Zoo dataset. Each confusion-matrix is obtained based on 10-fold cross-validation for a global similarity function. The title of each matrix describes the name of the dataset and the global similarity function used for the retrieval. Likewise, the y-axis represents the true class labels (label of the query case), and the x-axis represents the predicted class labels (label of the retrieved case). The Zoo dataset poses the most challenging classification task since the classifier needs to distinguish 7 different classes while only having 101 cases available, which leads to very low support cases during the evaluation. However, the general trend shows that the more features are included, the better the classifier is performing. This can be seen in Figure 2 where there are less misclassifications in the first row (all features included in the global similarity function), compared to the second row (75% of features included) and third row (50% of features included).

Figure 3 presents the F1-score distributions for the 10-fold cross-validation, where none of the individual scoring methods perform consistently across all

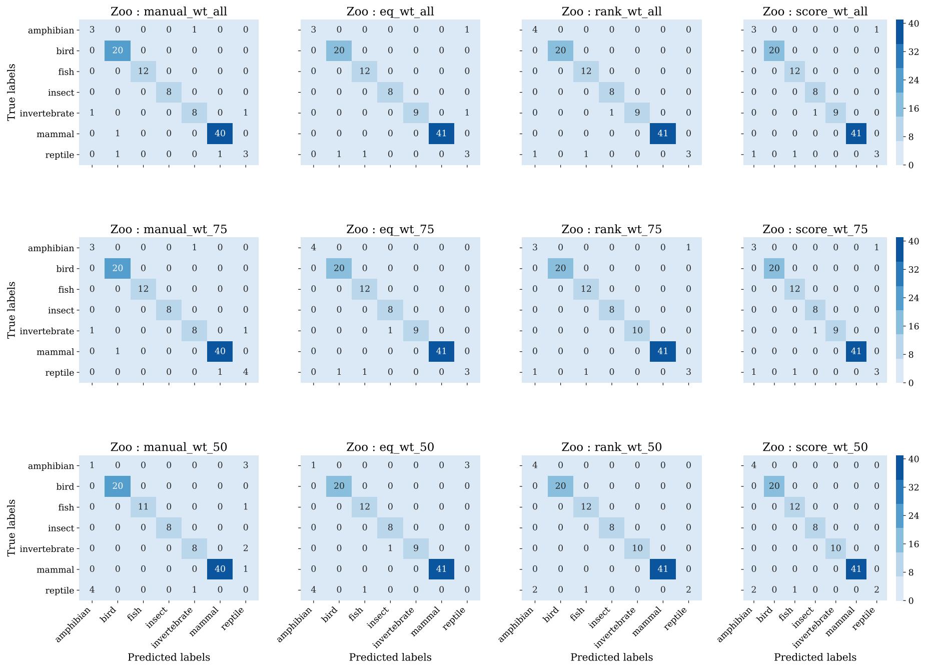

Fig. 2: Zoo dataset confusion matrices after one run of 10-fold cross validation with respect to the global similarity functions.

the datasets. Thus, in absence of domain knowledge our approach for predicting global similarity weights (*rank_all*) performs reasonably well across all the datasets.

Figure 4 presents the F1-score distributions with respect to the reduced feature percentages for all the four datasets. In this figure each row contains three sub-plots with respect to the *percent* values. The F1-scores are obtained based on 10-fold cross-validation for each global similarity function. The title of each plot describes the name of the dataset.

6 Discussion

The results are in accordance with the hypothesis of this paper: we can use distributions and statistical relationships within a dataset to define an initial global similarity measure. Our method can help to identify whether all or only a subset of features is necessary to carry out the desired classification task.

As the paper describes a method for discovering feature weights based on the data-driven approach with a possibility of feature reduction. The F1-score distributions in figure 4 shows whether a reduction of features has an effect on the retrieval. Since the entire approach of generating the similarity measures is automatic, we can now gradually reduce the number of features (selecting

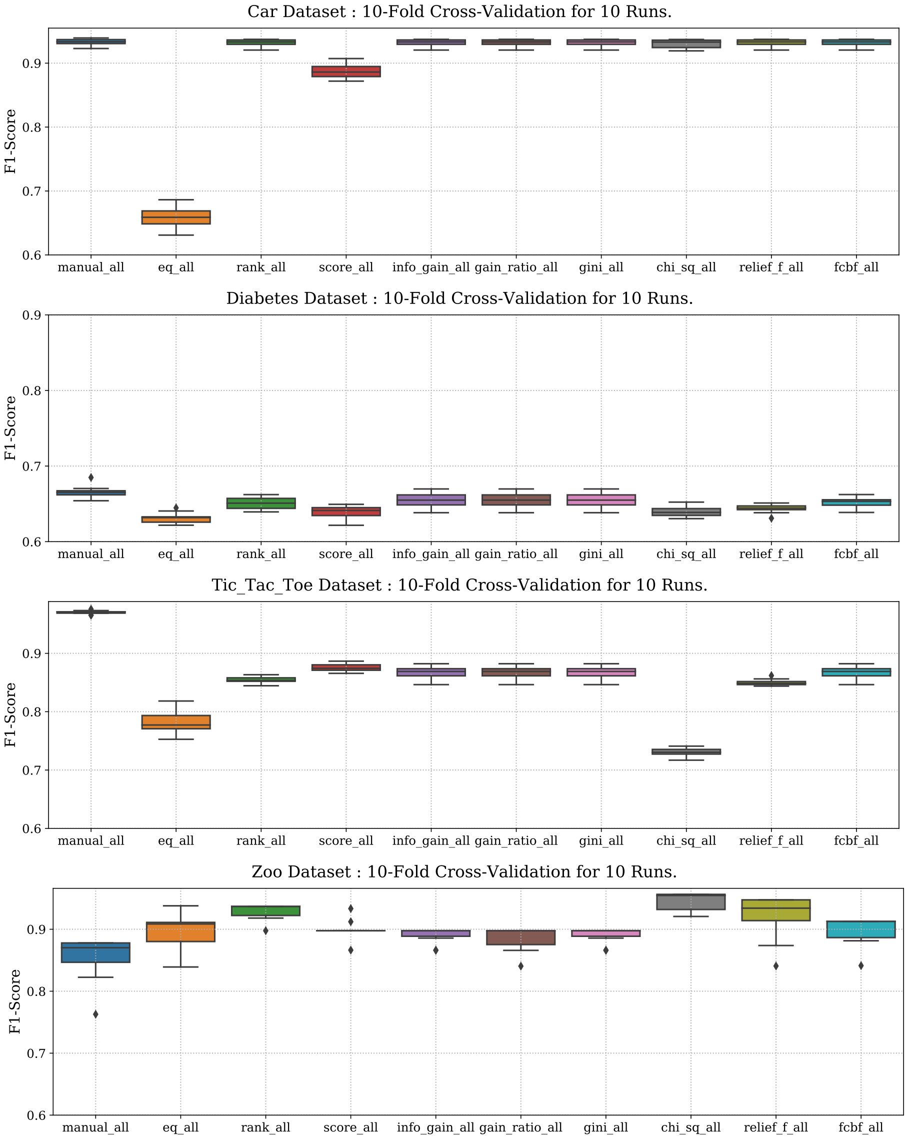

Fig. 3: Box-plots of evaluation metrics for 10-fold cross-validation over 10 runs. The plots are for datasets: Car, Diabetes, Tic Tac Toe, and Zoo respectively. All the plots are plotted with respect to the aforementioned global similarity functions, based on: manual, equal-weighted, rank, score, and the individual scoring methods.

smaller steps than presented here) and therewith find the best possible system,

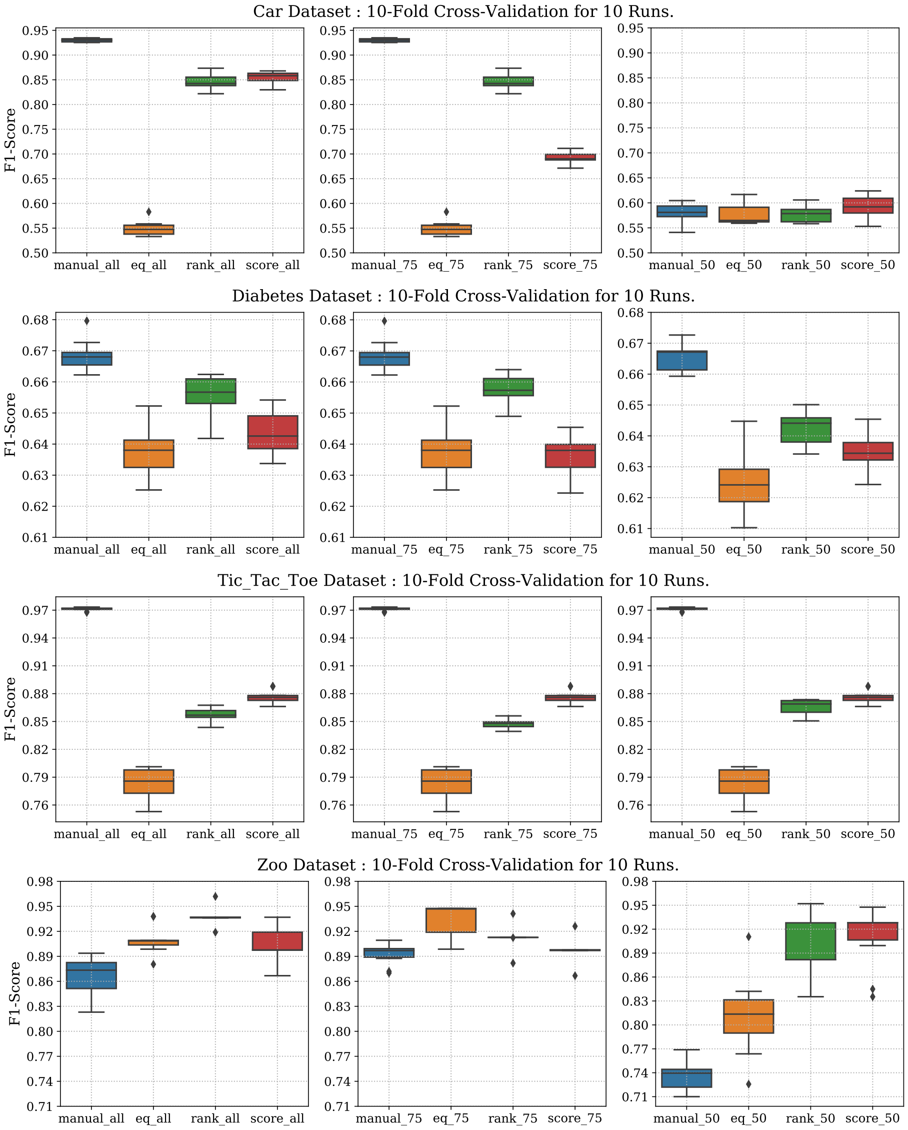

Fig. 4: Box-plots of F1-scores for 10-fold cross-validation with varying *percent* values. The plots are for datasets: Car, Diabetes, Tic Tac Toe dataset, and Zoo respectivelly. All the plots are plotted with respect to the aforementioned global similarity functions, on the x-axis.

before discussing the details with domain experts. This will save effort and has the advantage of collaborating and incrementally improving a CBR system.

The proposed method of the paper does not challenge the similarity modeling based on the domain knowledge. With one exception of the Zoo dataset, the manual knowledge engineering performed better than the automatic one - obviously, because a knowledge engineer can encode domain characteristics and is not dependent on the distribution within the dataset. However, in the absence of the correct domain knowledge, the presented approach holds better than equally weighted features. A similar pattern occurs when the number of features are reduced. As the features of the datasets are highly representative of the class, the significant difference might not be noticeable for the selected datasets.

In general, we can see that the global similarity function based on the rank and feature relevance scores are higher than the global similarity functions based on equal weights. However, the global similarity function based on manual modeling of the local and global similarities outperforms the automatically created CBR systems, except in the case of Zoo dataset. This is an expected outcome as the manual modeling is based on the domain knowledge.

In the absence of publicly available reference CBR systems we could not perform any comparison or bench-marking of our results. In order to allow other researchers comparisons with their work, we share our projects used in this paper[2].

7 Conclusion and Future Work

We have presented our method for discovering feature weights for modeling global similarity function for a CBR system using a data-driven approach. This method is well suited in the initial phases of a CBR system development. This method also provides an opportunity for the developer of the CBR system to discuss the setup with domain experts and present comparisons of the results to them in various configurations. Moreover, the feature selection is also supported and it's results can be compared against multiple choices of the *percent* values, as proposed in the paper. We comment that the method brings reduction in time of the development and prototyping phase of a CBR system. The approach does not involve iterations of learning, thus reduces the chances of over-fitting, which is also supported by the use of ensemble of multiple feature relevance scoring methods. With the publishing of the developed case bases and its similarity functions, the experiments become fully reproducible and can serve as reference implementations in the future.

Inspired by the results of the present method, we would use the approach over multiple other public datasets, and publish them to be used by researchers of CBR community. The future work in continuation with paper is to research on discovering the local similarities for symbolic features. Additionally, we currently apply this approach to the dataset of our ongoing research described in the paper [14,13], where a more complex dataset has been presented and different application scenarios are discussed.

[2] https://github.com/ntnu-ai-lab/cbr-benchmark-projects

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. Artificial Intelligence Communications **7**(1), 39–59 (1994)
2. Abdel-Aziz, A., Strickert, M., Hüllermeier, E.: Learning solution similarity in preference-based cbr. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **8765**, 17–31 (2014)
3. Aha, D.W.: Tic-tac-toe endgame database (1991), https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame
4. Aha, D.W.: Feature Weighting for Lazy Learning Algorithms, pp. 13–32. Springer US, Boston, MA (1998)
5. Aha, D.W., Goldstone, R.L.: Concept learning and flexible weighting. In: In Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society. pp. 534–539. Erlbaum (1992)
6. Bohanec, M.: Car evaluation database (1997), https://archive.ics.uci.edu/ml/datasets/Car+Evaluation
7. Ceriani, L., Verme, P.: The origins of the gini index: extracts from variabilità e mutabilità (1912) by corrado gini. The Journal of Economic Inequality **10**(3), 421–443 (Sep 2012)
8. Cost, S., Salzberg, S.: A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning **10**(1), 57–78 (1993)
9. Demšar, J., Curk, T., Erjavec, A., Črt Gorup, Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., Štajdohar, M., Umek, L., Žagar, L., Žbontar, J., Žitnik, M., Zupan, B.: Orange: Data mining toolbox in python. Journal of Machine Learning Research **14**, 2349–2353 (2013), http://jmlr.org/papers/v14/demsar13a.html
10. Forsyth, R.: Zoo database (1990), https://archive.ics.uci.edu/ml/datasets/Zoo
11. Gabel, T.: On the use of vocabulary knowledge for learning similarity measures. In: Althoff, K.D., Dengel, A., Bergmann, R., Nick, M., Roth-Berghofer, T. (eds.) Professional Knowledge Management. pp. 272–283. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
12. Hüllermeier, E.: Exploiting similarity for supporting data analysis and problem solving. In: Advances in intelligent data analysis : third international symposium. vol. 1642, pp. 257–268. Springer, Berlin (1999)
13. Jaiswal, A.: Personalized treatment recommendation for non-speciffc musculoskeletal disorders in primary care using case-based reasoning. In: Minor, M. (ed.) Workshop Proceedings of ICCBR 2018. pp. 214–218 (2018)
14. Jaiswal, A., Bach, K., Meisingset, I., Vasseljen, O.: Case representation and similarity modeling for non-specific musculoskeletal disorders - a case-based reasoning approach. In: The Thirty-Second International Florida Artificial Intelligence Research Society Conference (FLAIRS-32). pp. 359–362. AAAI Press (2019)
15. Kononenko, I., Šimec, E., Robnik-Šikonja, M.: Overcoming the myopia of inductive learning algorithms with relieff. Applied Intelligence **7**(1), 39–55 (1997)
16. Kullback, S.: Information Theory and Statistics. Wiley, New York (1959)
17. Novaković, J., Strbac, P., Bulatović, D.: Toward optimal feature selection using ranking methods and classification algorithms. Yugoslav Journal of Operations Research **21**(1), 119–135 (2011)

18. Pearson, K.: On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **50**(302), 157–175 (1900)
19. Prati, R.C.: Combining feature ranking algorithms through rank aggregation. Proceedings of the International Joint Conference on Neural Networks pp. 1–8 (2012)
20. Quinlan, J.R.: Induction of decision trees. Machine Learning **1**(1), 81–106 (1986)
21. Ricci, F., Avesani, P.: Learning a local similarity metric for case-based reasoning. In: Case-Based Reasoning Research and Development. pp. 301–312. Springer (1995)
22. Stahl, A.: Learning similarity measures: A formal view based on a generalized cbr model. In: Case-Based Reasoning Research and Development. pp. 507–521. Springer, Berlin, Heidelberg (2005)
23. Stahl, A., Gabel, T.: Optimizing similarity assessment in case-based reasoning. 21st national conference on Artificial Intelligence, AAAI'06 **21**, 1667–1670 (2006)
24. Stahl, A., Roth-Berghofer, R.T.: Rapid prototyping of cbr applications with the open source tool mycbr. In: Proceedings of the 9th European conference on Advances in Case-Based Reasoning. pp. 615–629. Springer (2008)
25. Tamoor, M., Gul, H., Qaiser, H., Ali, A.: An optimal formulation of feature weight allocation for cbr using machine learning techniques. In: IntelliSys 2015 - SAI Intelligent Systems Conference. pp. 61–67. IEEE (2015)
26. Valverde-Albacete, F.J., Peláez-Moreno, C.: 100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox. PloS one **9**(1), e84217 (2014)
27. Verma, D., Bach, K., Mork, P.J.: Modelling similarity for comparing physical activity profiles-a data-driven approach. In: International Conference on Case-Based Reasoning. pp. 415–430. Springer (2018)
28. VincentSigillito: Pima indians diabetes database (1997), `https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes`
29. Yu, L., Liu, H.: Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Proceedings of the 20th international conference on machine learning (ICML-03). pp. 856–863 (2003)