Applying CBR Principles to Reason without Negative Exemplars

Sidath Gunawardena, Rosina O. Weber

iSchool at Drexel, Philadelphia, USA
{sg349, rw37}@drexel.edu

Abstract

We investigate a method for applying CBR to a source of data where there are no negative exemplars. Our problem domain is one of recommending characteristics of multidisciplinary collaborators based on a collection of funded grants. Thus, there are no negative exemplars. Lacking sufficient domain knowledge, we seek to apply a feedback algorithm to learn weights even in the absence of negative exemplars. Our approach is based on the assumption that well aligned cases, cases where similar problems have similar solutions, are better suited for learning feature weights. Our approach clusters the problem and solution spaces separately to identify well aligned cases. We also identify poorly aligned cases that may hinder effective learning of weights, and exclude them. The clusters of well aligned cases provide a means to utilize feedback algorithms. We use two methods, case alignment and case cohesion, to show that our approach succeeds in identifying well aligned cases. We also compare our approach to a method based on single class learning, a machine learning approach for reasoning without negatives. Our results show that our approach is viable to learning weight in the absence of negative exemplars.

Introduction

Case-Based Reasoning (CBR) is used in a wide variety of reasoning tasks such as diagnosis, classification, and recommendations. In many problem domains, cases provide both positive and negative exemplars, allowing for the use of feedback algorithms to learn feature weights (Aha 1998). In this paper we consider applying CBR to problem contexts where there are no negative exemplars.

Our context is one where we seek to recommend, for an academic seeking a multidisciplinary collaboration opportunity, the characteristics of potential collaborators. Our dataset is a collection of funded multidisciplinary grants; this data lacks negative exemplars as we do not have information about collaborations that were not funded. We lack domain knowledge, so we seek a method that allows us to learn feature weights using a feedback algorithm even in the absence of negative exemplars.

We utilize the basic CBR tenet that similar problems have similar solutions (Leake 1996). Cases that meet this criteria we term *well aligned*, those that do not we term *poorly aligned*. To identify these two types of cases we apply a clustering algorithm that identifies clusters of dense regions, and also identifies as outliers data points in less dense regions. We choose density clustering as appropriate for this purpose (Richter and Weber 2013). We then utilize learned clusters in a feedback algorithm to learn feature weights (Aha 1998). As we separately in both the problem and solution spaces, we learn weights using results from both spaces and adopt feature weights that lead to best average accuracy. We also inform our research through a machine learning approach in the absence of negatives: Single Class Learning (SCL).

In the next section we detail our clustering-based approach to learning feature weights in the absence of negative exemplars. Then we verify our approach through methods from the literature that can identify aligned cases. We show that our clustering-based approach allows us to learn weights that provide an accuracy superior to not using weights. We compare our method to SCL. Finally, we present and discuss our results and end with our conclusions and thoughts for future work.

The Clustering-based Approach

In this section, we present our clustering-based approach to learning weights in the absence of negative exemplars.

A density clustering algorithm, (e.g., DBSCAN (Ester 1996)), is applied first on the problem and then on the solution space of the cases. Each result is used to learn weights using a feedback algorithm. The outliers identified by the algorithm are ignored when learning feature weights. Then all cases, including the outliers, are used to evaluate the different sets of weights learned.

We evaluate this approach on a case base of collaborations, which is composed of 198 cases. It is used to recommend, for a new collaboration seeker, the characteristics of two collaborators to form a three person collaboration. Each collaborator is described by three features: title, research interest, and institution type, e.g. a full professor of biology at a doctoral degree granting institution. The features are the same in both the problem and solution spaces.

In this implementation, we use DBSCAN to cluster the cases based on the distances between the problems and then based on the distances between the solutions. DBSCAN requires the specification of two parameters. The maximum distance between any two points (ε); and the minimum number of points required to form a cluster.

The distance measure between both problems and solutions is a variant of edit distance (Levenshtein 1966), where the distance between two collaborations is the number of features that need to be changed to make them identical. If the feature values to be changed are similar, e.g., Assistant Professor to Associate Professor, then the cost is only one half edit. The problems have three features, so the maximum edit distance between problems is three. The solutions have six features so the maximum edit distance between solutions is six.

The specific configurations of the density clustering algorithm are specified in the next section where we verify that our approach identifies well aligned cases.

Verifying the Clustering-based Approach

We evaluate the use of clustering to identify cases that are well aligned. We cluster using different parameters and find the best results using the parameters ε = 0.5, and seven as the minimum number of points to from a cluster. We compare the results against two methods that attempt the same, cohesion (Lamontagne 2006) and alignment (Massie et al. 2007).

Cohesion (Lamontagne 2006)

For a given case t(p_t, s_t), its cohesion score is a measure of whether t behaves similarly to its neighboring cases in both the problem and solution spaces. This requires specifying problem threshold (δ_P) and solution threshold (δ_S). For a case base C (c_1,..., c_n) cases can be divided, in relation to case t, into four types (Table 1).

		Problem	
Solution		Sim(p_t, p_{ci}) >= δ_P	Sim(p_t, p_{ci})< δ_P
	Sim(s_t, s_{ci}) >= δ_S	A	C
	Sim(s_t, s_{ci}) < δ_S	B	D

Table 1: Similarity Between Problem And Solution Parts

Case Cohesion (t) = A / (A+B+C)

Comparison to Cohesion

We calculate the average cohesion for the clustered cases (i.e., well aligned) and for the outliers (i.e., poorly aligned) resulting from the density clustering algorithm. **Table 2** and Table 3 show these results along with the percentage of poorly aligned cases. We see that clustering on the solution space results in the highest Cohesion scores.

Thresholds	Well Aligned	Poorly Aligned	% Poorly Aligned
$\delta_P = 0.5$, $\delta_S = 0.5$	0.53	0.39	1%

Table 2: Mean Cohesion Scores: Well and Poorly Aligned Cases Based On Problem Space

Thresholds	Well Aligned	Poorly Aligned	% Poorly Aligned
$\delta_P = 0.5$, $\delta_S = 0.5$	0.57	0.46	18%

Table 3: Mean Cohesion Scores: Well and Poorly Aligned Cases Based on Solution Space

Alignment (Massie *et al.* 2007)

Alignment (Massie et al. 2007) introduces a component of how well a case aligns with the problem and solution space overall. For a case t(p_t, s_t) problem p_t and solution s_t, its alignment with a case c_1 (p_1, s_1) is:

$$\text{Align}(t, c_1) = 1 - \frac{(D(s_t, s_1) - D_{Smin})}{(D_{Smax} - D_{Smin})}$$

Where D_{Smax} and D_{Smin} are the distances to furthest and nearest solutions of t. The overall alignment of t is calculated for a given number of nearest neighbors (n):

$$\text{Case Alignment}(t, c_1) = \frac{\sum_{i=1 to n} (1 - D(p_t, p_1)) \text{ x Alignment}(t, c_1)}{\sum_{i=1 to n} (1 - D(p_t, p_1))}$$

Comparison to Alignment

We repeat the comparison with Massie alignment scores for the well and poorly aligned cases using three nearest neighbors (NNs). **Table 4** and Table 5 show these results along with the percentage of poorly aligned cases. We see that clustering on the solution space results in the highest alignment scores.

NNs used to calculate Alignment	Well Aligned	Poorly Aligned	% Poorly Aligned
3	0.61	0.60	1%

Table 4: Mean Alignment Scores: Well and Poorly Aligned Cases Based on Problem Space

NNs used to calculate Alignment	Well Aligned	Poorly Aligned	% Poorly Aligned
3	0.64	0.57	18%

Table 5: Alignment Scores: Well and Poorly Aligned Cases Based on Solution Space

These results verify that the clustering-based approach results in a set of cases that have higher Cohesion and Alignment scores than the outliers. The best results occur when using the clusters on the solutions space. We can now use any feedback algorithm to learn features weights from clusters used as classes. We next compare this approach to a machine learning approach for reasoning without negative exemplars: single class learning.

Comparison to Single Class Learning

We describe the Single Class Learning (SCL) approach and how we adapt this approach to recommend collaborations using our dataset. We will then compare their resulting accuracy for recommending characteristics of collaborators.

Single Class Learning

SCL is a machine learning technique recommended for use when the data consists of only positive and unlabeled instances. The characteristics of the positive instances are used to build rules to identify likely negative instances in the set of unlabeled instances (Liu et al. 2003). SCL methods typically use a two-step process where the first step learns rules to identify negative instances, which are then used in a second classification step via methods such as SVM (Yu et al. 2004), Naïve-Bayes (Denis 2002), or Expectation Maximization (Liu 2002).

We greatly simplify this method to apply it in our problem context where there are only positive instances. We consider all the possible problem and solution combinations in the respective spaces of problem and solution features. We then consider the problems and solutions not represented in the set of cases as unlabeled data. We then consider the cases with combinations of features that do not occur, or occur seldom, to be examples of instances that are likely negative. These can then be translated into rules to identify negative instances based on their feature values. This approach is feasible where the feature space is relatively small.

This approach will determine whether a case is or not a negative instance by looking at the overall combination of feature values in both problems and solutions. For the collaboration cases from the data we described, an example of a negative instance would be a case that the problem (i.e., seeker) is an assistant professor and the solution is of two assistant professors. Each negative becomes a rule, that is, one rule is that a collaboration cannot have three assistant professors.

The way we use the SCL for recommendation of collaborations is in two steps. First, the case-based reasoner with no feature weights recommends a solution. Second, an additional step verifies whether the recommended characteristics in the solution results a collaboration that violates any of the rules learned with SCL.

Consequently, to compare with our clustering-based CBR approach, we generate recommendations using no weights, and use the SCL method to determine which recommendations should be made or not. For the collaboration cases, we test combinations of features that occur less than 1%, 5% and 10% of the time as likely to be negative instances of collaborations.

Combining the Approaches

We also test a configuration that combines both methods. We use the feature weights learned from the clustering-based approach and the SCL rules to make recommendations. If the resulting collaboration is one that violates a rule, then it is discarded and the second best option chosen and so on, until there are no recommendations left or one does not violate the rules.

Evaluation

We evaluate the accuracy of each configuration using a Leave-One-Out Cross-Validation (LOOCV). Accuracy is measured by the number of edits required to transform the recommended solution into the solution of the left out case. The smaller this edit distance, the more accurate the recommendation. We normalize the edit distance and convert it to a percentage that represents, for each iteration of the LOOCV, how well the recommended solution matches the solution of the left out case.

For the purposes of evaluation all the cases are used, including the poorly aligned cases. This presents a consistent basis for evaluation across all the configurations.

Results and Discussion

Table 6 lists the results of the accuracy from using no weights, the feature weights learned from the problem cluster, the feature weights learned from the solution clusters, the SCL rules, and the combined approach that uses the best set of feature weights learned and the SCL rules.

We present the best result for each of the configurations. For the SCL approach, this was when the rules were based on collaborations that occurred less than 5%. For the clustering-based approach this is for the clusters generated using an epsilon of 0.5 and at least 7 points to a form a cluster.

No Weights	Problem Cluster Weights	Solution Cluster Weights	SCL Rules	Combined: Rules+Solution Weights
63.5	63.9	67.2	66.7	70.4

Table 6: Average Accuracy (%)

The results demonstrate that the clustering-based approach is viable when applying CBR to a context without negative instances by showing an improvement over the no weights. We see that the combination of feature weights learned from the solution clusters and the rules learned from SCL provides a higher level accuracy. The percentage of outliers and the values computed by cohesion and alignment measures could have been interpreted as pointing to this. SCL provides another method to learn knowledge from the data in a way not contemplated by the CBR solution.

It should be noted that although the methods performed better than using no weights, the levels of accuracy are not overly impressive. This is related to fact that all cases are used during evaluation including the poorly aligned ones. We believe that this is a necessary component of a recommendation method due to diversity.

Related Work

Not all cases may equally contribute to the overall function of the case base (Smyth and McKenna, 1999). Delany (2009) demonstrates that removing cases from a case base can improve accuracy by reducing noise. Identifying such cases for CBR can be done via comparing a cases behavior to its similar neighbors (Lamontagne 2006, Massie *et al*., 2007). Cases that behave in a manner similar to their neighbors are more suitable to learn feature weights that represent their relative relevance because they are better aligned, i.e., they meet the basic assumption that similar problems have similar solutions (Leake 1996).

Conclusions and Future Work

In this paper we investigate a method to apply CBR in a problem context that lacks negative instances by leveraging the CBR tenet that similar problems have similar solutions. We developed a clustering-based approach to identify cases that are well versus poorly aligned. We verified this approach through measures for alignment (Massie *et al*., 2007) and cohesion (Lamontagne 2006) from the literature.

We leveraged the clusters created by our proposed method to apply supervised learning algorithms to learn feature weights. We compared this approach to one based on the principles of SCL, a machine learning approach used to reason without negatives. We found that while both approaches performed better than using no feature weights, a combination of the methods led to the best accuracy.

In the future we intend to expand on this simple application of SCL to utilize methods such as expectation maximization to determine the characteristics of potential negative instances. We seek to further validate these rules to contribute to the knowledge of the domain. We also seek to test this approach on higher dimensional datasets.

Acknowledgements

This work is supported by U.S. EPA STAR Program and the U.S. Dept. of Homeland Security; Grant # R83236201.

References

Aha, D.W. 1998. Feature weighting for lazy learning algorithms. In H. Liu & H. Motoda eds. Feature Extraction, Construction and Selection: A Data Mining Perspective. 13–32. Norwell, MA: Kluwer.

Denis, F., Gilleron, R., and Tommasi, M. 2002. Text classification from positive and unlabeled examples. In 9th International Conference Information Processing and Management of Uncertainty in Knowledge Based Systems, 1927–1934. ACM Press.

Delany, S. 2009. The Good, the bad and the incorrectly classified: profiling cases for case-base editing. Case-Based Reasoning Research and Development, 135-149.

Elkan, C., and Noto, K. 2008, August. Learning classifiers from only positive and unlabeled data. In Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 213-220. ACM Press.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. 1996 . A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, U. M. Fayyad eds. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining KDD-96, 226–231. AAAI Press: Menlo Alto.

Lamontagne, L. 2006. Textual CBR Authoring using Case Cohesion. In Proceedings of 3rd Textual Case-Based Reasoning Workshop at the 8th European Conf. on CBR, 33-43.

Leake, D. B., ed. 1996 Case-Based Reasoning: Experiences, Lessons, and Future Directions. Menlo Park, CA: AAAI Press/MIT Press.

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics-Doklady, 8: 707–710.

Liu, B., Lee, W. S., Yu, P., and Li, X. 2002 . Partially supervised classification of text documents. In Proceedings of the Nineteenth International Conference on Machine Learning, 387-394.

Liu, B., Dai, Y., Li, X., Lee, W. S., and Yu, P. S. 2003, November . Building text classifiers using positive and unlabeled examples. In Data Mining, 2003. ICDM 2003. Third IEEE International Conference on Data Mining,179-186. IEEE Press.

Massie, S., Wiratunga, N., Craw, S., Donati, A., and Vicari, E. 2007. From anomaly reports to cases. Case-Based Reasoning Research and Development, 359-373.

Richter M.M. and Weber R.O. 2013. *Case-based Reasoning: a textbook*. Berlin, Springer, in Press.

Smyth, B., and McKenna, E. 1999. Footprint-based retrieval. Case-Based Reasoning Research and Development, 719-719.

Yu, H., J. Han, J. and Chang, K. C.-C. 2004. PEBL: Web page classification without negative examples, IEEE Transactions on Knowledge and Data Engineering, 16(1): 70–81.