Reasoning with Organizational Case Bases in the Absence Negative Exemplars

Sidath Gunawardena, Rosina O. Weber

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

Abstract. Organizational case bases are gathered based on the organization they serve; cases are not selected taking reasoning into account. Thus, organizational case bases may lack negative exemplars and have multiple solutions to one problem, making it difficult learn weights for reasoning. Case bases in typical Process-Oriented Case-Based Reasoning (POCBR) contexts are organizational, thus inheriting those problems. This paper describes an approach to identify a subset of cases from an organizational case base that meets the criterion that similar problems have similar solutions. This subset is then used to characterize classes, establishing positive and negative exemplars that are then used to learn weights for reasoning with the entire case base. We apply this approach to three organizational case bases, showing significant improvements in accuracy with weights learned with this approach in case bases without negative exemplars.

Keywords: organizational case bases, process-oriented case-based reasoning, processes, workflows, negative exemplars, learning, weights

1 Introduction

The focus of CBR towards tasks, processes, and workflows widens the applicability and usefulness of the CBR methodology. This focus led to the rise of Process-Oriented Case-Based Reasoning (POCBR) systems [10], [16]. POCBR systems are typically organizational in that cases are included because they are relevant to the organization the system is designed to support. The decision of which cases to include does not take into account the purpose of reasoning. This results case bases that are difficult to use for reasoning.

CBR systems deployed in organizational contexts aim to solve a variety of problems. When POCBR systems are used for managing and modeling workflows, the goal is to find one workflow sufficiently similar to a workflow in use to be adapted. In these problems, it is hard to determine what makes one workflow more similar to another to

assign weights to represent relative relevance. Identifying negative exemplars is also hard because given enough adaptation knowledge, any workflow can be considered similar.

When POCBR systems are used for recommendation (e.g., e-commerce, expert locator systems), case bases include characteristics of entities. This is another class of problems where there may be no negative exemplars. Conceptually it is hard to say whether a combination of characteristics is not suitable or it simply n ever happened. This issue may also be present in other uses of POCBR systems such as prediction and simulation, where the goal is to identify a similar workflow to reuse its sequence or next step.

The absence of negative exemplars can be very problematic as it prevents the use of the feedback algorithms typically recommended for learning weights [1]. For this reason, we present a systematic approach for organizational case bases.

The approach we present takes a case base and reduces it to a subset that meets the criterion that similar problems have similar solutions. In this process, it eliminates boundary cases and some diverse cases. The approach creates clusters that are used as classes, allowing the distinction between positive and negative exemplars as cases that, respectively, belong or not to a class. This resulting subset of cases organized in classes enables the learning of weights to represent relative relevance of individual features. Subsequently, boundary and diverse cases can be incorporated again into the case base.

In the next section we present a general description of our method. In the following section, we describe a study where we apply the method to three case bases. The study shows that our method leads to learning weights that result in average accuracies that are equivalent to alternative methods when negative exemplars are available. In the absence of negative exemplars, our proposed method leads to learning weights that result in average accuracies that are significantly higher than when no weights are used. We then summarize some related work, and conclude with a few remarks on the implications of the results and future work.

2 Method

2.1 Introduction

We want to overcome the problem of lack of negative exemplars and the presence of diverse and boundary cases so we can learn weights that represent the relative relevance of features. To this end we seek to identify core cases: sets of cases where either the problem parts or the solution parts meet a certain threshold of similarity. Those cases will become instances of a class. In this approach, a boundary case does not have sufficient cases near it to be a core case. A diverse case is a core case whose solution (or problem) is not sufficiently similar to the solutions (or problems) of other core cases belonging to the same class. While such cases are valuable as they promote diversity, they violate the CBR tenet that similar problems have similar solutions and inhibit the ability to learn a set of

consistent weights. **Fig. 1** represents cases in problem-solution pairs (P_i, S_j), and shows three cases {P_2, S_2}, {P_5, S_5} and {P_6, S_6} that are examples of core cases. {P_4, S_4} is a boundary case as there are not enough cases similar to its problem or solution. {P_1, S_1} and {P_3, S_3} are diverse cases as they violate the principle that similar problems have similar solutions. These cases may be the same, have some overlap, or be completely disjoint.

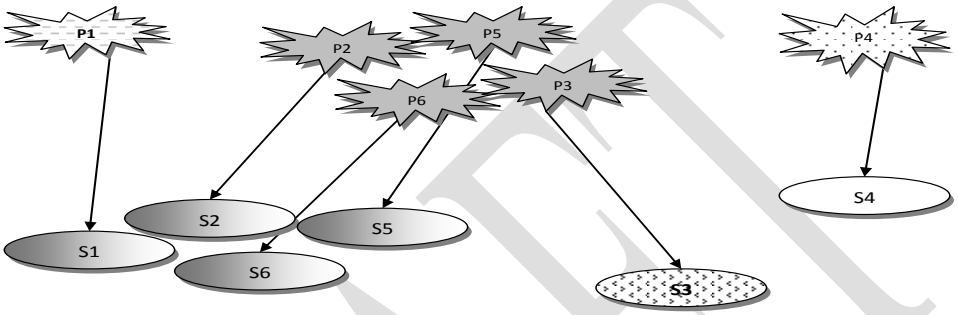

Fig. 1. Examples of core, diverse and boundary cases

Our method is motivated by the CBR assumption that similar problems have similar solutions [7]. We identify cases that meet this criterion by removing cases that violate it by clustering cases based on problems and solutions. The resulting clusters are treated as classes where positive exemplars belong and negative exemplars do not. By removing all cases that do not comply with these distinct classes, we eliminate boundary cases and some diverse. In absence of domain knowledge, our method further assumes that two entities (e.g., problems, solutions) are similar when they share common features of a given representation [16].

2.2 General Methodology

Our methodology is comprised of three steps: in the first step we employ a density clustering algorithm to remove boundary and diverse cases. We illustrate this methodology in **Fig. 2**. Step 1 is comprised of two independent phases where cases are clustered both on the problems and also on the solutions. In the Step 2 we utilize the clusters to learn weights and in Step 3 we evaluate the quality of the weights generated by both clustering phases by applying them to assess the average accuracy of the entire case base. The weights learned after the clustering phase that result higher accuracy on overall case base are recommended for adoption.

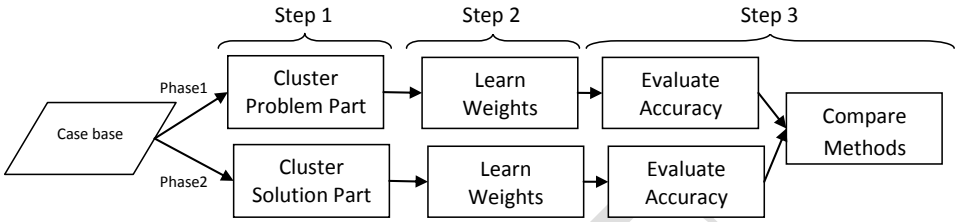

Fig. 2. General Methodology

Step 1 identifies groups of cases based on the problem space and on the solution space. We cluster cases based on both problem and solution because we do not know before-hand which will produce the better clusters overall. During clustering, both outliers and diverse cases are excluded. The resulting clusters will then be used as classes in Step 2.

We use a density clustering algorithm as they target removal of outliers, so it removes cases that are diverse and boundary. The goal is to obtain a clear distribution of cases in clusters as shown in Fig. 3a and 3b. Note that Fig. 3a refers to clustering based on the problem space, and Fig. 3b to clustering on the solution space.

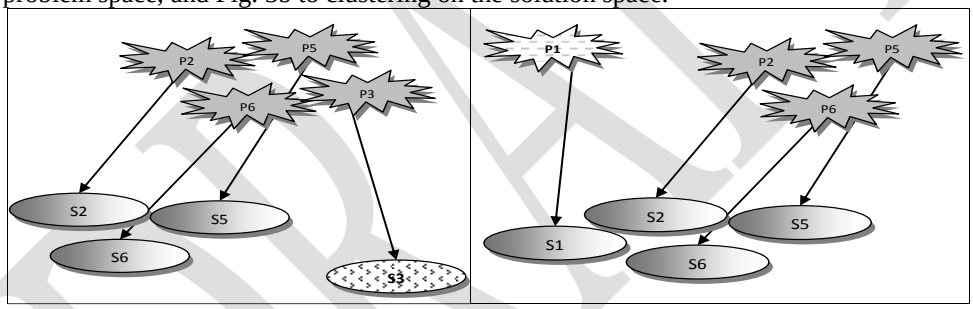

Fig. 3a. Clustered cases based on problems **Fig. 3b**. Clustered cases based on solutions

Step 2 is where the resulting classes enable learning of weights. Any feedback algorithm with performance bias can be used for learning weights for classifiers. The clusters are the reference outcome. The learned weights are then used in the CBR system.

Step 3 is the evaluation step. The evaluation is done on the entire case base, which includes cases removed during clustering. We use cross-validation to determine the resulting average accuracy for each set of weights learned from each phase in Step 1. In other words, there are two evaluations, one that uses weights learned based on clustering the problems (Phase 1), one that uses weights learned based on clustering the solutions (Phase 2). The evaluation computes average accuracy using a suitable similarity measure. The more accurate weighting scheme is to be adopted.

2.3 Further Aspects

The clustering is run separately on problem and solution parts. We note that if a case is removed in both it is a boundary case and if only in one it is a diverse case. The resulting clusters depend on the specifics of the data, that is, whether problems or solutions are more similar to each other. It should be noted that if there are a sufficient number of diverse cases so that they form their own core cluster, then they will not be removed.

An important aspect of this method is that it uses the uncertain concept that similar problems have similar solutions. The uncertainty inherited from this concept is embedded in its results. It is not the goal of the method to eliminate this uncertainty but rather to make the dataset usable. In the next section we describe an example using the method.

3 Applying the Method

We wish to investigate the following hypotheses:
H1. In the absence of negative exemplars, clusters of cases can be used as classes to identify negatives to learn weights that lead to significant improvements in average accuracy in organizational case bases over feature counting.
H2. In the presence of negative exemplars, clusters of cases used as classes to identify negatives to learn weights lead to average accuracy in organizational case bases equivalent to when the actual negatives are used.

3.1 Data

We apply our proposed method to three organizational case bases. Casebase 1 describes a process at a high level with the solution being a specific implementation of that process. This case base does not have negative exemplars. Casebase 2 describes a collaboration of entities. It also lacks negative exemplars. It describes a recommendation problem. Casebase 3 describes a process with the solution representing the success or failure outcome of the process. This is the only case base that has negative exemplars and describes a binary classification problem. **Table 1** shows a summaries the case bases.

Table 1. Summary of case base characteristics

	Number of Cases	**Number of Features**	**Has Negative Exemplars**	**Problem**	**Solution**
Casebase 1	254	3	No	Abstract Process	Specific Process
Casebase 2	198	3	No	Collaboration Seeker	Recommended Collaboration

Casebase 3	88	23	Yes	Description of Process	Success/Failure

3.2 Experiments

In all three case bases we expect diverse and boundary cases. When applying our proposed method, we use the standard density clustering algorithms DBSCAN [4]. The clusters are used to learn weights with a genetic algorithm, and the performance of the learned weights is evaluated via a Leave-one-out Cross-Validation (LOOCV) on each case base.

Step1. For Casebase 1 & 2, we implement phases 1 and 2, i.e., clustering based on both problems and solutions. For Casebase 3, which describes binary classification cases, solutions are either 0 or 1, so we only cluster on the problem part of the cases. **Table 2** shows resulting number of cases and number of clusters.

Table 2. Step 1: Clustering cases based on problems and solutions

	Casebase 1 254 cases		**Casebase 2 198 cases**		**Casebase 3 88 cases**	
	# of cases	# of clusters	# of cases	# of clusters	# of cases	# of clusters
Phase 1: Clustering on problems	129	40	189	8	31	11
Phase 2: Clustering on solutions	77	29	124	10	NA	

Step 2. The clusters now provide us with positive and negative exemplars of the classes they represent. We use the ability to correctly make this classification as the fitness function to learn weights via a genetic algorithm. Two hundred randomly generated sets of weights represent the chromosome, where each individual weight can be thought of as a gene. The algorithm is run for 1000 iterations. In all iterations the better performing chromosomes (sets of weights) have a greater chance of contributing their genes (weights) towards the next generation. To reduce the likelihood of being stuck in a local maximum we introduce a 5% of mutation where instead a random gene is inserted

Step 3. In this step, we evaluate the quality of the weights. For this we use the learned weights and assess accuracy for the entire case base based on the subsets of cases determined in Step 1. We now present those results in **Table 5** to 5. To evaluate the statistical significance of the experiments, we conducted separate one way within-subjects ANOVAs followed by post-hoc analysis with Tukey's Honest Significant Difference test, with alpha = 0.5. Table 3 shows the average accuracy for Casebase 1.

Table 3. Casebase 1, Average accuracy using LOOCV, * significant difference at $\alpha = 0.05$

Similarity	Feature counting	Weights From Problem Clusters	Weights From Solution Clusters
kNN =1	23%	24%	28%*

Casebase 1 has 254 cases and no negative exemplars. We cluster on both the problem parts and solution parts. We learn two sets of weights based on these clusters. Accuracy is measured based on a gold standard, where for 81 cases of the 254, the next best solution is determined manually. This gold standard is used as basis for LOOCV run on the 81 cases, where only the top scoring result is selected. Where there are ties, one of the tied results is chosen randomly. This process is repeated 10 times for each set of weights, and the average is presented here. The results show a statistically significant improvement (α =0.05) when using the weights from the solution clusters.

Table 4. Casebase 2, Average accuracy using LOOCV, * significant difference at α = 0.05

Similarity	Feature counting	Weights from negatives from problem clusters	Weights From negatives from Solution Clusters
Sim1	63%	63%	67%*
Sim2	66%	65%	67%*
Sim3	63%	64%	67%*

Casebase 2 has negative no exemplars, and so we cluster on both the problem parts and solution parts. The different similarity functions are based on the level of abstraction. We learn two sets of weights based on these clusters. Then for the entire case base, the learned weights from clustering on problems and solutions are compared to using no weights. The accuracy is measured as the edit distance between the recommended solution and the solution of the removed case. The results show a statistically significant improvement (α =0.05) when using the weights from the solution clusters.

Table 5. Casebase 3, Average accuracy using LOOCV, * significant difference at α = 0.05

Similarity	Feature counting	Weights learned from actual negatives	Weights learned from negatives from clusters
kNN =1	80%	85%	84%
kNN =3	81%	92%	92%

The results in Table 5 show no significant difference between weights learned from actual negatives and weights learned from negatives from clusters. The resulting average accuracy when actual negatives are used are equivalent to average accuracy resulting when negatives are defined based on interpreting clusters as classes.

3.3 Results & Discussion

Table 6. H1. Absence of Negative Exemplars

Case base	Negative Instances	Improvement over Feature Counting?	Difference is Statistically Significant
Casebase 1	No	Yes	Yes
Casebase 2	No	Yes	Yes

The results of applying our method on case bases 1 and 2 support our Hypothesis 1, "In the absence of negative exemplars, clusters of cases can be used as classes to learn weights that lead to significant improvements in average accuracy in organizational case bases over feature counting."

Table 7. H2. Presence of Negative Exemplars

Case base	Negative Instances	Comparable to using actual Negative Exmplars?	Difference is Statistically Significant
Casebase 3	Yes	Yes	No

Results in Table 7 supports Hypothesis 2, "In the presence of negative exemplars, clusters of cases used as classes to identify negatives to learn weights lead to average accuracy in organizational case bases equivalent to when the actual negatives are used."

Our results show that learning weights from clustering on the solutions and removing boundary and diverse cases can lead to a significant increase in the accuracy when compared to using no weights. When this method is applied where negative instance exist, it produces comparable results to standard methods.

The weights produced from clustering on the problems do not increase accuracy from not using weights. Thus, it is the set of core cases resulting from clustering on solutions that better meets the requirement that similar problems have similar solutions.

4 Related and Background Work

[1] has discussed the use of weight learning methods with (i.e., *performance bias*) and without feedback for classifiers (i.e., *preset bias*). In this paper, we discuss datasets that are not necessarily being used for classification tasks. The most important problem we face is the lack of negative exemplars, i.e., we do not know what a negative instance of a process looks like. Previous work on reasoning with no negative examples focuses mainly on single-class learning and classification problems where there is a large collection of

unlabelled data and a small collection of positively labeled data, e.g., web pages and DNA sequences. Typical methods iterate a two-step process where the first step learns heuristics to identify negative instances that can then be used in the second step by a classifier such as SVM [15], [17] or Naïve-Bayes [2],[8].

Previous work has experimented with generating artificial negative exemplars via induction, where all feasible cases that do not exist are considered to be negatives [6]. Because these organizational case bases we discuss may very likely need diverse solutions, we do not know if a process that is not present is a bad example or one that does not exist or that has not yet been tested.

Process cases are complex and learning models of the entire problem space can lead to overgeneralization. Clustering has been using in process mining to subdivide the problem space so multiple models can be learned [5], [14]. For choice of algorithm, we select density clustering, recommended when we want to remove outliers [13]. The general idea of density clustering is it does not determine a centroid or number of clusters, but that the data has areas of density that we want to cluster. Among a variety of density clustering algorithms, we use DBSCAN [4]. In this paper we do not however plan to make a thorough review or assessment of density clustering methods but simply to demonstrate our approach.

Works that explored the CBR requirement of similar problem being associated with similar solutions include [9] who show that boundary cases can also affect performance of classification systems and to improve classification accuracy suggesting valid cases may need to be removed. This property of CBR is also investigated by [3] who shows that some cases can also be a liability to the case base by promoting misclassification. Other approaches employ singular value decomposition [12] to reduce dimensionality, or use a threshold based on verified cases [11] to determine 'good' cases.

5 Concluding Remarks

POCBR case bases are typically gathered because they are pertinent to an organization and not due to their potential contribution to reasoning. This organizational orientation produces case bases that may lack negative examples and include boundary and diverse cases. Our proposed method takes a set of cases and reduces it to a set that, within a reasonable distance, have similar problems with similar solutions.

We describe a study using three case bases, two that lack negative exemplars and one where negative examples. The study shows that our method leads to learning weights that result in average accuracies that are equivalent to an alternative method when negative exemplars are available. In the absence of negative exemplars, we note that there are no alternative systematic methods and thus we compare the resulting accuracy of the case bases using weights learned from our method versus using no weights. The results for both datasets produce significantly higher average accuracy.

This work is a first step towards dealing with real world datasets from organizations that have many experiences to contribute, mostly being processes or workflows. The approach discussed here can also benefit datasets that do not necessarily represent processes; but like business processes, are gathered by one organization and may be difficult to learn due to the lack of negative exemplars. It is suitable for learning weights when systems are being designed, and also for systematic maintenance.

Among the next steps we plan to refine the method further. Specifically, we want to investigate the impact of removing more cases after clustering. For example, when clustering based on the problems, we will remove cases whose solutions to not fit well the clustering organization provided by the problems. We will also investigate different clustering algorithms and their potential for this task.

6 Acknowledgments

The authors are supported in part by the U.S. EPA Science to Achieve Results (STAR) Program and the U.S. Department of Homeland Security Programs, Grant # R83236201. We would like to thank our reviewers for their detailed and insightful feedback.

7 References

1. 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. Norwell, MA: Kluwer.
2. Denis, F., Gilleron, R., Tommasi, M., (2002). Text classification fro positive and unlabeled examples. In: The 9th Internat. Conf. Information Processing and Management of Uncertainty in Knowledge Based Systems, IPMU 2002, pp. 1927–1934.
3. Delany, S.J. (2009). The Good, the Bad and the Incorrectly Classified: Profiling Cases for Case-Base Editing. ICCBR 2009: 135-149
4. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, Usama M. Fayyad (eds.). Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press:Menlo Alto. pp. 226–231.
5. Ferreira, R., Zacarias, M., Malheiros, M., and Ferreira, P. (2007) Approaching Process Mining with Sequence Clustering: Experiments and Findings. BPM: 360-374
6. Goedertier, S., Martens, D. , Vanthienen, J., Baesens, B. (2009). Robust Process Discovery with Artificial Negative Events. Journal of Machine Learning Research 10: 1305-1340
7. Leake, D. B., ed. (1996). Case-Based Reasoning: Experiences, Lessons, and Future Directions. Menlo Park, CA: AAAI Press/MIT Press, Menlo Park, CA.
8. Liu, B., Lee, W. S., Yu, P., and Li, X. (2002). Partially supervised classification of text documents. ICML-02.

9. Massie, S., Craw, S., Wiratunga, N. (2007). When Similar Problems Don't have Similar Solutions. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp.92-106 . Springer, Heidelberg
10. Minor, M., Bergmann, R., Görg, S., and Walter, K. (2010). Towards Case-Based Adaptation of Workflows. In I. Bichindaritz and S. Montani (Eds.): ICCBR 2010, LNAI 6176, pp. 421–435, 2010. Springer-Verlag Berlin Heidelberg
11. O'Mahony, M., Hurley, N. and Silvestre, G. (2006). Detecting noise in recommender system databases. In Proceedings of the 11th international conference on Intelligent user interfaces (IUI '06). ACM, New York, NY, USA, pp. 109-115
12. Symeonidis P. (2007). Content-based Dimensionality Reduction for Recommender Systems, Proceedings of the 31st Conference of the German Classification Society (GfKl'2007), Freiburg.
13. Richer, M. M. and Weber, R. O. (2012). Case-based reasoning: a textbook. Berlin: Springer. In press.
14. Veiga, G., Ferreira, D. (2009). Understanding Spaghetti Models with Sequence Clustering for ProM. Business Process Management Workshops: 92-103
15. Wang, C., Ding, C., Meraz, R. F. and Holbrook, S. R.. (2006). PSoL: a positive sample only learning algorithm for finding non-coding RNA genes. Bioinformatics, 22(21):2590–2596.
16. Weber, B., Wild, W., Breu, R. (2004). Cbrflow: Enabling adaptive workflow management through conversational case-based reasoning. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 434–448. Springer, Heidelberg
17. Yu, H., J. Han, J. and Chang,, K. C.-C. (2004). PEBL: Web page classification without negative examples, IEEE Trans. Knowl. Data Eng., vol. 16, no. 1, pp. 70–81, Jan.