Learning and Applying Case Adaptation Rules for Classification: An Ensemble Approach

Vahid Jalali  
Indiana University  
vjalalib@indiana.edu

David Leake  
Indiana University  
leake@indiana.edu

Najmeh Forouzandehmehr  
University of Houston  
nforouzandehmehr2@uh.edu

Abstract
The ability of case-based reasoning systems to solve novel problems depends on their capability to adapt past solutions to new circumstances. However, acquiring the knowledge required for case adaptation is a classic challenge for CBR. This motivates the use of machine learning to generate adaptation knowledge. Much adaptation learning research has studied the case difference heuristic (CDH) approach, which generates adaptation rules from pairs of cases in the case base by ascribing observed differences in case solutions to the differences in the problems they solve, to generate rules for adapting similar problem differences. Extensive research has successfully applied the CDH approach to adaptation rule learning for case-based classification (numerical prediction) tasks. However, classification tasks have been outside of its scope. The work presented in this paper addresses that gap by extending CDH-based learning of adaptation rules to apply to cases with categorical features and solutions. It presents the generalized case value heuristic (GCDH) to assess case and solution differences and applies it in an ensemble-based case-based classification method, ensembles of adaptations for classification (EAC), built on the authors’ previous work on ensembles of adaptations for regression (EAR). Experimental results support the effectiveness of EAC.

1 Introduction
Case-based reasoning (CBR) systems (e.g., López de Mantaras et al. [2005]) solve problems by retrieving prior cases and adapting their solutions to fit new circumstances. The flexibility of CBR systems to address novel problems derives from their case adaptation knowledge. However, endowing CBR systems with the required knowledge has proven difficult; addressing the case adaptation problem remains a key challenge for CBR. One approach to addressing this challenge has been the application of machine learning. A wide range of methods has been investigated. For example, McSherry [1998] proposes generating adaptations from selected case triples, Leake et al. [1996] propose applying CBR to case adaptation, Patterson et al. [2002] propose learning adaptations by linear regression on the differences between the top nearest neighbors of an input query, Policastro et al. [2008] propose training a set of regression learners for estimating initial solutions and adjusting/combining their values, Craw et al. [2001] and Jarmulak et al. [2001] propose subdividing the case base into a small set of probe cases, used to retrieve cases from which to generate adaptation rules, and Wiratunga et al. [2006] propose applying tree learning algorithms. Interactive learning methods have been proposed as well (e.g., Cordier et al. [2008]).

An especially rich research current has applied methods based on the case difference heuristic (CDH), first proposed by Hanney and Keane [1996]. CDH methods learn case adaptation rules from knowledge already contained in the CBR system’s case base. Given a pair of cases, the case difference heuristic approach compares the problems they address, and ascribes the difference in their solutions to the differences in those problems. It then generates a new adaptation rule, used to adapt retrieved cases when their problems and the input problem have similar differences. The rule adjusts the retrieved case’s solution by the solution difference observed when the rule was generated.

The case difference heuristic has been widely and successfully applied to regression (numerical prediction) tasks (e.g., [McSherry, 1998; McDonnell and Cunningham, 2006; Jalali and Leake, 2013]), and some research has pursued difference-based methods for structured cases [Badra et al., 2009; Muller and Bergmann, 2015]. Previous CDH work relied on exact matching for comparing non-numeric problem features. In that setting, it is relatively straightforward to calculate the differences between problem descriptions and between solutions. This paper presents research by Jalali et al. [2016] on extending the CDH approach to categorization tasks and to problem descriptions in which non-numeric features are important, as often holds for categorization problems. It includes two main contributions. The first is the Generalized Case Difference Heuristic (GCDH), which determines differences using the heterogeneous value difference metric defined by the authors based on the classic value distance metric [Stanfill and Waltz, 1986]. Because the value-distance metric is a “knowledge light” approach, requiring only statistical information, GCDH enables generating adaptation rules with no additional domain knowledge.
Automatically generating adaptation rules from the case base transforms the classic adaptation setting for CBR: Rather than relying on a small set of hand-crafted rules, a CBR system has access to many rules, but their accuracy may not be assured. The second main contribution of the work addresses that problem with an ensemble-based method for applying the learned adaptation rules, Ensemble of Adaptations for Classification (EAC). EAC generalizes our prior work on ensemble-based adaptation, Ensembles of Adaptations for Regression (EAR) [Jalali and Leake, 2013; 2015b]. An EAR implementation is available as a Weka plugin [Jalali and Leake, 2015a]. This paper presents an evaluation of both the GCDH approach and EAC. Results support the accuracy benefit of EAC.

2 The Case Difference Heuristic Approach

The case Difference Heuristic (CDH) generates adaptation rules from the case base by comparing pairs of cases. If their solutions differ, CDH ascribes the solution difference to the differences in the problems each addresses. It generates a rule, applicable when an input problem and retrieved case have similar problem differences, to adapt the retrieved solution by the corresponding solution difference. As an extremely simplified example, in the rent prediction domain, if apartment A has one more bedroom than similar apartment B, and the rent for A is $100 more, the CDH approach might generate: If an apartment has one bedroom fewer than the most similar prior case R, subtract $100 from R’s rent.

Generating rules with the CDH method depends on three choices: (1) how to characterize the problem differences, (2) what solution modification should be generated from an observed solution difference (possibilities for numerical prediction include, e.g., adjusting by a fixed value or fixed percent, or applying a more complex formula), and (3) how to choose the pairs of cases from which to generate rules. The work in this paper focuses on (1) and (3). Applying the resulting rules also depends on an additional choice, (4) how to select the case(s) to adapt.

Our prior work on Ensemble of Adaptations for Regression (EAR) focused on (3) and (4). For (3) we tested three alternatives: Local cases-Local neighbors builds adaptation rules by comparing cases in vicinity of the query; Global cases-Local neighbors builds rules by comparing every case in the case base with a top few nearest neighbors; and Global cases-Global neighbors builds rules by comparing every case in the case base with all other cases in the case base. EAC uses Global cases-Local neighbors.

2.1 The Need to Address Categorical Features

The case difference heuristic has proven effective for regression tasks. However, problems arise for domains with categorical input features or target values. When assessing similarity of categorical input features, prior CDH approaches resort to exact matching. The lack of richer handling of categorical input features can limit effectiveness even for regression tasks. For example, in the apartment rental domain, rents may depend on the type of flooring, e.g., hardwood, laminate, vinyl, or carpet. A system relying on exact matching for case similarity cannot differentiate between four apartments that have identical features except for each having different flooring, even though an apartment with hardwood flooring might be more expensive. For problems with categorical target values (i.e., classification tasks), prior CDH methods provide no guidance.

Determining similarity for categorical features is also important for selecting adaptation rules to apply. For example, consider adapting the rent from a retrieved case with vinyl flooring to an apartment with hardwood flooring, with two adaptation rules available, neither an exact match:

Rule1: If input flooring is carpet and R’s flooring type is hardwood, then adapt price by subtracting $300.

Rule2: If input flooring is laminate and R’s flooring type is hardwood, then adapt price by subtracting $200.

Here, Rule2 is intuitively more appropriate to address the differences between the input query and the retrieved case compared to Rule1, and should be selected.

3 The Generalized Case Difference Heuristic Approach

The Generalized Case Difference Heuristic (GCDH) approach [Jalali et al., 2016] extends the Case Difference Heuristic (CDH) to handle categorical features and target values. It calculates the differences by the heterogeneous value difference metric. Depending on whether case or rule differences are being calculated, the heterogeneous value difference metric uses the Case or Rule Value Distance Heuristic Metric (CaseVDHM/RuleVDHM). In this section we introduce the heterogeneous value difference metric by focusing mainly on its application on case retrieval (i.e., caseVDHM). However, with an appropriate rule representation, the process for RuleVDHM is very similar to CaseVDHM.

Representing problem differences: As in other CDH research, we treat input problem descriptions as represented by feature vectors, with each component of the vector representing the difference in that component. From each pair of cases, GCDH generates a single rule in the form of Eq. 1:

\[(\Delta_{f_1}, ..., \Delta_{f_k}) \Rightarrow \Delta_t\]  (1)

In Eq. 1, \(f_1, ..., f_k, \) and \(t\) represent the input and target features in the underlying domain respectively, and \(\Delta_x\) represents the distance between the given pair of values for feature \(x,\) if \(x\) is numeric, and the ordered pair of given values of feature \(x\) (e.g., \((t_i, t_j)\)) if \(x\) is a categorical feature.

Source case selection: Source case retrieval is an integral component of GCDH. During rule generation, GCDH determines the cases from which adaptation rules should be generated; during prediction, it determines the cases from which the solution is generated. GCDH calculates the distance between two cases \(c_1\) and \(c_2\) as:

\[\text{distance}(c_1, c_2)\]
follows (for space we abbreviate target classes in the domain. Also, diff
sequent pair does not match the value of the selected source
rules by removing those whose first element of the con-
filters out the non-applicable rules from the set of gen-
Rules
are the building blocks of the adaptation rules, and
distance between pairs of values of categorical features that
"Global cases-Local neighbors" alternative for learning the
rule retrieval methods respectively.
The
CaseHVDM
Algorithm 1 summarizes EAC's value estimation process.
4.1 Estimating the Target Value with EAC

\[
\text{casediff}(c_1, c_2) \equiv \sum_{i=1}^{n} |\text{CaseVDHM}_i(c_{1,i}, c_{2,i})|^2
\]

The Case Value Distance Heuristic Metric CaseVDHM calculates the distance between values a and b of feature f as follows (for space we abbreviate CaseVDHM as CVDHM):

\[
\text{CVDHM}_f(a, b) = \begin{cases} 
1 & \text{if } a \text{ or } b \text{ is unknown} \\
1 & \text{if } f \text{ is categorical and } a \text{ or } b \text{ is not observed} \\
\text{vd}_{f}(a, b) & \text{if } f \text{ is categorical} \\
\min(1, \text{diff}_f(a, b)) & \text{if } f \text{ is numeric}
\end{cases}
\]

Here \(\text{vd}_f\) is the Value Difference Metric VDM. [Stanfill and Waltz, 1986], which calculates the distance between values a and b of the categorical feature f as:

\[
\text{vd}_{f}(a, b) \equiv \sum_{c=1}^{C} \left| \frac{N_{f,a,c}}{N_{f,a}} - \frac{N_{f,b,c}}{N_{f,b}} \right|
\]

In Eq. 4, \(N_{f,a,c}\) is the number of cases that have value a for feature f, \(N_{f,a,c}\) is the number of cases that have value a for feature f and target value c, and C is the number of distinct target classes in the domain. Also, \(\text{diff}_f(a, b)\) is calculated as:

\[
\text{diff}_f(a, b) = \frac{|a - b|}{4\sigma_f}
\]

Where \(\sigma_f\) is the standard deviation of feature f. If the values of feature f follow a normal distribution then 95% of them are expected to fall within a \(4\delta\) distance range of each other.

4 Ensemble of Adaptations for Classification

Ensemble of Adaptations for Classification is an extension to Ensemble of Adaptations for Regression (EAR), applying the EAR approach to domains with categorical features and target values. The EAR configuration questions on source case selection and selecting cases from which to generate adaptations, discussed in Section 2, apply to EAC as well. In addition, EAC requires choosing the cases used to train CaseVDHM and RuleVDHM.

EAC trains CaseVDHM by using every case in the case base. To train RuleVDHM it only uses the generated rules, comparing every case with its top nearest neighbors.

4.1 Estimating the Target Value with EAC

Algorithm 1 summarizes EAC's value estimation process. The CaseHVDM function is passed as an input to the al-
gorithm. NeighborhoodSelector and RankRules are the case and
rule retrieval methods respectively. HVDMRuleTrain is the
"Global cases-Local neighbors" alternative for learning the
distance between pairs of values of categorical features that
are the building blocks of the adaptation rules, and Filter-
Rules filters out the non-applicable rules from the set of gen-
erated rules by removing those whose first element of the con-
squential pair does not match the value of the selected source

case. MajorityRuleVote returns the majority value of the sec-
ond element of the consequent pair of the retrieved rules and
MajorityVote returns the majority value of the adjusted source
cases' values. Ties are broken arbitrarily.

4.2 EAC's Process

Fig. 1 illustrates EAC's process. At the fist step, EAC trains CaseVDHM for case retrieval. Next, it uses CaseVDHM to select the nearest neighbors of the input query and the cases from which adaptation rules will be generated. After generating the adaptation rules, EAC trains RuleVDHM to learn the distance between pairs of pairs of categorical val-
ues. The newly trained RuleVDHM along with the differ-
ces between the input query and the source cases are used to select the rules that should be applied. The value of each source case is adapted by applying an ensemble of adaptation rules and the final estimation will be generated by combining the adjusted values of source cases using majority voting.

5 Experiments

We conducted experiments to study the accuracy of EAC and how it is affected by EAC's ensemble-based approach and the training to test data size ratio.

5.1 Experimental Design

EAC was implemented using Spark MLlib [Meng et al., 2015], Apache Spark’s scalable machine learning library. It provides classification methods, grid search for tuning pa-
rameters, and cross validation for testing. Experiments test
Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)

Table 1: Estimation error of EAC, kNN-CVDHM, RF, and NB methods in four sample domains

<table>
<thead>
<tr>
<th>Domain name</th>
<th>EAC</th>
<th>RF</th>
<th>kNN-CVDHM</th>
<th>NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balance</td>
<td>15.98%</td>
<td>16.76%</td>
<td>25.26%</td>
<td>44.31%</td>
</tr>
<tr>
<td>Bankruptcy</td>
<td>1.35%</td>
<td>2.05%</td>
<td>3.43%</td>
<td>13.51%</td>
</tr>
<tr>
<td>Car</td>
<td>3.95%</td>
<td>6.25%</td>
<td>6.50%</td>
<td>30.71%</td>
</tr>
<tr>
<td>Credit</td>
<td>15.64%</td>
<td>17.10%</td>
<td>18.11%</td>
<td>30.81%</td>
</tr>
</tbody>
</table>

The experiments compare the percent accuracy of EAC and three baseline classification methods: Random Forest (RF), Naive Bayes (NB), and kNN, using Apache Spark implementations of Random Forest and Naive Bayes. The kNN version used, kNN-CVDHM, is an implementation of kNN with CaseVDHM as the similarity measure. kNN − CVDHM is a special case of EAC in which no adaptation rule is applied.

5.2 Experimental Results

Comparative accuracy

Table 1 shows the accuracy of EAC and the other classification methods on the four sample domains from Section 5.1.

As shown, EAC achieves the best accuracy and Random Forest achieves second best in all domains. EAC shows 5%, 34%, 37%, and 8% decreases in estimation error over Random Forest in the Balance, Bankruptcy, Car, and Credit domains respectively.

EAC shows 37%, 60%, 38%, and 14% improvement in estimation error over kNN − CVDHM in Balance, Bankruptcy, Car, and Credit domains respectively. For domains with only categorical features (i.e., all domains except Credit), we note EAC’s gain over kNN − CVDHM is higher when the curse of dimensionality is more severe. For the Balance and Car domains, on average there is one case per n-dimensional unit of the problem specification space while this decreases to 0.3 for Bankruptcy domain. However, the possible effect of dimensionality needs more investigation before a general conclusion will be possible.

Effect of applying ensembles of adaptations

To assess the effect of using ensembles of adaptations, we implemented an ablated version of EAC named EAC-a, which applies a single adaptation to adjust the source case values. Note that EAC-a only limits the number of adaptations to be applied per source case and not the number of source cases used for building the solution. Figure 2 depicts the percentage decrease in estimation error for EAC and EAC-a over kNN − CVDHM, with identical tunings of EAC-a and EAC. In all test domains EAC outperforms EAC-a. For the Car domain, applying a single adaptation (EAC-a), rather than an ensemble of adaptations (EAC), results in average accuracy slightly lower than that of kNN − CVDHM, which uses no adaptations.

The effect of Test/Train Ratio on EAC’s performance

Figure 3 shows the percent improvement of EAC, kNN − CVDHM, and RF over NB in the car evaluation domain for different ratios of training vs test data size. As the training versus test size ratio increases, EAC best benefits from the training data, with kNN − CVDHM second, but fairly close to Random Forest. For smaller training sets, Random Forest can outperform EAC (when training size ratio is 10% and 30%). Trends in other domains were similar.

6 Conclusions and Future Research

This paper presented Ensemble of Adaptations for Classification, a method for automatic adaptation learning in domains with categorical input features and target values. EAC applies the Generalized Case Difference Heuristic approach, an extension to the case difference heuristic.

Future work includes exploring other categorical similarity measures and other variations of EAC’s rule generation. To help alleviate the processing cost of EAC, we are applying ideas developed in BEAR [Jalali and Leake, 2015b] for increased scalability of EAC for large case bases. As part of this goal, we are developing Locality Sensitive Hashing methods for domains with categorical input features [Jalali and Leake, 2017].

Acknowledgments

This paper summarizes results from Jalali et al. [2016], IC-CBR 2016 Best Paper, which includes additional detail.
References


