Adaptive Class Association Rule Mining for Human Activity Recognition

Martin Atzmueller[1], Mark Kibanov[1], Naveed Hayat[1],
Matthias Trojahn[2], and Dennis Kroll[3]

[1] University of Kassel, Research Center for Information System Design
Knowledge and Data Engineering Group, Kassel, Germany
`{atzmueller, kibanov, hayat}@cs.uni-kassel.de`

[2] Volkswagen AG, Wolfsburg, Germany
`matthias.trojahn@volkswagen.de`

[3] University of Kassel, Research Center for Information System Design
Chair for Communication Technology, Kassel, Germany
`dennis.kroll@comtec.eecs.uni-kassel.de`

Abstract. The analysis of human activity data is an important research area in the context of ubiquitous and social environments. Using sensor data obtained by mobile devices, e. g., utilizing accelerometer sensors contained in mobile phones, behavioral patterns and models can then be obtained. However, the utilized models are often not simple to interpret by humans in order to facilitate assessment, evaluation and validation, e. g., in computational social science or in medical contexts. In this paper, we propose a novel approach for generating interpretable rule sets for classification: We present an adaptive framework for mining class association rules using subgroup discovery, and analyze different techniques for obtaining the final classifier. The approach is investigated in the context of human activity recognition. For our evaluation, we apply real-world activity data collected using mobile phone sensors.

1 Introduction

With more and more ubiquitous devices emerging in our daily lives, sensor data capturing human activities is becoming a universal data source for the analysis of human behavioral patterns, and for building according models. However, often such models are either "black-box" models like neural networks, or are rather complex, e. g., in the case of random forests or large decision trees. Rule-based models can then often provide simpler models with comparable accuracy, estimated using quality measures [6, 7], in order to facilitate human interpretation.

In this paper, we propose a novel approach for *class association rule mining* using *subgroup discovery*. We present an *adaptive framework* for mining such rules, and demonstrate the effectiveness of the proposed approach using real-world activity data collected using mobile phone sensors. Specifically, we focus on *activity recognition*, as a prominent research field with respect to the classification of human activities.

Class association rules are special association rules with a fixed class attribute in the rule consequent. In order to mine such rules, we apply subgroup discovery [4,42] – an exploratory approach for discovering interesting subgroups defined by a *description*, e. g., a conjunction of attribute–value pairs (i. e., a typical rule body) with respect to a binary target concept. In the case of class association rules, the respective class can be defined as the target concept (i. e., the rule head). Then, subgroup discovery can be adapted as a rule generator for class association rule mining. As we will discuss below, there are further adaptations for mining the final rule set, which we integrate into a comprehensive framework for adaptive class association rule mining.

Our contribution can be summarized as follows:

1. We adapt subgroup discovery to class association rule mining, and embed it into an adaptive approach for obtaining a rule set that aims to target a simple rule base with an adequate level of predictive power, i. e., combining simplicity and accuracy.
2. For constructing the rule base, we utilize standard methods of rule selection and evaluation, and demonstrate the integration into our framework.
3. We provide an evaluation using real-world activity data obtained by mobile phone sensors, and demonstrate the effectiveness of our approach by a comparison with typical descriptive models, i. e., using Ripper as a rule-based baseline, and C4.5 as a decision tree classifier.

The rest of the paper is structured as follows: Section 2 discusses related work. Then, Section 3 introduces the necessary background. After that, Section 4 introduces the adaptive framework for class association rule mining. In Section 5 we describe the applied dataset. Next, Section 6 presents the results of our experiments and discusses them in detail. Finally, Section 7 concludes with a summary and provides interesting options for future work.

2 Related Work

Below, we discuss related work concerning general approaches for the classification of sensor data, subgroup discovery and associative classification.

2.1 Classification and Sensor Data

Classification of activities based on sensor data is a prominent research area. Several authors investigated the topic using wearable sensors, e. g., as also integrated into mobile phones. These sensors can be attached to parts of the body like arms, legs or the hip. The first works in this regard were already done at the end of the 1990s [30]. In the research of Foerster et. al. [23] 24 participants wore sensors on sternum, wrist, thigh and the lower leg. Nine activities were then replicated. Also, Bao and Intille [16] asked 20 subjects to perform some everyday activities while wearing five biaxial accelerometers on different parts of the body.

Fabian et al. [21] developed a real-time mobile system to recognize six different activities in both standing and sitting positions. Therefore three motion band devices were attached to the wrist, hip and the dominant ankle of the participants. These devices contained an accelerometer, a magnetometer and a gyroscope. While the training was done offline on a Desktop PC, the following recognition process was done in real time with a smartphone collecting the sensor data from the attached motion bands.

In this paper, we consider the field of wearable sensors, specifically on those embedded in mobile phones, focusing on the accelerometer: Kwapisz et al. [29], for example, collected and labeled data from 29 users and tried to classify six basic activities (like standing or walking). Reddy et al. [39] considered the problem of usage of mobile phones to determine transportation mode (such as walking, biking, or in motorized transport) and used additionally GSM receiver of the device. Berthold et al. [17] presented ActiServ — an architecture which creates an evolving activity classification system using feedback from the user community. Yang [43] proposed a physical activity diary based on automatic sensor data classification to use in mobile healthcare and further applications (currently such applications emerge, e. g., Apple ResearchKit [4]).

In contrast to most of the presented works, we concentrate on some special activities, some of which assume active interaction with mobile phones. We also define a group of disrupt activities — activities which are similar to a usual activity – to examine if the presented classifier may recognize small differences in activities. Furthermore, we consider up to 8 sensors for improving activity recognition. In contrast, related work discussed above only uses accelerometer or in a few cases a limited number of two or three sensors.

2.2 Subgroup Discovery

Subgroup discovery [2,4,15,27,42] has been established as a general and broadly applicable technique for descriptive and exploratory data mining: It aims at identifying descriptions of subsets of a dataset that show an interesting behavior with respect to certain interestingness criteria, formalized by a quality function, e. g., [4,25,27].

Overall, subgroup discovery and analytics are important tools for descriptive data mining: They can be applied, for example, for obtaining an overview on the relations in the data, for automatic hypotheses generation, and for data exploration. Prominent application examples include knowledge discovery in medical, technical, and social domains, e.g., [3,10,14,15,24,31,37]. Subgroup discovery is especially suited for identifying local patterns in the data, that is, *nuggets* that hold for specific subsets: It can uncover hidden relations captured in small subgroups, for which variables are only significantly correlated in these subgroups. Typically, the discovered patterns are especially easy to interpret by the users and domain experts, cf. [11,24,25].

[4] http://researchkit.org

Standard subgroup discovery approaches commonly focus on a *single* target concept as the property of interest [25, 27, 31], while the quality function framework also enables *multi-target concepts*, e. g., [12,28]. Furthermore, more complex target properties [20, 32] can be formalized as *exceptional models*, cf. [32]. In the case of a binary target variable, the share in a subgroup can be compared to the share in the dataset in order to detect deviations in (large) subgroups. This is also the approach considered in this paper, where we focus on a specific *class* (a set of classes, respectively) as the target concept(s). In addition to basic subgroup discovery which aims at providing the obtained subgroups in exploratory and descriptive fashion, we embed subgroup discovery as the basis of our rule generation approach. We apply an adaptive method that aims to generate rules with increasing complexity (and accuracy) based on a performance estimate of the current subgroup set. In addition, we apply a rule selection strategy in order to obtain the final set of class association rules for classification.

2.3 Associative Classification

Associative classification approaches integrate association rule mining and classification strategies. Thabtah [41] provides a survey on the field. This includes the first approach by Liu et al. [35] for class association rule mining, which includes association rule mining and subsequent rule selection in the CBA algorithm. It applies a covering strategy, selecting rules one by one, minimizing the total error. Alternative approaches include the CMAR algorithm by Li et al. [34] which also applies covering, but allows for multiple rules to cover an instance. The CPAR algorithm by Yin and Han [44] integrates rule mining and selection, and achieves comparable accuracy compared to CBA and CMAR. In addition to the rule mining and selection techniques, there are several strategies for the final decision of how to combine the rules for the classification ("voting" of the matching rules), e. g., [40].

Compared to the approaches discussed above, our proposed approach applies subgroup discovery for class association rule mining, which allows for suitable selection of a (complex) quality function for mining the rules, in constrast to the (simple) confidence/support-based approaches applied by association rule mining approaches. Then, for example, significance criteria can be simply embedded. Furthermore, the presented approach applies an adaptive strategy for balancing rule complexity (size) with predictive accuracy by applying a ruleset assessment function, in addition to the rule selection function. However, our framework is general in that respect, that we do not enforce a specific strategy. Instead, this decision can be configured by the specific implementation of the framework. In our implementation throughout this paper, for example, we follow the rule selection strategy of CBA; the ruleset assessment is done by a median-based ranking of the according confidences of the rules, i. e., estimated by the respective shares of the class contained in the subgroups covered by the respective rules. We will describe these concepts below in more detail.

3 Background

Below, we first introduce some basic notation. After that, we summarize basics on subgroup discovery, before we sketch how to mine class association rules using subgroup discovery.

3.1 Basic Notation

Formally, a *database* $DB = (I, A)$ is given by a set of individuals I and a set of attributes A. A *selector* or *basic pattern* $sel_{a_i=v_j}$ is a Boolean function $I \to \{0, 1\}$ that is true if the value of attribute $a_i \in A$ is equal to v_j for the respective individual. The set of all basic patterns is denoted by S.

For a numeric attribute a_{num} selectors $sel_{a_{num} \in [min_j; max_j]}$ can be defined analogously for each interval $[min_j; max_j]$ in the domain of a_{num}. The Boolean function is then set to true if the value of the respective attribute a_{num} is within the respective interval.

3.2 Patterns and Subgroups

Basic elements used in subgroup discovery are patterns and subgroups. Intuitively, a *pattern* describes a *subgroup*, i. e., the subgroup consists of instances that are covered by the respective pattern. It is easy to see, that a pattern describes a fixed set of instances (subgroup), while a subgroup can also be described by different patterns, if there are different options for covering the subgroup' instances. In the following, we define these concepts more formally.

Definition 1. *A* subgroup description *or (complex)* pattern *sd is given by a set of basic patterns $sd = \{sel_1, \ldots, sel_l\}$, where $sel_i \in S$, which is interpreted as a conjunction, i.e., $sd(I) = sel_1 \wedge \ldots \wedge sel_l$, with $length(sd) = l$.*

Without loss of generality, we focus on a conjunctive pattern language using nominal attribute–value pairs as defined above in this paper; internal disjunctions can also be generated by appropriate attribute–value construction methods, if necessary.

Definition 2. *A* subgroup (extension)

$$sg_{sd} := ext(sd) := \{i \in I | sd(i) = true\}$$

is the set of all individuals which are covered by the pattern sd.

As search space for subgroup discovery the set of all possible patterns 2^S is used, that is, all combinations of the basic patterns contained in S. Then, appropriate efficient algorithms, e. g., [8, 13, 33] can be applied.

3.3 Interestingness of a Pattern

The interestingness of a pattern is determined by a quality function, which is selected according to the analysis task.

Definition 3. *A* quality function $q \colon 2^S \to \mathbb{R}$ *maps every pattern in the search space to a real number that reflects the interestingness of a pattern (or the extension of the pattern, respectively).*

While a large number of quality functions has been proposed in literature, many quality functions for a single target concept, e. g., in the binary or numerical case, trade-off the size $n = |ext(sd)|$ of a subgroup and the deviation $t_{sd} - t_0$, where t_{sd} is the average value of a given target concept in the subgroup identified by the pattern sd and t_0 the average value of the target concept in the general population. In the binary case, the averages relate to the *share* of the target concept. Thus, typical quality functions are of the form

$$q_a(sd) = n^a \cdot (t_{sd} - t_0),\ a \in [0; 1] . \tag{1}$$

For binary target concepts, this includes, for example, the *weighted relative accuracy* for the size parameter $a = 1$ or a simplified binomial function, for $a = 0.5$. An extension to a target concept defined by a set of variables can be defined similarly, by extending common statistical tests.

While a quality function provides a *ranking* of the discovered subgroup patterns, often also a statistical assessment of the patterns is useful in data exploration. Quality functions that directly apply a statistical test, for example, the Chi-Square quality function, e. g., [4] provide a p-Value for simple interpretation. However, the Chi-Square quality function estimates deviations in two directions. An alternative, which can also be directly mapped to a p-Value is given by the *adjusted residual* quality function q_r, since the values of q_r follow a large standard normal distribution, cf. [1]:

$$q_r = n(p - p_0) \cdot \frac{1}{\sqrt{np_0(1 - p_0)(1 - \frac{n}{N})}} \tag{2}$$

The result of top-k subgroup discovery is the set of the k patterns $sd_1, \ldots, sd_k$, where $sd_i \in 2^S$ with the highest interestingness according to the applied quality function. A subgroup discovery task can now be specified by a 5-tuple:

$$(DB, c, S, q, k) .$$

We focus on the case of a binary target concept $c \colon I \to \Re$ specifying the property of interest: In the context of class assocation rule mining, it maps each instance in the dataset to a target value c corresponding to the respective class of the instance. The search space 2^S is defined by set of basic patterns S.

Furthermore, we consider additional constraints with respect to the *complexity* of the patterns. We can restrict the length l of these descriptions to a certain maximal value, e. g., with length $l = 1$ we only consider subgroup descriptions containing one selector, with length $l = 2$ we consider a conjunction of two selectors etc. Then, the complexity of the discovered patterns can also be adaptively adjusted as described in Section 4.

3.4 Subgroup Discovery for Mining Class Association Rules

For mining class association rules, we apply subgroup discovery, such that for every class $c \in S$, we create an according target concept c. Then, we discover a set of the top-k patterns $CAR_c = \{sd_1^c, sd_2^c, \ldots, sd_k^c\}$ for each target concept. It is easy to see, that a subgroup pattern directly corresponds to a class association rule - the head of the rule is given by the target concept, while the body of the rule is given by the specific subgroup description. Then, these rules can be applied for building the classifier. For that, a specific *rule selection* strategy needs to be applied, after the total set of class association rules has been determined. It usually aims at selecting the subset with the best predictive power, e. g., using one of the algorithms discussed above in Section 2.

When applying the model, different *rule combination* strategies can be used, e. g., taking the best rule, or aggregating the votes of the individual matching rules, cf. [40]. Basically, for each rule r that matches an instance $i \in I$ that we want to classify, we can combine the different classifications of the individual r_i in order to combine the final classification. The *best rule* strategy just selects the rule with the highest confidence (and its respective classification). In addition, we can apply voting methods for obtaining the final classification, cf. [40], i. e., for combining individual predictions as votes for the individual classification. Essentially, for classifying an individual (instance) $i \in I$, this works as follows:

$$\mathit{class}(i) = \arg\max_{c_i \in C} \sum_{r \in R_i} \mathit{weight}(r)\,, \tag{3}$$

where R_i is the subset of rules matching instance $i \in I$ of class c_i, and $C \subseteq S$ denotes the set of available classes in our dataset.

The weight of a rule $\mathit{weight}(r)$ depends on the chosen weighting method. Following [40], we applied the *unweighted* strategy, where $\mathit{weight}_U(r) = 1$ for all rules r, and the laplacian weight strategy $\mathit{weight}_L(r) = \mathit{Laplace}(r)$, where the laplacian weight is determined according to the Laplace correction [18] to the estimated class probabilities of the applied dataset:

$$\mathit{Laplace}(r) = \frac{p_i^r + 1}{\sum_{c_j \in C} p_j^r + |C|}\,, \tag{4}$$

where p_j^r (and p_i^r) are the numbers of covered examples by rule r that belong to the respective classes c_j (and c_i of the rule, respectively).

4 An Adaptive Framework for Class Association Rule Mining

In this section, we provide an overview on the proposed approach presenting our novel framework Carma, an Adaptive Framework for Class Association Rule Mining, and provide examples of its instantiation in Section 6. For our adaptive framework, we distinguish two phases: The *learning phase* that constructs the model, and the *classification phase* that applies the model.

Learning: Model Construction For the construction of the model, we apply the steps described in Algorithm 1. Basically, CARMA starts with discovering class association rules for each class c contained in the dataset. Using subgroup discovery (line 5, calling procedure *SubgroupDiscovery* that needs to be instantiated with an appropriate subgroup discovery algorithm), we collect a set of class association rules for the specific class, considering a maximal length of the concerned patterns. After that, we apply a boolean *ruleset assessment* function a (line 6) in order to check, if the quality of the ruleset is good enough. If the outcome of this test is positive, we continue with the next class (line 10). Otherwise, we increase the maximal *length* of a rule (up to a certain user-definable threshold $\mathcal{T}$, line 12). After the final set of all class association rules for all classes has been determined, we apply the *rule selection function* r (line 14) in order to obtain a set of class association rules that optimizes predictive power on the trainingset. That is, the rule selection function aims to estimate classification error and should select the rules according to coverage and accuracy of the rules on the trainingset.

Algorithm 1 CARMA

Require: Set of classes C, k specifying the number of top-k patterns, maxlength $\mathcal{T}$ denoting the maximal possible length of a subgroup pattern, quality function q, ruleset assessment function a, rule selection function r.

1: Patterns $P = \emptyset$
2: **for all** $c \in C$ **do**
3:     Current length threshold $\mathit{length} = 1$
4:     **while** true **do**
5:         Obtain candidate patterns CP by $CP = \mathit{SubgroupDiscovery}(DB, c, S, q, k, \mathcal{T})$
6:         **if** Current candidate patterns are good enough, i. e., $a(CP) = \mathit{true}$ **then**
7:             $P = P \cup CP$
8:             break
9:         **else if** $\mathit{length} > \mathcal{T}$ **then**
10:             break
11:         **else**
12:             $\mathit{length} = \mathit{length} + 1$
13: Add a default pattern (rule) for the most frequent class to P
14: Apply rule selection function: $P = r(P)$
15: **return** P {Model, consisting of the result set of rules}

Classification For the classification phase, we apply all the rules contained in the model P. For aggregating the predictions of the (matching) rules for an individual (instance) $i \in I$, and for obtaining the final classification, we apply a specific *rule combination* strategy, see Section 3 for examples.

5 Dataset

We collected a dataset containing a diverse set of activities (classes) split into two categories: (1) Activities which demand the direct usage of the device , e. g., holding the device close to the ear, or putting the device in a specific place, and (2) typical walking activities, e. g., walking slowly or normally. We defined five scenarios that consist of sets of different activities. While doing these activities the person used a smartphone with a running application. This application recorded the sensor data. The persons used the smartphone actively (e. g., putting device in the pocket) or passively (e. g., while walking). Another smartphone was used to record the exact start and finish time of each activity. 39 test persons of different sex and age repeated each scenario six times. The resulting dataset consists of a total of 3077 valid single activities. Table 1 shows an overview on the dataset, specific activities and class distributions in detail.

Table 1. Activity dataset – Overview: Description of the individual activities (classes), body position, device context, number of instances (samples) for each activity/class.

ID	**Description** Activity/Class	**Body** Position	**Device** Usage	**No. of** Samples
1	Put device in right trousers pocket	Sit	Yes	54
2	Put device in right trousers pocket	Stand	Yes	290
3	Put device in shirt pocket	Sit	Yes	54
4	Put device in shirt pocket	Stand	Yes	162
5	Take device from right trousers pocket	Sit	Yes	54
6	Take device from right trousers pocket	Stand	Yes	290
7	Take device from shirt pocket	Sit	Yes	54
8	Take device from shirt pocket	Stand	Yes	162
9	Put device on the table	Sit	Yes	55
10	Put device on the table	Stand	Yes	272
11	Take device from the table	Sit	Yes	55
12	Take device from the table	Stand	Yes	272
13	Give device to another person	Sit	Yes	109
14	Give device to another person	Stand	Yes	163
15	Take device from another person	Sit	Yes	55
16	Take device from another person	Stand	Yes	217
17	Hold device near the ear	Stand	Yes	217
18	Take device away from the ear	Stand	Yes	54
19	Walk slowly (device in hand)	—	No	54
20	Walk slowly (device near ear)	—	No	54
21	Walk normally (device in shirt pocket)	—	No	54
22	Walk normally (device in hand)	—	No	54
23	Walk normally (device near ear)	—	No	55
24	Walk normally (device in right trousers pocket)	—	No	55
25	Walk fast (device in hand)	—	No	54
26	Walk fast (device near ear)	—	No	54
27	Walk fast (device in right trousers pocket)	—	No	54

Table 2. Overview on the features generated using the collected sensor data.

Feature	Sensor
Average/Minimum/Maximum Value	All
Standard Deviation	All
Zero-Crossings	All Without Light and Proximity Sensors
75th Percentile	All Without Light and Proximity Sensors

Overall, we recorded data from eight different sensors, installed on Samsung Galaxy Nexus Device, particularly: (1) Accelerometer, (2) Magnetometer, (3) Gyroscope, (4) Light sensor, (5) Proximity sensor, (6) Rotation vector, (7) Gravity sensor, and (8) Linear acceleration. Using these, we created a set of features applying window-based techniques. A fixed window size of 1 second was used. This size was already proven to be efficient for walking activities [26]. We created 6 features per window and per sensor as described in Table 2. Zero-crossings describes the number of changes from positive to negative and negative to positive values, respectively. The 75th percentile represents the lowest value that is greater than or equal to 75% of the values. Other features were the calculated mean, min/max and standard deviation for the given window. The features were extracted for every axis of every sensor. The only exception were light and proximity sensors. Zero-crossings and the 75th percentile were not calculated for these sensors because of the nature of their returning values. Thus 4 features were obtained for both the light and proximity sensor and 18 for each of the others, resulting in a total set of 116 features. In order to use the features for class association rule mining, we employed the discretization technique by Fayad & Irani [22] for deriving according selectors.

6 Evaluation

Below, we compare an instantiation of the proposed Carma framework against two baselines: The Ripper algorithm [19] as a rule-based learner, and the C4.5 algorithm [38] for learning decision trees. For the subgroup discovery step in the Carma framework, we apply the BSD algorithm [33] using the implementation provided by the VIKAMINE system [9]. Further details are described below when we discuss the experimental setup and results.

As the basic evaluation measures, we consider (multi-class) *model accuracy* and *model complexity* with respect to activity recognition on the 116 features and 27 classes (shown in Table 1), cf. Section 5. Accuracy is defined as a portion of samples that were classified correctly. Furthermore, complexity relates to the size of a model using two parameters: the total number of rules contained in a rule-based model (also corresponding to the number of leaves in a decision tree), and its average complexity (i. e., for a decision tree the average length of path from a root to a leaf of a tree). All experiments were performed in a standard 10-fold cross-validation setting.

6.1 Baselines Results

We applied both *JRip* and *J48* algorithms as baseline methods. We compare results with the described approach and explore the influence of different parameters in terms of accuracy and model complexity.

Table 3. Baseline results using C4.5 (J48) and Ripper (JRip).

Algorithm	Accuracy	Complexity	
		No. of Rules	*Avg. Complexity*
J48	69.02%	1394	6.76
JRip	66.87%	176	3.40

Table 3 shows performance and complexity of the baseline algorithms. J48 showed a better performance but built a more complex model with 1394 rules and average rule complexity of 6.76. JRip's accuracy is 2% lower but the model is much smaller with only 176 rules and an average rule length of 3.40.

6.2 Results and Discussion

When applying the CARMA framework, we need to instantiate several components according to the analytical question. In the context of our experiments we instantiate these elements as follows:

- For the subgroup discovery algorithm, we selected the BSD algorithm [33].
- For the ruleset assesssment function, we just check, if the median of the rules' confidences is above a certain threshold τ_c. In our experiments, we applied a threshold $\tau_c = 0.5$.
- Furthermore, for the rule selection function, we apply an adaptation of the CBA algorithm [35].
- In addition to the basic CBA algorithm, we also implemented a variant, which we call CBA*. This algorithm ensures, that there is at least one rule for each class in the derived model, i. e., when estimating classification performance on the training set, it is checked that at least one rule for each class exists in the final classifier. We default to the rule with the highest confidence, if there is none contained in the initial model.
- Since we are interested in easily interpretable rules, we also selected the quality function q_r (adjusted residuals, described above) which directly maps to significance criteria.
- We opted for interpretable patterns with a maximal length of 7 conditions, and set the respective threshold $\mathcal{T} = 7$ accordingly.
- In the evaluation, we used three different *TopK* values: 100, 200 and 500.
- For the rule combination strategy, we experimented with four strategies: taking the best rule according to confidence and Laplace value, the unweighted voting strategy, and the weighted voting (Laplace) method (see Section 3).

Table 4. Evaluation results: The table shows accuracy and complexity of CARMA depending on different choices of k, the rule selection techniques *CBA* and *CBA**, and the following rule combination strategies: *UnweightedVote* (unweighted voting), *LaplaceVote* (voting using laplacian weights), *BestLaplace* (best rule using Laplace value), and *BestConfidence* (best rule according to rule confidence), cf. Section 3 for a detailed discussion.

TopK		**CBA**			**CBA***		
		Accuracy	*No. of Rules*	*Avg. Complexity*	*Accuracy*	*No. of Rules*	*Avg. Complexity*
100	***UnweightedVote***	67.14 %	347.2	2.79 ± 1.00	**67.31** %	345.3	2.82 ± 1.04
	LaplaceVote	66.47 %	347.1	2.80 ± 1.00	66.96 %	345.0	2.81 ± 1.04
	BestLaplace	59.60 %	349.4	2.81 ± 1.00	59.10 %	345.4	2.79 ± 0.98
	BestConfidence	63.31 %	349.8	2.82 ± 1.03	62.22 %	346.5	2.81 ± 1.01
200	*UnweightedVote*	67.82 %	424.9	2.91 ± 1.01	67.99 %	422.5	2.88 ± 1.00
	LaplaceVote	68.20 %	426.7	2.91 ± 1.02	**69.09** %	424.8	2.89 ± 1.00
	BestLaplace	59.45 %	421.8	2.90 ± 1.01	59.63 %	423.1	2.88 ± 1.01
	BestConfidence	64.75 %	424.7	2.87 ±1.01	64.93 %	422.8	2.89 ± 1.04
500	***UnweightedVote***	69.38 %	517.3	3.05 ± 0.97	**70.52 %**	522.3	3.05 ± 0.98
	LaplaceVote	69.95 %	518.4	3.05 ± 0.95	69.96 %	522.1	3.05 ± 0.96
	BestLaplace	60.95 %	518.3	3.06 ± 1.01	60.60 %	521.7	3.04 ± 0.97
	BestConfidence	66.80 %	525.4	3.06 ± 0.98	66.80 %	520.6	3.06 ± 0.97

Table 4 shows the results of our experiments. Overall, it is easy to see that the proposed approach outperforms the baselines both in accuracy as well as in complexity, i. e., an instantiation with the *UnweightedVote* or *LaplaceVote* functions and $k = 500$ outperforms even the C4.5 baseline clearly. If we especially concentrate on the complexity (or simplicity) of the model, we can observe that CARMA demonstrates its advantages since it clearly generates less complex models than the baselines with a comparable accuracy, e. g., C4.5. If we consider the Ripper algorithm, we can observe that it still has a better average complexity (i. e., lower average complexity of a rule) while it outperforms Ripper in terms of accuracy clearly.

Considering the voting functions, we observe that the functions (unweighted voting, and weighted Laplace) always outperform the rest. In our experiments, using larger values of k indicates a higher accuracy – here also the compexity (in the number of rules) can be tuned. We observe a slight trade-off between accuracy and complexity here. Basically, the parameter k seems to have an influence on the complexity, while the remaining instantiations do not seem to have a strong influence. This can be explained by the fact, that the *model generation* phase is mainly dependent on k (and the maximum length of the patterns) but not on the applied *voting method*. *CBA* and *CBA** seem quite close in terms of accuracy and complexity, while we can observe a slight improvement for CBA*. In empirical evaluations it turned out that the difference between *CBA** and *CBA* was even more pronounced for lower numbers of k, leading to slightly better models for *CBA**. However, for our parameter selection, we do not see strong improvements of *CBA** compared to *CBA*.

In summary, the proposed framework always provides a more compact model than the baseline algorithms concerning rule complexity, with simple rules such as: *IF* $minProx = (0.5 - 3] \wedge minMagnetY > 34 \wedge zeroCrossAccelX = (0.5 - 1.5]$ *THEN Class =Hold device near the ear*. In our experiments, it is at least in the same range or even better than the baselines concerning accuracy. In particular, considering the best parameter instantiations, the proposed approach is able to outperform both baselines concerning the accuracy (see Figures 1-2).

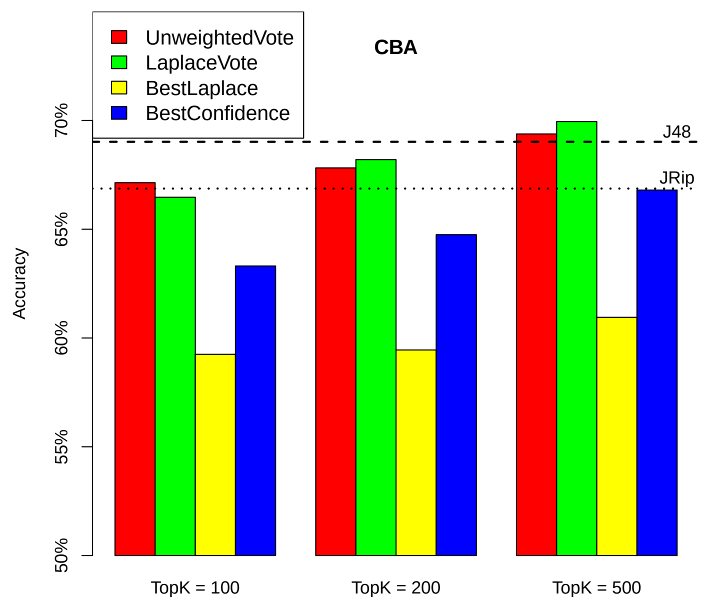

Fig. 1. Comparison of the accuracy of CARMA using the standard CBA method for rule selection, with different rule combination strategies to the baselines.

7 Conclusions

Human activity recognition, and interpretable models for classification are prominent research directions, especially considering the ever-increasing amount of available sensor data and social media. In this paper, we presented a unifying view on these topics, proposing a novel approach adaptive class association rule mining using subgroup discovery. We successfully applied and evaluated this approach in the field of human activity recognition.

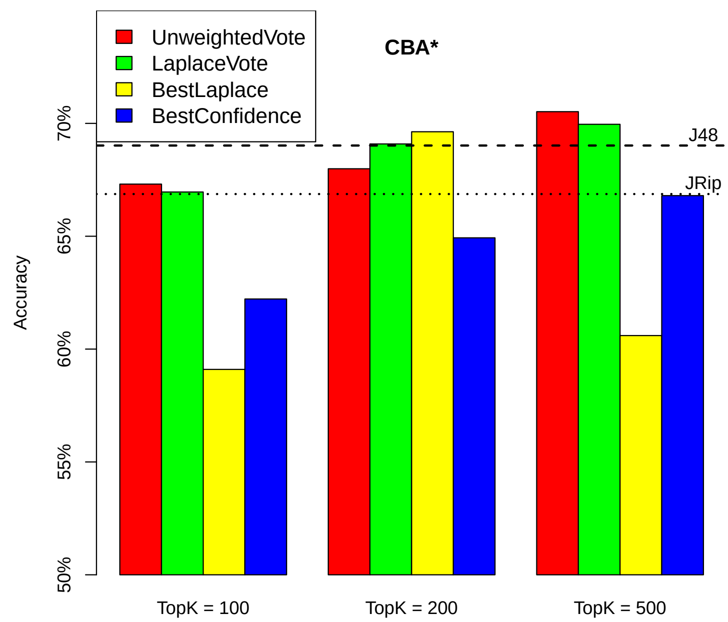

Fig. 2. Comparison of the accuracy of CARMA using the (improved) CBA* method for rule selection, with different rule combination strategies to the baselines.

The proposed CARMA framework is especially suited for generating interpretable rule sets for classification, with a low model complexity. We discussed and analyzed different instantiations of CARMA, e. g., for parameter selection and for obtaining the final classifier. For our evaluation, we applied real-world data collected for different activities using mobile phone sensors. Our experiments showed, that the proposed approach can outperform the baselines clearly, both in terms of accuracy and complexity of the resulting predictive model.

For future work, we aim to consider more datasets, in order to extend the evaluation further. In addition, we aim to analyze the performance of CARMA in further domains, e. g., in the medical domain, or for classifying social media. Furthermore, we plan to investigate further rule assessment and rule selection strategies in detail, e. g., [36], in order to perform further algorithmic comparison and assessment. Based on these, we aim to provide guidelines for instantiating the CARMA framework for specific contexts, also in semi-automatic scenarios [5].

References

1. Agresti, A.: An Introduction to Categorical Data Analysis. Wiley-Blackwell (2007)
2. Atzmueller, M.: Knowledge-Intensive Subgroup Mining – Techniques for Automatic and Interactive Discovery, DISKI, vol. 307. IOS Press (March 2007)
3. Atzmueller, M.: Data Mining on Social Interaction Networks. JDMDH 1 (2014)
4. Atzmueller, M.: Subgroup Discovery – Advanced Review. WIREs Data Mining and Knowledge Discovery 5(1), 35–49 (2015)
5. Atzmueller, M., Baumeister, J., Hemsing, A., Richter, E.J., Puppe, F.: Subgroup Mining for Interactive Knowledge Refinement. In: Proc. Conf. on Artificial Intelligence in Medicine. pp. 453–462. LNAI 3581, Springer, Heidelberg, Germany (2005)
6. Atzmueller, M., Baumeister, J., Puppe, F.: Quality Measures and Semi-Automatic Mining of Diagnostic Rule Bases. In: Proc. INAP. pp. 65–78. No. 3392 in LNAI, Springer, Heidelberg, Germany (2005)
7. Atzmueller, M., Baumeister, J., Puppe, F.: Semi-Automatic Learning of Simple Diagnostic Scores Utilizing Complexity Measures. Artif Intell Med 37(1) (2006)
8. Atzmueller, M., Lemmerich, F.: Fast Subgroup Discovery for Continuous Target Concepts. In: Proc. ISMIS. LNCS, vol. 5722, pp. 1–15. Springer, Heidelberg, Germany (2009)
9. Atzmueller, M., Lemmerich, F.: VIKAMINE - Open-Source Subgroup Discovery, Pattern Mining, and Analytics. In: Proc. ECML/PKDD. LNAI, vol. 7524. Springer, Heidelberg, Germany (2012)
10. Atzmueller, M., Lemmerich, F.: Exploratory Pattern Mining on Social Media using Geo-References and Social Tagging Information. IJWS 2(1/2) (2013)
11. Atzmueller, M., Lemmerich, F., Krause, B., Hotho, A.: Who are the Spammers? Understandable Local Patterns for Concept Description. In: Proc. 7th CMS Conference. Oprogramowanie Nauko-Techniczne, Krakow, Poland (2009)
12. Atzmueller, M., Mueller, J., Becker, M.: Mining, Modeling and Recommending 'Things' in Social Media, LNAI, vol. 8940, chap. Exploratory Subgroup Analytics on Ubiquitous Data. Springer, Heidelberg, Germany (2015)
13. Atzmueller, M., Puppe, F.: SD-Map - A Fast Algorithm for Exhaustive Subgroup Discovery. In: Proc. ECML/PKDD. LNAI, vol. 4213, pp. 6–17. Springer, Heidelberg, Germany (2006)
14. Atzmueller, M., Puppe, F.: A Case-Based Approach for Characterization and Analysis of Subgroup Patterns. Journal of Applied Intelligence 28(3), 210–221 (2008)
15. Atzmueller, M., Puppe, F., Buscher, H.P.: Exploiting Background Knowledge for Knowledge-Intensive Subgroup Discovery. In: Proc. 19th International Joint Conference on Artificial Intelligence. pp. 647–652. Edinburgh, Scotland (2005)
16. Bao, L., Intille, S.S.: Activity Recognition from User-Annotated Acceleration Data. In: Pervasive Computing, pp. 1–17. Springer (2004)
17. Berchtold, M., Budde, M., Gordon, D., Schmidtke, H., Beigl, M.: ActiServ: Activity Recognition Service for Mobile Phones. In: Proc. ISWC. pp. 1–8 (Oct 2010)
18. Clark, P., Boswell, R.: Rule Induction with CN2: Some Recent Improvements. In: Proc. EWSL. pp. 151–163. Springer, Heidelberg, Germany (1991)
19. Cohen, W.W.: Fast Effective Rule Induction. In: Twelfth International Conference on Machine Learning. pp. 115–123. Morgan Kaufmann (1995)
20. Duivesteijn, W., Knobbe, A., Feelders, A., van Leeuwen, M.: Subgroup Discovery Meets Bayesian Networks–An Exceptional Model Mining Approach. In: Proc. ICDM. pp. 158–167. IEEE, Washington, DC, USA (2010)

21. Fábián, Á., Győrbíró, N., Hományi, G.: Activity Recognition System for Mobile Phones using the MotionBand Device. In: Proc. MOBILWARE. p. 41. ICST (2008)
22. Fayyad, U.M., Irani, K.B.: Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In: IJCAI. pp. 1022–1029 (1993)
23. Foerster, F., Smeja, M., Fahrenberg, J.: Detection of Posture and Motion by Accelerometry: A Validation Study in Ambulatory Monitoring. Computers in Human Behavior 15(5), 571–583 (1999)
24. Gamberger, D., Lavrac, N.: Expert-Guided Subgroup Discovery: Methodology and Application. Journal of Artificial Intelligence Research 17, 501–527 (2002)
25. Herrera, F., Carmona, C., Gonzalez, P., del Jesus, M.: An Overview on Subgroup Discovery: Foundations and Applications. KAIS 29(3), 495–525 (2011)
26. Huynh, T., Schiele, B.: Analyzing Features for Activity Recognition. In: Proc. 2005 Joint Conf. on Smart Objects and Ambient Intelligence. pp. 159–163. ACM (2005)
27. Klösgen, W.: Explora: A Multipattern and Multistrategy Discovery Assistant. In: Advances in Knowledge Discovery and Data Mining, pp. 249–271. AAAI (1996)
28. Klösgen, W.: Handbook of Data Mining and Knowledge Discovery, chap. 16.3: Subgroup Discovery. Oxford University Press, New York (2002)
29. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity Recognition using Cell Phone Accelerometers. SIGKDD Explor. Newsl. 12(2), 74–82 (Dec 2010)
30. Lara, O.D., Labrador, M.A.: A Survey on Human Activity Recognition using Wearable Sensors. Communications Surveys & Tutorials, IEEE 15(3), 1192–1209 (2013)
31. Lavrac, N., Kavsek, B., Flach, P., Todorovski, L.: Subgroup Discovery with CN2-SD. Journal of Machine Learning Research 5, 153–188 (2004)
32. Leman, D., Feelders, A., Knobbe, A.: Exceptional Model Mining. In: Proc. ECML/PKDD. LNCS, vol. 5212, pp. 1–16. Springer (2008)
33. Lemmerich, F., Rohlfs, M., Atzmueller, M.: Fast Discovery of Relevant Subgroup Patterns. In: Proc. FLAIRS. pp. 428–433. AAAI Press, Palo Alto, CA, USA (2010)
34. Li, W., Han, J., Pei, J.: CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. In: Proc. ICDM. pp. 369–376. IEEE (2001)
35. Liu, B., Hsu, W., Ma, Y.: Integrating Classification and Association Rule Mining. In: Proc. KDD. pp. 80–86. AAAI Press (August 1998)
36. Liu, B., Ma, Y., Wong, C.K.: Improving an Association Rule Based Classifier. In: Principles of Data Mining and Knowledge Discovery. pp. 504–509. Springer (2000)
37. Puppe, F., Atzmueller, M., Buscher, G., Huettig, M., Lührs, H., Buscher, H.P.: Application and Evaluation of a Medical Knowledge-System in Sonography (SonoConsult). In: Proc. ECAI. pp. 683–687. IOS Press (2008)
38. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
39. Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., Srivastava, M.: Using Mobile Phones to Determine Transportation Modes. ACM TOSN 6(2), 13:1–13:27 (2010)
40. Sulzmann, J.N., Fürnkranz, J.: A Comparison of Techniques for Selecting and Combining Class Association Rules. In: Proc. LWA. Technical Report, vol. 448, pp. 87–93. Dept. of Computer Science, University of Würzburg, Germany (2008)
41. Thabtah, F.: A Review of Associative Classification Mining. KER 22(1) (2007)
42. Wrobel, S.: An Algorithm for Multi-Relational Discovery of Subgroups. In: Proc. PKDD. LNCS, vol. 1263, pp. 78–87. Springer, Heidelberg, Germany (1997)
43. Yang, J.: Towards Physical Activity Diary: Motion Recognition Using Simple Acceleration Features with Mobile Phones. In: Proc. Intl. Workshop on Interactive Multimedia for Consumer Electronics. pp. 1–10. ACM, New York, NY, USA (2009)
44. Yin, X., Han, J.: CPAR: Classification based on Predictive Association Rules. In: Barbará, D., Kamath, C. (eds.) Proc. SDM. pp. 331–335. SIAM (2003)