An Empirical Investigation of Bias and Unfairness in Case-Based Adaptation

Thomas Gabel

Faculty of Computer Science and Engineering
Frankfurt University of Applied Sciences
60318 Frankfurt am Main, Germany
tgabel@fra-uas.de

Abstract. When constructing artificially intelligent systems, it may happen that bias and unfairness against a protected group, that do exist in the real world or in real-world data, are mapped into the digital world. If such systems are then used in practice for decisions that influence humans' life, the obvious question is: Do the algorithms just pass through, perhaps mitigate, or even exacerbate such undesirable characteristics? That question has recently been addressed for the retrieve and retain phase of Case-Based Reasoning (CBR) systems. The paper at hand aims at providing an answer to that question for the *reuse* step in CBR, viz for case-based adaptation in k-nearest neighbor classification tasks.

1 Introduction

The consideration of fairness and bias as well as ethical implications of machine learning algorithms and their application to sensitive domains has become a topic of active research in recent years. Accordingly, a plethora of fairness metrics has been developed and, correspondingly, a large number of (meta) analyses has been published [30, 12]. Research in Case-Based Reasoning (CBR) has started to address fairness issues in previous years, too. Knowing that the CBR paradigm advocates that "Similar problems have similar solutions" [36], one might adapt that principle to a human-focused and fairness-related setting by requiring that "Similar individuals should be given similar decisions" [11]. Thus, unfairness may arise in a CBR system, if it underpredicts a desirable outcome for a specific group of people (e.g. a credit approval decision for a specific gender), or, likewise, overpredicts a non-desirable outcome for some group (e.g. crime recidivism prediction for people of some specific skin color or race).

Blanzeisky et al. [10] have shown that nearest neighbor retrieval, that usually resides at the core of a CBR system, is generally not bias-free. They target this issue by suggesting a multi-objective approach for learning a similarity metric that yields less biased retrieval results. In later work [9], the same authors have focused on the case data itself. The case base stores experience, and if that experience reflects biased decisions taken in real life, then predictions made on their basis might be easily biased, too. Thus, they propose a case-base maintenance approach that removes cases from the case base that cause unfair classifications.

Editio Emendata: This version corrects a few typographical errors in the official paper published in the conference proceedings.

Based on Richter' knowledge container model [35], we observe that the similarity knowledge and the case data container have been subject to fairness-related studies, but the vocabulary and adaptation knowledge containers have, to the best of the author's knowledge, remained unexplored. The paper at hand attempts to fill a part of that gap by investigating how and to which extent case-based adaptation approaches can yield unfair decisions. More specifically, we restrict our focus to classification problems and utilize adaptation as a mean to combine the solutions (i.e. class labels) of a query's k nearest neighbors to one uniquely predicted class membership. Past reuse research has addressed both, the question of (a) how many nearest neighbors should be considered as well as (b) how specifically to amalgamate their predictions. Traditionally, the focus has been mostly on performance-related issues like classification accuracy or scores, but not on aspects of fairness. Our goal is, thus, to highlight aspects of fairness in the context of the reuse phase for case-based k-nearest neighbor (k-NN) classification. An analysis of more sophisticated reuse approaches such as structural or transformational case adaptation [24] is, however, beyond our scope.

Fairness and bias in machine learning have been a topic of active research for many years. We therefore devote Section 2 to a concise review of relevant related work and foundations on approaches to assess bias and fairness in machine learning. We discuss sources of unfairness, practically adopted approaches to quantify it, derive factors that reinforce it and, in so doing, bridge the gap to established k-nearest neighbor reuse methods in CBR. Section 3 presents the methodological approach and contributions required for our empirical investigations on the impact of case reuse on fairness. Section 4 outlines our experiments and the characteristics of the classical fairness-impaired data sets we utilized. Moreover, a comprehensive overview of our results is given, including detailed analyses on the influence of k on fairness subject to the characteristics of the case data as well as with respect to the specific reuse approach taken.

2 Foundations and Related Work

In this section, we introduce the necessary vocabulary, paying special attention to bridging the gap between terms used in the fairness literature and in the CBR community. Additionally, we interweave relevant related work.

2.1 Notation

We start by clarifying the notation used throughout the paper, aligning fairness-related and CBR-related concepts. As indicated, our focus is on classification tasks. Since a case c from the set of all cases $\mathcal{M} = (\mathbb{X}, Y)$ consists of a problem part x and a solution part y, i.e. $c = (x, y)$, the solution is an element of a set of class labels Y. We also denote the solution as the target variable. By contrast, the problem part x comprises the case's features, i.e. $x \in \mathbb{X} = (X_1 \times \cdots \times X_d)$. As usual, $sim : \mathbb{X} \times \mathbb{X} \rightarrow [0, 1]$ denotes a similarity measure to assess the similarity between two cases' problem parts or between a query and problem.

One or more of these features are considered to be *sensitive* ones, such as gender, race, handicaps or others, which must be treated carefully. Without loss of generality, we stick to the existence of exactly one such sensitive feature S in the remainder of this paper, i.e. $\exists s \in \{1, \ldots, d\}$ with $S = X_s$. Moreover, we say that individuals in the protected group, which are potentially subject to discriminatory treatment, have the sensitive feature value of 0 (i.e. $x_s = 0$ or, for short, $S = 0$) whereas the rest has $S = 1$.

In classification, there is usually a desired prediction – a class label that is generally considered better than other outcomes (e.g. a loan granted vs. not granted). Without loss of generality, we subsequently stick to two classes and denote the desired outcome by $y = 1$ and the non-desirable one by $y = 0$. We also differentiate between the solution y of a case and the predicted solution $\hat{y}(q) \in Y$ for a new query $q \in \mathbb{X}$. If the context is clear, we write $\hat{y}$ for brevity.

Finally, for a given case data set $\mathcal{D} \subset \mathcal{M}$ we will repeatedly refer to the commonly used pairwise disjoint decomposition into a training set (the case base) $\mathcal{C}$, a validation set $\mathcal{V}$, and a test set $\mathcal{T}$ with $\mathcal{D} = \mathcal{C} \cup \mathcal{V} \cup \mathcal{T}$.

2.2 Background on Bias and Fairness

A widely accepted distinction concerning potential causes for unfairness and bias is made by [26]. First, unfairness may arise from *prejudice* that is due to some statistical dependence between the sensitive feature S and the target variable Y *or* between S and non-sensitive features from $\mathbb{X}$. Then, the (in)direct use of S during decision-making yields a predictor that basically supports discrimination with respect to S. While there exist approaches towards removing prejudice [26], in a CBR context this may be addressed by the contents of the vocabulary knowledge container [35], which is beyond the focus of this paper.

Second, *negative legacy* is frequently highlighted as a source of unfairness. It refers to biased sampling or case labelling habits in the real world that made it into the data. For example, if prison inmates' early release decisions were affected by judges disadvantaging some minority group due to skin color, race, or other sensitive features, then the recorded case data distribution will be skewed. This type of bias is notably problematic in a CBR setting which fundamentally relies on storing and re-using past experience. Addressing that type of prejudice falls into the realm of CBR's retain phase [1] and has already been examined in [9].

The third source of prejudice is called *underestimation* and denotes a decision model that generalizes too much and is thus susceptible to making unfair classifications as it does not regard minorities sufficiently. One might expect that CBR is rather robust against that kind of bias, as it does not build a model in a supervised way that might underfit the data. However, underestimation can nonetheless emerge in a CBR system, as the similarity measure employed in the retrieve phase may select nearest neighbors in a biased way. Moreover, the way in which the neighbors' solutions are adapted and combined may give rise to unfairness as well. While the former, i.e. the learning of a similarity measure that is geared towards fairness, has been addressed in [10], fairness and underestimation issues during the reuse step are the topic of the paper at hand.

2.3 Quantifying Unfairness

There exists an abundance of fairness measures in the literature, for a concise review the reader is referred to [11] or [12]. A discrimination measure that is widely referred to in the judicial system (e.g. in US courts or in the UK Sex Discrimination Act), is the *disparate impact* [17], also denoted as slift [34] or impact ratio [44]. It represents the ratio of desired outcomes ($\hat{y} = 1$) for the protected group ($S = 0$) to the desired outcomes for the non-protected (majority) group on a given test data set $\mathcal{T} \subset \mathcal{M}$:

$$DI_S(\mathcal{T}) = \frac{P(\hat{y}(q) = 1|(q, y) \in \mathcal{T}, q_s = 0)}{P(\hat{y}(q) = 1|(q, y) \in \mathcal{T}, q_s = 1)}, \text{ or for short, } DI_S^{\mathcal{T}} = \frac{P(\hat{y} = 1|S = 0)}{P(\hat{y} = 1|S = 1)}. \tag{1}$$

In legal practice, it is commonly assumed that discriminatory behavior is present if $DI_S < \tau := \frac{4}{5}$, i.e. if the desired outcomes for the protected group are 20% or more below those of the general group. Interestingly, the DI measure may also be applied to the full training data set (or case base) and, thus, hint to the amount of discrimination in the given data. We will return to that point in Section 3.1.

When assessing the additional unfairness introduced by a machine learning-based system, [8] suggest the *underestimation score* which is a simplified, group-specific modification of the underestimation index from [26]. For a group $g \in S$, the underestimation score is defined for a train ($\mathcal{C}$) and test case set ($\mathcal{T}$) as

$$US_{S=g}(\mathcal{C}, \mathcal{T}) = \frac{P(\hat{y}(q) = 1|(q, y) \in \mathcal{T}, q_s = g)}{P(y = 1|(x, y) \in \mathcal{C}, x_s = g)}, \text{ short, } US_{S=g} = \frac{P(\hat{y} = 1|S = g)}{P(y = 1|S = g)} \tag{2}$$

and it tells whether the desired outcome (1) for a considered group g is actually lower ($US < 1$), equal, or higher ($US > 1$) than what one would expect from the training data. In particular, if $US_{S=0} \ll US_{S=1}$, one has to attest a severe amount of unfairness specifically against protected group $g = 0$.

Factors for Underestimation In [8], the authors identify four factors that may contribute to the degree of underestimation a learning algorithm is subject to. Besides (1) noise (irreducible error), (2) imbalance among the target classes, and (3) under-represented groups – all of which refer to the given training data, i.e. the learner itself is not liable for them – they point to (4) the level of regularization an algorithm applies and confirm for various classic learning algorithms that regularization impacts underestimation and generally reduces the US value.

In line with that, Blanzeisky et al. have proposed two approaches to mitigate underestimation in CBR. They learn a less biased similarity metric using particle swarm optimization [10] where the solution found approaches Pareto optimality with respect to accuracy *and* underestimation. Furthermore, they eliminate cases from the case base [9] that enforce an underestimation of the desirable outcome for the protected group. Both approaches can be interpreted as regularizing the system – they regularize the similarity measure and prune the case base, accepting minor negative impacts on classification performance in exchange for fairness gains. Our intent is to complement these pieces of work by analyzing how bias, underestimation and, hence, unfairness can arise in CBR's reuse phase.

2.4 *k*-Nearest Neighbor Reuse Methods

The goal of the CBR cycle's [1] second phase (reuse) is to adapt, combine or merge a query's k nearest neighbors' solutions to a new suggested solution [36]. As we focus on classification, the new solution must be a class label from Y, reducing the reuse phase to some variant of voting among the k nearest neighbors.

2.4.1 *k*-NN-Based Classification

Formally, given a case base $\mathcal{C} \subset \mathcal{M}$ of n cases, a query $q \in \mathcal{M}$, some similarity measure sim, and $k \geq 1$, a CBR system's retrieve step determines a sorted list $\mathcal{N}_k$ of k cases from $\mathcal{C}$ that are most similar to q. Thus, $\mathcal{N}_k = [c_1, \ldots, c_k]$ is sorted by similarity of its entries to q, i.e. $sim(q, c_i) \geq sim(q, c_j) \forall i < j$. As a shortcut, we refer to these similarities by s_i subsequently ($s_i := sim(q, c_i) \forall c_i \in \mathcal{N}_k$). Moreover, we refer to the cases' solutions by y_i, i.e. if $c_i = (a, b)$, then $y_i := b \in Y$.

The question how to combine the nearest neighbors' solutions (i.e. the actual reuse) has been studied for decades, starting with the introduction of the k-NN rule by [18]. The general k-NN rule determines the suggested solution $\hat{y}$ as:

$$\hat{y}(q) = \arg\max_{l \in Y} \sum_{i=1}^{k} w_i \delta(y_i, l) \ (=: \hat{y} \text{ for short}), \tag{3}$$

where the Kronecker δ gives one, if its two arguments are equal, and zero otherwise. Here, the crucial question becomes how to define the weights w_i and how to set k in relation to n. For example, the mentioned early k-NN rule implements a simple majority voting due to using constant weights ($1/k$), but much work has focused on deriving and analyzing more elaborated variants (cf. Section 2.4.2).

2.4.2 Interpretation of *k*-NN as a Regularizer

It is well-known that the number k of nearest neighbors used in case-based classification can be interpreted as a regularization mechanism. The bias of the 1-nearest neighbor classifier is often low at a high variance, as it reuses only the class label of the query's most similar case [23]. For larger k, the reused class label depends on the k most similar neighbors and can thus, e.g., mitigate noise by exerting more bias while reducing variance. Cover and Hart [13] have shown that asymptotically the error of the 1-nearest neighbor classifier is never more than twice the Bayes-optimal error. Moreover, it has been shown that the original, simple majority vote is optimal for $n \to \infty$, if $k \to \infty$ and $\frac{k}{n} \to 0$ [14].

Practical applications typically favor finite k values and non-uniform weighting. Since the best k is dictated by the model's bias and variance characteristics, this technical choice directly affects the classifier's fairness. Consequently, optimizing for accuracy can unintentionally bias the results when sensitive features are involved. While fairness has received little attention in conjunction with k-nearest neighbor voting, there exist k-NN classification methods specifically tailored for imbalanced data sets which aim at re-weighting neighbors accordingly [4, 42, 15]. Likewise, there are approaches that determine a good value of k

on the fly [3, 20, 38] or that take the distribution of feature values into account when weighting cases [29, 33]. Moreover, there are reuse approaches that advocate the mingling of cases to form centroids per class, so-called local mean non-parametric classifiers, and make the classification decision using those adapted cases (original work in [31] with many follow-up approaches).

Since a full coverage of all existing approaches is not feasible, we have devised a selection of rather simple, though established approaches from the literature that we compare with one another, subsequently (cf. Section 3.2). Our selection is by far not exhaustive, which is why the interested reader is referred to [19] and [2] for comprehensive overviews of many further k-NN reuse methods.

3 Methods

There exists a rich body of methods for fairness analyses as well as for k-NN-based case reuse. We complement this by some selective enhancements.

3.1 Fairness as a Result of Balanced Under- and Overestimation

The established underestimation score also covers overestimation situations for values of $US_{S=g}$ larger than one. This hints to a critical point when using the underestimation score as a measure for comparing the fairness of different systems in case that some system *over*estimates the desirable outcome. One group, for example, the protected one, may feature the desired outcome far less in the data, i.e. $P(y = 1|S = 0) \ll P(y = 1|S = 1)$ with $P(y = 1|S = 0)$ near zero. As a consequence, for some overestimating classifier the numerical value $US_{S=0}$ may take rather large values $\gg 1$ due to the small denominator in Equation 2 which are hard to compare with scores < 1 for genuine underestimation. Thus, comparing US values is fine, as long as the focus is exclusively on underestimation, but underestimation scores larger than one should be treated with care.

A learning system that has been trained on some data set $\mathcal{C} \subsetneq \mathcal{D}$ may – if it exerts some bias – generally affect not just a single group of instances unfair (e.g. the sensitive group), but probably all groups. For example, a trained classifier may grant credit approvals to 20% of the female applicants (from test set $\mathcal{T} \subsetneq \mathcal{D}$), while 30% of the females within $\mathcal{C}$ obtained the credit. Clearly, this is unfair. We might also observe that the group of male applications contained in $\mathcal{C}$ received the credit in 40% of the cases and, hence, conclude that the training data is subject to negative legacy. What is even worse, however, is when, at the same time the classifier grants credits to 50% of the male applicants from $\mathcal{T}$ at application time. Therefore, we suggest to consider underestimation for the sensitive group jointly with under-/overestimation for the other (e.g. majority) group, and vice versa. This point has been indicated by other authors in the fairness literature, coining differing terms for it [28, 7], and is reflected in the original definition of disparate impact (cf. Equation 1).

As pointed out in Section 2.3, we propose to utilize the disparate impact measure to assess the unbalancedness within the given case base $\mathcal{C}$. We therefore

define

$$DI_S(\mathcal{C}) = \frac{P(y=1|(x,y)\in\mathcal{C}, x_s=0)}{P(y=1|(x,y)\in\mathcal{C}, x_s=1)} \text{ or, for short, } DI_S^{\mathcal{C}} = \frac{P(y=1|S=0)}{P(y=1|S=1)} \tag{4}$$

as the baseline disparate impact that quantifies the level of unequal treatment among the sensitive ($S=0$) and remaining ($S=1$) group as it is given by the case data $\mathcal{C}$. Putting into relation the actually observed disparate impact $DI_S^{\mathcal{T}}$ on test cases (Equation 1) and the baseline disparate impact (Equation 4), we propose the *relative disparate impact*

$$RDI(\mathcal{C},\mathcal{T}) = \frac{DI_S(\mathcal{T})}{DI_S(\mathcal{C})} \text{ (=: } RDI \text{ for short)}, \tag{5}$$

as a measure to quantify the additional amount of unfairness introduced by a learned prediction system. With some basic algebraic manipulations we see that

$$RDI = \frac{DI_S^{\mathcal{T}}}{DI_S^{\mathcal{C}}} = \frac{P(\hat{y}=1|S=0)}{P(y=1|S=0)} \cdot \frac{P(y=1|S=1)}{P(\hat{y}=1|S=1)} = \frac{US_{S=0}(\mathcal{C},\mathcal{T})}{US_{S=1}(\mathcal{C},\mathcal{T})} \underset{\text{short}}{=} \frac{US_{S=0}}{US_{S=1}}.$$

In alignment with the original definition of the DI and its frequent use in different areas (cf. Section 2.3), we propose to ascribe a relative disparate impact to some classifier, if $RDI \notin [\tau, \frac{1}{\tau}]$ with the commonly used value $\tau = \frac{4}{5}$, i.e. if it underestimates or overestimates one or the other group by more than 20%.

3.2 Utilized k-Nearest Neighbor Classification Rules

As argued in Section 2.4.1, deciding for a class label in a case-based way using the suggestions of a query's k nearest neighbors represents a (possibly weighted) voting problem (cf. Equation 3). A large body of k-NN rules has been proposed, which can be broadly categorized into four groups:

a) The *simple majority vote* (cf. Section 2.4) is made by the original k-NN rule [18] using a constant weight w_i (e.g. $w_i = 1$ or $1/k$).
b) *Similarity-weighted approaches* apply some function f to the numeric similarity s_i between query and the ith neighbor from $\mathcal{N}_k$, i.e. $w_i = f(s_i)$ (where $s_i = sim(q, \mathcal{N}_k[i])$.
c) *Rank-weighted approaches* apply some function g to the rank i of c_i within the list of q's nearest neighbors, i.e. $w_i = g(i)$.
d) Specialized variants: These may include, but are not limited to
   - *combinations* of b) and c) that apply some function h to both, the similarity s_i and rank i of some neighbors c_i from $\mathcal{N}_k$: $w_i = h(s_i, i)$, or
   - *class-based normalizations* that additionally adjust the weight w_i of each neighbor c_i from $\mathcal{N}_k$ w.r.t. the frequency of its class y_i within $\mathcal{N}_k$.

Since an exhaustive coverage of such rules in infeasible, we have selected a few representatives for each such category that we include in our investigations (see Table 1). Notably, rule R_7 has been highlighted for offering good performance across many domains [21]. Moreover, the selection also includes two rules that we propose (R_{12} and R_{13}). Although the underlying ideas are not fundamentally new, their specific formulation appears to be novel within the existing literature.

Cat.	No.	$w_i =$	Description	Origin
a)	R_1	$const$	simple majority vote	[18]
b)	R_2	s_i	similarity-based vote	[16]
	R_3	$s_i - s_k$	similarity-based vote relative to base similarity	
	R_4	$\frac{s_i - s_k}{s_1 - s_k}$	normalized similarity-based vote	[16]
	R_5	s_i^2	squared similarity-based vote	[40]
	R_6	$e^{\alpha s_i^\beta}$	exponential similarity-based vote	[41]
	R_7	$\frac{s_i - s_k}{s_1 - s_k} \cdot \frac{s_k + s_1}{s_k + s_1 - s_i}$	dual-normalized similarity-based vote	[21]
c)	R_8	$k - i + 1$	rank-based vote	[16]
	R_9	$\frac{1}{i}$	inverse rank-based vote	[22]
	R_{10}	$w_{i+1} + w_{i+2}$	Fibonacci-based vote	[32]
d)	R_{11}	$\frac{s_i - s_k}{s_1 - s_k} \cdot \frac{1}{i}$	multiplicative combination of R_4 and R_9	[22]
	R_{12}	cf. Equation 7	class representatives-based similarity vote	Sec.3.3.1
	R_{13}	cf. Equation 8	sim-based vote normalized by class frequency	Sec.3.3.2

Table 1. k-NN rules considered in the remainder of this paper. Note that each rule that features a fraction also covers the special case of a zero denominator by letting $w_i = 1$. If the rules allow for parameter adjustments, we work with fixed settings throughout this paper: $\alpha = \beta = 1$. For R_{12} and R_{13} see the text for full explanations.

3.3 *k*-NN Rules Under Imbalanced Data

For imbalanced data sets where one class $y \in Y$ dominates the data a voting among a large number k of neighbors becomes (for increasing k) more and more biased since the majority class will dominate the result. Similarity-weighted voting rules, for instance, represent a first approach to mitigate that problem. The issue becomes even more prevalent, however, if we know that there is some protected group $g \in S$ that is presumed to having been treated unfair in practice and, therefore, is represented in the data in a biased way. The underlying idea of both of the following rules is to take into account the distribution of case solutions in the vicinity of the query.

3.3.1 Class Representatives-Based Similarity Vote

The idea is to split the list of nearest neighbors $\mathcal{N}_k$ into sublists with respect to the classes of the cases contained. Voulgaris and Magoulas have pursued a similar approach, with the difference of forming multiple lists upfront [39]. Thus, for each possible solution $y \in Y$ we obtain $\mathcal{N}_k^y = [c_j \in \mathcal{N}_k | c_j = y]$, a sorted list with at most k entries. If there exists an $y \in Y$ with $\mathcal{N}_k^y = \emptyset$, then the vote is unambiguous for the other solution (in case of two classes). If there are multiple $y \in Y$ with $|\mathcal{N}_k^y| > 0$, then we denote by κ the number of cases belonging to the class with the least number of instances within $\mathcal{N}_k$, i.e. $\kappa = \min_{y \in Y} |\mathcal{N}_k^y|$. We then use the class y as the suggested solution whose κ top entries in $\mathcal{N}_k^y$ have the highest average similarity to q:

$$\hat{y}(q) = \arg\max_{y \in Y} \frac{1}{\kappa} \sum_{i=1}^{\kappa} sim(q, \mathcal{N}_k^y[i]) \tag{6}$$

As a side node, this definition of $\hat{y}$ can easily be integrated into the weight-based framework from above (cf. Equations 3) by letting

$$w_i = \begin{cases} s_i & \text{if } |\{c_j \in \mathcal{N}_k | j \leq i, y_i = y_j\}| \leq \kappa \\ 0 & \text{else.} \end{cases} \tag{7}$$

Thus, the weights are query-specific and derived during retrieval with minimal computational effort, which is in contrast to related approaches like [15].

3.3.2 Normalized Class Similarity Support

The previous weighting scheme focussed equally on the same number of (top) representatives from each class. We next utilize the entire retrieval result $\mathcal{N}_k$, normalizing each case's weight by the frequency of its class in $\mathcal{N}_k$. If all cases in $\mathcal{N}_k$ belong to the same class, again, the vote is unambiguous, otherwise:

$$w_i = \frac{s_i}{|\mathcal{N}_k^{y_i}|} \tag{8}$$

As a matter of fact, this yields vote results that are identical to those that would arise if we would assign to $\hat{y}$ the class whose cases within $\mathcal{N}_k$ have on average the highest similarity to q. A similar idea has been used by [43] in conjunction with a feature weighting approach. An appealing characteristic of this approach is that the influence of the majority class is constrained. Of course, some noisy data point c_η with y_η might take the lead for small k, but for larger k – where we expect to find further non-noisy neighbors within $\mathcal{N}_k$ that have class y_η as well – the averaging of similarities will filter out the noise.

3.4 Optimization of k with Respect to Fairness

We follow a straightforward procedure to analyze the optimizability of k for the goal of fair case reuse. As mentioned, the given case data $\mathcal{D}$ is randomly split into three disjoint sets: $\mathcal{C}$ forms the actual case base, while a smaller part is used as a separate test query set $\mathcal{T}$ on which final classifier performance is measured and reported. Moreover, there is a validation set $\mathcal{V}$ that we use to search (brute force) for the value of $k^\star$ that maximizes some optimization criterion $\mathcal{L}$ over $\mathcal{V}$, i.e. $k^\star = \arg\max_{k \in \{1,\ldots,n\}} \mathcal{L}(\mathcal{V})$. We compare the simple F_1 score [37] on $\mathcal{V}$

$$\mathcal{L}_1 = F_1(\mathcal{V}) = \frac{2TP}{2TP + FP + FN} = \frac{2P(\hat{y} = 1, y = 1 | (q,y) \in \mathcal{V})}{P(\hat{y}(q) = 1 | (q,y) \in \mathcal{V}) + P(y = 1 | (q,y) \in \mathcal{V})} \tag{9}$$

with a multi-objective criterion that additionally takes into account a fair treatment of groups using the relative disparate impact:

$$\mathcal{L}_2 = F_1(\mathcal{V}) \cdot \begin{cases} RDI(\mathcal{C}, \mathcal{V}) & \text{if } RDI(\mathcal{C}, \mathcal{V}) \leq 1 \\ \frac{1}{RDI(\mathcal{C}, \mathcal{V})} & \text{else} \end{cases} \tag{10}$$

Using $\mathcal{L}_2$ we are thus interested in the value of k that brings about both, good performance (measured with F_1) on the validation set as well as a relative disparate impact as near to one as possible.

Dataset Name (and Area)	#Features and #Cases	Sensitive Feature	Protected group (and Share of)	Desired Outcome (and Share of)	Src
Adult (finance)	14 / 30162	gender	female (32.4%)	earns >50k/yr (24.9%)	U
Bank (marketing)	16 / 45211	marital	single (28.3%)	sale made (11.7%)	K
Credit (finance)	20 / 1000	gender	female (31.0%)	creditworthy (70.0%)	U
Credit Card (fin.)	15 / 653	gender	female (31.1%)	card approved (45.3%)	U
Heart (health)	13 / 270	age	midage (51.1%)	ill (44.4%)	U
Liver (health)	10 / 579	gender	female (24.2%)	ill (71.5%)	K
Recidivism_1 (law)	7 / 7214	race	black (51.2%)	no recidivism (54.9%)	[5]
Recidivism_2 (law)	47 / 16198	race	black (56.1%)	no recidivism (42.2%)	K
Titanic	6 / 887	gender	male (64.6%)	survived (45.3%)	K

Table 2. Data sets used for our analyses: The last column provides a reference to the source where U indicates the UCI Machine Learning Repository [27] and K stands for a Kaggle dataset [25].

4 Experiments

We performed an extensive empirical evaluation using standard datasets to examine the interplay of classifier performance and classifier fairness subject to the algorithmic parameters of CBR's reuse phase, viz the choice of the reuse method and its nearest neighbor weighting as well as the dependency on k.

4.1 Experimental Approach, Datasets and Performance Measures

In all our experiments, we use a large fraction (80%) of the given data $\mathcal{D}$ as case base $\mathcal{C}$ and employ the remainder (10%) as a validation set $\mathcal{V}$ to analyze the domain of possible k values as well as an independent test set $\mathcal{T}$ (10%). All experiments were repeated 100 times (for the smaller data sets) or 10 times (for the much larger ones) to form average values.

We selected a number of datasets frequently used in fairness research. Instances with missing or unknown feature values were omitted. Table 2 summarizes their core characteristics. The retrieve phase utilizes a knowledge-poor default similarity measure modelled according to the local-global principle [6].

4.2 Bias as a Function of k: Visualizing Underestimation

We start by focusing on the classic Adult domain where the task is to predict whether a human earns more than 50k$ per year. With a disparate impact of $DI_S(\mathcal{C}) = 0.362$ this domain is known to be heavily biased against females and has, thus, been frequently used in fairness studies.

The left part of Figure 1 shows what happens, if we perform k-NN classification for increasing k (log-scaled abscissa denoting k/n) using the simple majority voting rule (R_1). The thick-plotted data series show – separately for the sensitive group (gray, $S = 0$) and the remaining group (black, $S = 1$) – that the true negative rates converge to one, while the true positive rates converges to zero since

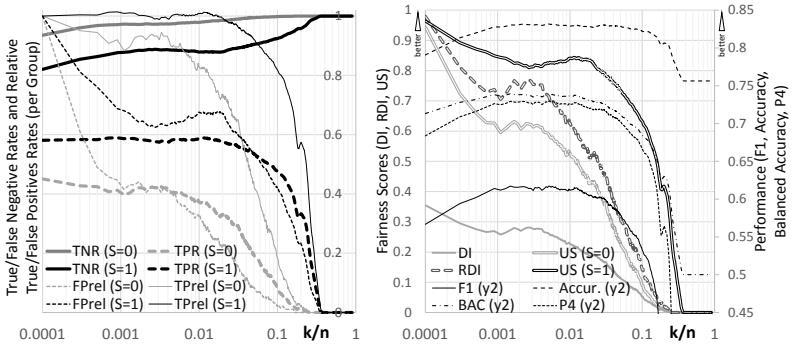

Fig. 1. Left: k-NN classification using the simple majority vote for increasing values of k/n. Separately for the protected group ($S = 1$) and the rest, the chart visualizes the development of true negative and true positive rates as well as the shrinking share of positive classifications, i.e. of desired outcomes. Right: For the same data, this chart plots the underestimation scores for both groups, the relative disparate impact as well as four different performance measures (y_2) of the resulting case-based classifier.

the k-NN classifier becomes more likely to assign more queries to the majority class (which is not earning $50k\$/yr$) for increasing k. Thus, for $k/n > 0.35$ the classifier is the trivial one always returning the non-desired, i.e. negative class.

The thin series in the same chart show how k-NN achieves this, setting the number of false/true positives of the k-NN classifier in relation to the number of false/true positives of the 1-NN classifier. Obviously, *both* the number of false (dashed) positives and true (solid) positives drops much faster for the sensitive group ($S = 0$) compared to the decline for the non-sensitive group ($S = 1$). Of course, this heavily impacts the underestimation scores $US_{S=0}$ and $US_{S=1}$ and, accordingly, the relative disparate impact as can be seen in the right part of Figure 1. The widely used threshold for disparate impact ($\tau = 0.8$) is undercut for $k/n \geq 0.00035$. Additionally, that chart plots (y_2 axis) some performance measures for the obtained k-NN classifier. All of those (accuracy, balanced accuracy, F_1 and P_4 score [37]) take nearly the same shape with maxima in the interval $k/n \in [10^{-3}, 10^{-2}]$ where k-NN's RDI is critical between 0.6 and 0.75.

4.3 k as a Driver for Disparate Impact and Underestimation

We next extend our scope by focusing on multiple domains with fairness issues (Table 2). The bottom-left chart of Figure 2 conveys an overview of the domains' core characteristics. It plots the share of desired outcomes against the share of the protected group within the case data, using data point bubble sizes to indicate the amount of data-inherent bias as expressed by the disparate impact measure $DI_S(\mathcal{C})$ (Equation 4). While each of the domains considered is subject to fairness problems ($DI < 1$), in some of them the protected group takes actually the role of a majority (y value larger than 0.5), and in some of them the desired outcome happens to occur in more than half of the cases (x value larger than 0.5).

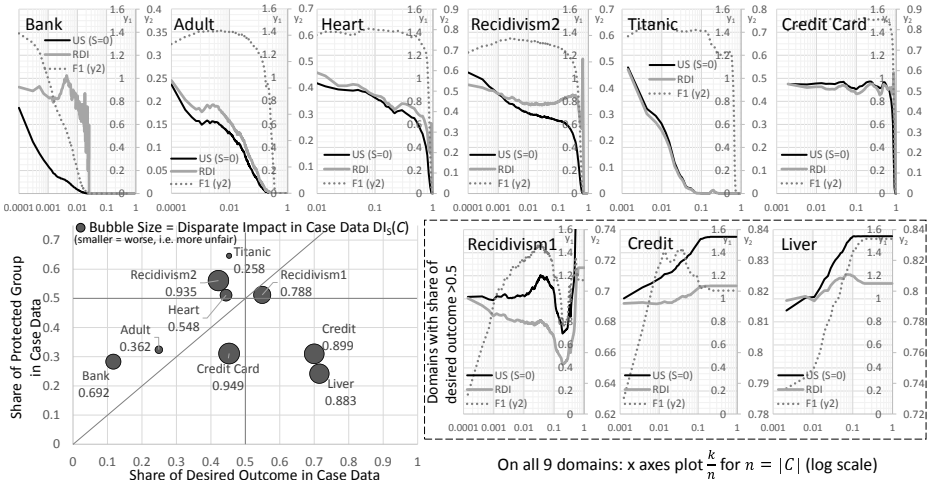

Fig. 2. The bottom-left chart gives an overview of the fairness-related characteristics of the domains considered. The remaining charts visualize for each of the domains the level of underestimation of the protected group subject to k/n as well as the resulting relative disparate impact of the k-NN classifier. F_1 scores are given as a reference to indicate the range of k values that would be practically relevant.

The impact of the latter can be seen in the three bottom right charts of Figure 2. With $k/n \to 1$, the majority class (here, the desired outcome) outnumbers the non-desired outcome resulting in strong overestimation (black curve with US scores larger than one). We also see that domains with roughly balanced shares of the protected group and of the desired outcome (near 0.5/0.5, e.g. Heart oder Recidivism) are only mildly influenced by additional disparate impact due to the k-NN classifier (RDI values mostly within $[\frac{4}{5}, \frac{5}{4}]$ except for extreme k/n values) which is in line with notes by [8] on imbalance among the target class as a source for underestimation. Finally, we identify domains which either come with a low $DI_S(\mathcal{C})$ (small bubbles) value upfront or with a strong imbalance between protected and non-protected group and an imbalance in the class label distribution. For those domains (Bank, Heart, Titanic) both, the underestimation score as well as the disparate impact decline rapidly for growing k values.

4.4 Experiments on the Bias of Case Reuse Methods

Returning to the Adult domain, we examine the impact of the employed reuse methods on underestimation, disparate impact, and the actual performance of the resulting classifier when altering k. Figure 3 contrasts representatives from each of the categories listed in Table 1. Purely sim-based voting methods (R_2 and R_{12}) produce a rather fast decline of the US and (R)DI scores, approaching zero for $k/n > 0.2$, while there are negligible differences in classifier peak performance at $k/n \approx 10^{-2}$. For the two reuse methods where rank information is included in

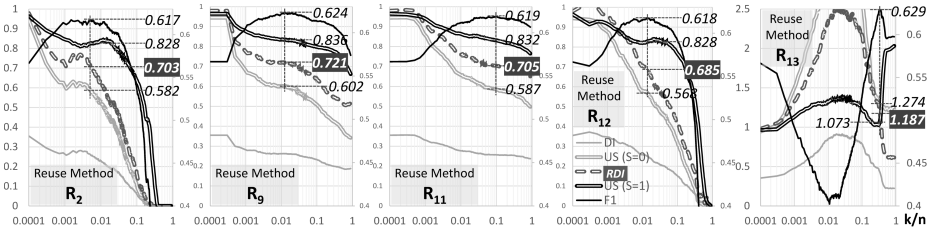

Fig. 3. Adult domain: k is varied from 1 to n, i.e. $k/n \in [10^{-4}, 1]$. We contrast 5 out of the listed (cf. Table 1) k-NN voting methods w.r.t. their influence on the (R)DI and underestimation for the (non-)protected group (primary y-axis). Additionally, the F_1 score of the resulting classifier on independent test data is shown on the secondary ordinate, including markers for the F_1 maximum value and belonging fairness assessments. Used reuse methods from left to right: similarity-based vote, inverse rank-based, multiplicative combination, class representatives-based sim vote, and sim-based vote normalized by class frequency. Mind the differently scaled y_1 axis for the latter one.

the vote (R_9, R_{11}) all fairness-related curves are less steep, approaching values larger zero for $k/n \to 1$. Although this might suggest an improvement in terms of handling biased data, the best value of k, which a practitioner would select using some validation set, yields nearly identical fairness (RDI ~ 0.7) and performance ($F_1 \sim 0.62$) characteristics as the previously discussed sim-based rules.

In contrast to those, rule R_{13} (right in Figure 3) strongly overestimates the desired outcome for both groups. Since this kind of overestimation is even more pronounced for the protected group, the RDI grows to values larger than one, thus "repairing" some of the disparate treatment among groups in the given case data. While this comes generally at the cost of an impaired F_1 score for smaller values of k, the best value of k found ($k/n = 0.335$) also produces the best F_1 score among *all* k-NN rules from Table 1 with no underestimation and a relative disparate impact within the desirable interval ($1.187 \in [\frac{4}{5}, \frac{5}{4}]$).

Search for Fair Values of k In another large-scale series of experiments we iterated over all 9 domains and 13 rules and compared the quality of a k-NN classifier and its fairness properties for 7 specific settings of k. Besides the 1-NN classifier (for which all rules behave identically), we set k to increasing shares of the case base size ($n = |\mathcal{C}|$), viz $\frac{k}{n} \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$. Moreover, we used the values $k^\star$ resulting from the optimization procedure described in Section 3.4 for a separate validation set $\mathcal{V}$ with optimization criteria $\mathcal{L}_1$ and $\mathcal{L}_2$.

Part a) of Figure 4 plots F_1 and RDI on the test cases $\mathcal{T}$ and shows (except for a few outliers) how "clusters" for the different levels of k arise. It is remarkable how the $\mathcal{L}_1$-optimized (i.e. geared towards optimal F_1 score) classifiers "gather" in the bottom-right part of the zoomed-in sub-chart. Likewise, the rules whose k values have been optimized w.r.t. fairness (via $\mathcal{L}_2$ and the incorporation of RDI) appear to be shifted to the top-left compared to their $\mathcal{L}_1$ counterparts. Stated differently, on average a loss of ~ 0.015 in terms of the classifiers' F_1 scores (from ~ 0.695 to ~ 0.68) is compromised for an RDI increase of about

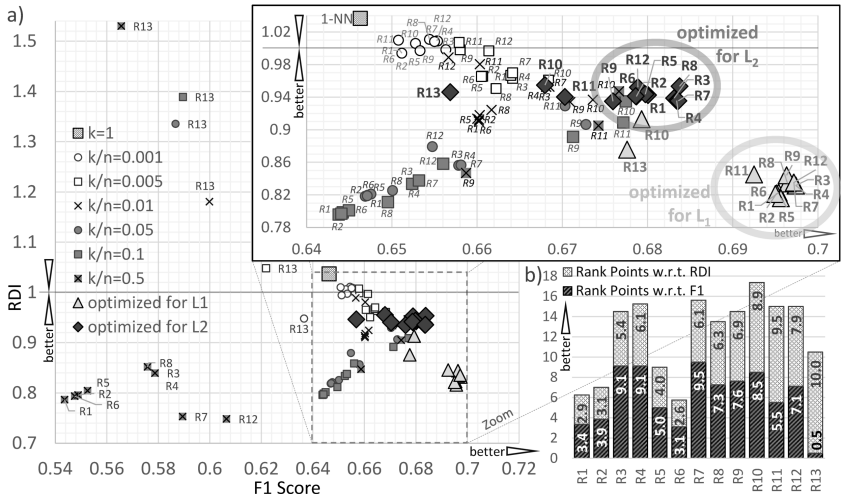

Fig. 4. a) Trade-off between performance (F_1 on x) and fairness (RDI on y), averaged over all domains. The zoomed-in subgraph clarifies specific combinations of reuse method and k setting. b) Attempt to discern the 13 used reuse methods by their average rank (as achieved for different k values) regarding both, performance and fairness.

0.12 (from ~ 0.82 to ~ 0.94, roughly using the centers of the ellipses in Figure 4 as indicator). This shows empirically that case adaptation in CBR's reuse phase can be designed towards reduced bias and unfairness against protected groups.

Rule Ranking We finish our study with a brief comparison across our selection of k-NN rules. The bar chart in part b) of Figure 4 reports aggregate performance of all rules over all domains. Within each specific setting of k (e.g. $k/n = 0.05$) we *ranked* all 13 rules by both, their achieved F_1 score and RDI value, respectively, and awarded each rule $13 - r$ points, if it achieved rank r, forming averages over all settings of k considered previously. Apparently, R_{13} performs best in terms of fairness by keeping the RDI close to one, whereas R_7 is best-in-class with respect to maximizing F_1 (which is in line with [21]). Be aware that these numbers are averages over a wide range of k values including those that might be practically irrelevant – which is why the bars are merely meant as a general indication as to what extent each rule trades fairness for classification performance.

5 Conclusion

We have focused on the algorithmic bias of the reuse step in CBR. For the special case of k-nearest neighbor classification we have systematically analyzed how the specific value of k in conjunction with the employed voting rule exerts bias by treating members of a protected group disparately. In a comprehensive empirical study with established fairness research datasets we found that fairness can be improved at a moderate cost in classification performance by selecting a tailored value of k in combination with favoring appropriate k-NN rules over others.

Declaration of GenAI Use. Writing this manuscript, the author used Lumo and ChatGPT for disambiguating word meanings and LaTeX assistance. He reviewed and edited the content as needed to take full responsibility for the paper.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Commun. 7(1), 39–59 (1994)
2. Abdalla, H., Amer, A.: Enhancing Data Classification Using Locally Informed Weighted kNN Algorithm. Expert Systems With Applications 276 (2025)
3. Acikkar, M., Tokgöz, S.: Improving Multi-Class Classification: Scaled Extensions of Harmonic Mean-Based Adaptive k-Nearest Neighbors. Appl. Intelligence 55 (2025)
4. Ando, S.: Classifying Imbalanced Data in Distance-Based Feature Space. Knowledge and Information Systems 46(3), 707–730 (2016)
5. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine Bias (2016)
6. Bergmann, R., Richter, M., Schmitt, S., Stahl, A., Vollrath, I.: Utility-Oriented Matching: A New Research Direction for CBR. In: (GWCBR) (2001)
7. Berk, R., Heidari, H., Jabbari, S., Kearns, M.: Fairness in Criminal Justice Risk Assessments: The State of the Art. Soc. Methods & Research 50, 3–44 (2017)
8. Blanzeisky, W., Cunningham, P.: Algorithmic Factors Influencing Bias in Machine Learning. In: Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2021). pp. 559–574. Springer (2021)
9. Blanzeisky, W., Cunningham, P.: Addressing Underestimation Bias in CBR Through Case-Base Maintenance. In: 31st International Conference on Case-Based Reasoning (ICCBR 2023). pp. 233–243. Springer, Aberdeen, Scotland (2023)
10. Blanzeisky, W., Smyth, B., Cunningham, P.: Algorithmic Bias and Fairness in Case-Based Reasoning. In: Proceedings of the 30th International Conference on Case-Based Reasoning (ICCBR 2022). pp. 48–62. Springer, Nancy, France (2022)
11. Castelnovo, A., Crupi, R., Greco, G., Regoli, D., Penco, I.: A Clarification of the Nuances in the Fairness Metrics Landscape. Nature Scientific Rep. 12(4209) (2022)
12. Caton, S., Haas, C.: Fairness in ML: A Survey. ACM Comput. Surv. 56(7) (2024)
13. Cover, T., Hart, P.: Nearest Neighbor Pattern Classification. IEEE Trans. Information Theory 13, 21–27 (1967)
14. Devroye, L., Gyorfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer (1996)
15. Dubey, H., Pudi, V.: Class Based Weighted K-Nearest Neighbor over Imbalance Dataset. In: 17th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD). pp. 305–316. Springer, Goald Coast, Australia (2013)
16. Dudani, S.: The Distance-Weighted k-Nearest Neighbor Rule. IEEE Transactions on Systems, Man, and Cybernetics SMC-6, 325–327 (1976)
17. Feldman, M., Friedler, S., Moeller, J., Scheidegger, C.: Certifying and Removing Disparate Impact. In: 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 259–268. ACM Press, Sydney, Australia (2015)
18. Fix, E., Hodges, J.: Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties (1951), Tech. Rep. 4, USAF School of Aviation Medicine, USA
19. Geler, Z., Kurbalija, V., Radovanovic, M., Ivanovic, M.: Comparison of Different Weighting Schemes for the kNN Classifier. Knowl Inf Syst 48, 331–378 (2016)
20. Ghosh, A.: On Optimum Choice of k in Nearest Neighbor Classification. Computational Statistics & Data Analysis 50(11), 3113–3123 (2006)

21. Gou, J., Du, L., Zhang, Y., Xiong, T.: A New Distance-Weighted k-Nearest Neighbor Classifier. J Inf Comput Sci 9, 1429–1436 (2012)
22. Gou, J., Xiong, T., Kuang, Y.: A Novel Weighted Voting for K-Nearest Neighbor Rule. J. Comput. 6(5), 833–840 (2011)
23. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Springer (2013)
24. Hotz, M., Malburg, L., Bergmann, R.: Advanced Search Techniques for Determining Optimal Sequences of Adaptation Rules in Process-Oriented CBR. In: 33rd International Conf. on CBR (ICCBR 2025). pp. 236–251. Biarritz, France (2025)
25. Kaggle Inc.: Data Science Community and Competitions Repository. https://www.kaggle.com (2026), accessed: March 2026
26. Kamishima, T., Akaho, S., Asoh, H., Sakuma, J.: Fairness-Aware Classifier with Prejudice Remover Regularizer. In: Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2012). pp. 35–50. Springer, Bristol, UK (2012)
27. Kelly, M., Longjohn, R.: The UCI ML Repository, https://archive.ics.uci.edu
28. Kusner, M., Loftus, J., Russell, C., Silva, R.: Counterfactual Fairness. In: Proceedings NeurIPS 2017. pp. 4066–4076. NIPS Foundation, Long Beach, USA (2017)
29. Liu, W., Chawla, S.: Class Confidence Weighted kNN Algorithms for Imbalanced Data Sets. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2011). pp. 345–356. Springer (2011)
30. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A Survey on Bias and Fairness in Machine Learning. ACM Comput. Surv. 54(6), 1–35 (2021)
31. Mitani, Y., Hamamoto, Y.: A Local Mean-Based Nonparametric Classifier. Pattern Recognition Letters 27(10), 1151–1159 (2006)
32. Pao, T., Chen, Y., Yeh, J., Cheng, Y., Lin, Y.: A Comparative Study of Different Weighting Schemes on KNN-Based Emotion Recognition in Mandarin Speech. In: Int'l Conf. on Intelligent Computing (ICIC 2007). pp. 997–1005. Springer (2007)
33. Patel, H., Thakur, G.: A Hybrid Weighted Nearest Neighbor Approach to Mine Imbalanced Data. In: Int'l Conf. Data Mining (DMIN 2016). pp. 106–110 (2016)
34. Pedreschi, D., Ruggieri, S., Turini, F.: Measuring Discrimination in Socially-Sensitive Decision Records. In: 9th SIAM International Conference on Data Mining (SDM). pp. 581–592. Soc. for Industr. & Appl. Mathematics, Sparks, USA (2009)
35. Richter, M.: Knowledge Containers. Readings in CBR. Morgan Kaufmann (2003)
36. Richter, M., Weber, R.: Case-Based Reasoning. A Textbook. Springer (2013)
37. Sitarz, M.: Extending F1 Metric, Probabilistic Approach. arXiv:2210.11997 (2022)
38. Song, Y., Huang, J., Zhou, D., Zha, H., Giles, C.: IKNN: Informative k-Nearest Neighbor Pattern Classification. In: ECML PKDD. pp. 248–264. Springer (2007)
39. Voulgaris, Z., Magoulas, G.: Extensions of the k nearest neighbour methods for classification problems. pp. 23–28. ACTA Press (2008)
40. Wu, X., Kumar, V., Quinlan, R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P., Zhou, Z., Steinbach, M., Hand, D., Steinberg, D.: Top 10 Algorithms in Data Mining. Knowledge and Information Systems 14, 1–37 (2008)
41. Zavrel, J.: An Empirical Re-Examination of Weighted Voting for k-NN. In: 7th Belgian-Dutch Conference on Mach. Learn. (BeNeLearn-97). pp. 139–145 (1997)
42. Zhang, X., Li, Y.: A Positive-Biased Nearest Neighbour Algorithm for Imbalanced Classification. In: 17th Pacific-Asia Conference on Advances in Knowl. Discovery and Data Mining (PAKDD). pp. 293–304. Springer, Goald Coast, Australia (2013)
43. Zhou, Z., Pan, G., Wang, X.: Classification Based on Attribute Positive Correlation and Average Similarity. Res. J. Appl. Sci. Eng. Technol. 5(4), 1420–1423 (2013)
44. Zliobaite, I.: Measuring Discrimination in Algorithmic Decision Making. Data Min Knowl Disc 31 (2017)