# Model-based and actual independence for fairness-aware classification



doi: 10.1007/s10618-017-0534-x



Data Min Knowl Disc (2018) 32:258–286
https://doi.org/10.1007/s10618-017-0534-x

# Model-based and actual independence for fairness-aware classification

**Toshihiro Kamishima**[1] · **Shotaro Akaho**[1] · **Hideki Asoh**[1] · **Jun Sakuma**[2,3]

Received: 27 March 2016 / Accepted: 24 July 2017 / Published online: 8 August 2017


**Abstract** The goal of fairness-aware classification is to categorize data while taking into account potential issues of fairness, discrimination, neutrality, and/or independence. For example, when applying data mining technologies to university admissions, admission criteria must be non-discriminatory and fair with regard to sensitive features, such as gender or race. In this context, such fairness can be formalized as statistical independence between classification results and sensitive features. The main purpose of this paper is to analyze this formal fairness in order to achieve better trade-offs between fairness and prediction accuracy, which is important for applying fairness-aware classifiers in practical use. We focus on a fairness-aware classifier, Calders and Verwer's two-naive-Bayes (CV2NB) method, which has been shown to be superior to other classifiers in terms of fairness. We hypothesize that this superiority is due to the difference in types of independence. That is, because CV2NB achieves actual inde-

Responsible editor: Andrea Passerini,Thomas Gaertner, Celine Robardet and Mirco Nanni.

✉ Toshihiro Kamishima
mail@kamishima.net
http://www.kamishima.net/

Shotaro Akaho
s.akaho@aist.go.jp

Hideki Asoh
h.asoh@aist.go.jp

Jun Sakuma
jun@cs.tsukuba.ac.jp

1 National Institute of Advanced Industrial Science and Technology (AIST), AIST Tsukuba Central 2, Umezono 1–1–1, Tsukuba, Ibaraki 305-8568, Japan

2 University of Tsukuba, 1–1–1 Tennodai, Tsukuba 305-8577, Japan

3 RIKEN Center for Advanced Intelligence Project, 1–4–1 Nihonbashi, Chuo-ku, Tokyo, Japan

pendence, rather than satisfying model-based independence like the other classifiers, it can account for model bias and a deterministic decision rule. We empirically validate this hypothesis by modifying two fairness-aware classifiers, a prejudice remover method and a reject option-based classification (ROC) method, so as to satisfy actual independence. The fairness of these two modified methods was drastically improved, showing the importance of maintaining actual independence, rather than model-based independence. We additionally extend an approach adopted in the ROC method so as to make it applicable to classifiers other than those with generative models, such as SVMs.

**Keywords** Fairness · Discrimination · Classification · Cost-sensitive learning

## 1 Introduction

The goal of fairness-aware data mining is to analyze data while taking into account potential issues of fairness, discrimination, neutrality, and/or independence. Techniques of fairness-aware data mining are helpful for avoiding unfair treatments as follows. Data mining techniques are increasingly being used for serious decisions that affect individual's lives, such as decisions related to credit, insurance rates, or employment applications. For example, credit decisions are frequently made based on past credit data together with statistical prediction techniques. Such decisions are considered unfair in both a social and legal sense if they have been made with reference to sensitive features such as gender, religion, race, ethnicity, disabilities, or political convictions. Pedreschi et al. (2008) were the first to propose the concept of fairness-aware data mining to detect such unfair determinations. Since the publication of their pioneering work, several types of fairness-aware data mining tasks have been proposed.

In this paper, we discuss fairness-aware classification, which is a major task of fairness-aware data mining. Its goal is to design classifiers while taking fairness in the prediction of a class into account. Such fairness can be formalized based on independence or correlation between classification results and sensitive features. In general, some degree of prediction accuracy must be sacrificed to satisfy a fairness constraint. However, if a predictor violates the constraint, the predictor cannot be deployed in the real world, because social demands, such as equality of treatment, should not be ignored. Even though a predictor can classify accurately, if it violates a fairness constraint, it does not truly perform the classification task from a social perspective. Therefore, it is important to improve the trade-off between fairness and accuracy in order that a fairness-aware classifier can effectively predict under a specified fairness constraint in practical use.

The main purpose of this paper is to discuss the theoretical background of formal fairness in classification, and to identify important factors for achieving a better trade-off between accuracy and fairness. We here focus on Calders and Verwer's two-naive-Bayes (CV2NB) method (Calders and Verwer 2010), which is a pioneering fairness-aware classifier. This CV2NB classifier has achieved a high level of fairness, as we will show in our experimental section. We analyze this method and hypothesize that

the effects of model bias and a deterministic decision rule are essential for improving fairness–accuracy trade-offs.

We introduce two important factors: model bias and the deterministic decision rule. Model bias is the degree of difference between a true distribution to fit and an estimated distribution represented by a model of a classifier, and such bias has been well discussed in the context of bias-variance theory (Bishop 2006, Sect. 3.2). A fairness constraint must be satisfied based on a sensitive feature and the true distribution of a class. However, if we use a distribution restricted by a model instead of a true distribution, the satisfied fairness constraint diverges from the constraint that we have to satisfy. Hence, model bias may damage the fairness of the learned classifier. A deterministic decision rule is another factor that can worsen the quality of fairness. Once class posteriors or decision functions of a classifier are learned, a class label for a new instance is deterministically chosen by applying a decision rule. For example, a class whose posterior is maximum among a set of classes is deterministically chosen to minimize the risk of misclassification (Bishop 2006, Sect. 1.5). If we assume that classes are probabilistically generated according to a class posterior when designing a fairness-aware classifier, the class labels that are actually produced will deviate from the expected ones. This deviation worsens the quality of fairness. For these two reasons, the influence of model bias and a deterministic decision rule must be carefully maintained in order to satisfy a fairness constraint with the least possible loss of a classifier.

Our first contribution is to distinguish notions of two types of independence: model-based independence and actual independence. Model-based independence is defined as statistical independence between a class and a sensitive feature following a model distribution of a classifier. On the other hand, in the case of actual independence, the effects of model bias and a deterministic rule are considered in the context of a fairness constraint. We formally state these two types of independence, which are important in a context of fairness-aware data mining.

Our second contribution is modifying two existing fairness-aware classifiers so as to satisfy actual independence in order to validate the above hypothesis. The first classifier is our logistic regression with a prejudice remover regularizer (Kamishima et al. 2012), which was originally designed to satisfy a model-based independence condition. The second classifier is a reject option-based classification (ROC) method (Kamiran et al. 2012), which changes decision thresholds according to the values of sensitive features. Though the degree of fairness is adjusted by a free parameter in the original method, we here develop a method to find settings of parameters so that the resultant classifiers respectively satisfy model-based independence and actual independence conditions. By comparing the performance of classifiers satisfying model-based and actual independence, we validate the hypothesis that the effects of model bias and a deterministic rule cannot be negligible.

Our final contribution is to extend an approach adopted in the ROC method so as to make it applicable to classifiers beyond those with generative models. Any type of classifier, such as those with discriminative models or discriminant function, can be modified so as to make fair decisions using this extension technique.

Our contributions are summarized as follows:

– We propose notions of model-based and actual independence, the difference between which is an essential factor for improving trade-offs between the fairness and accuracy of fairness-aware classifiers.
– We empirically show that the fairness of classifiers was drastically improved by modifying them to satisfy actual independence. This fact validates the importance of the difference between the two types of independence.
– We extend an approach adopted in the ROC method so as to make it applicable to any type of classifiers.

This paper is organized as follows. In Sect. 2, we briefly review the task of fairness-aware classification. In Sect. 3, after introducing the CV2NB method, we examine the reasons for the superiority of the CV2NB method and propose notions of model-based and actual independence. In Sects. 4 and 5, we respectively modify a prejudice remover regularizer and the ROC method so as to satisfy actual independence. We also show an extension of the ROC method in Sect. 5. Section 6 empirically shows the superiority of classifiers satisfying an actual independence condition, which validates our hypothesis that the effects of model bias and a decision rule are significant. Section 7 covers related work, and Sect. 8 concludes our paper.

## 2 Fairness-aware classification

This section summarizes the concept of fairness-aware classification. Following the definitions of notations and tasks, we introduce a formal notion of fairness.

### 2.1 Notations and task formalization

The goal of fairness-aware data mining is to analyze data while taking into account potential issues of fairness. Formal tasks of fairness-aware data mining can currently be classified into two groups: unfairness discovery and unfairness prevention (Ruggieri et al. 2010). We here focus on fairness-aware classification, which is a major task of unfairness prevention. The goal of *fairness-aware classification* is to categorize data while simultaneously taking into account issues or potential issues of fairness, discrimination, neutrality, and independence. Three types of variables, $Y$, $\mathbf{X}$, and $S$, are considered in fairness-aware classification. The random variables $S$ and $\mathbf{X}$ denote a *sensitive feature* and a set of *non-sensitive features*, respectively. A sensitive feature represents information with respect to which fairness must be maintained. For example, in the case of avoiding discrimination in credit decisions, a sensitive feature might correspond to gender, religion, race, or some other characteristic specified from a social or legal viewpoint, and credit decisions must be fair in terms of these features. Non-sensitive features, $\mathbf{X}$, consist of all other features. $\mathbf{X}$ is composed of $m$ random variables, $X^{(1)}, \ldots, X^{(m)}$. The random variable $Y$ denotes a *class variable* that represents a class, such as the result of a credit decision.

In this paper, we restrict the types of random variables because many problems of fairness in data mining are still unsolved even for such a restricted and simple case. A class variable $Y$ represents a binary class. The class, 0 or 1, signifies an unfavorable

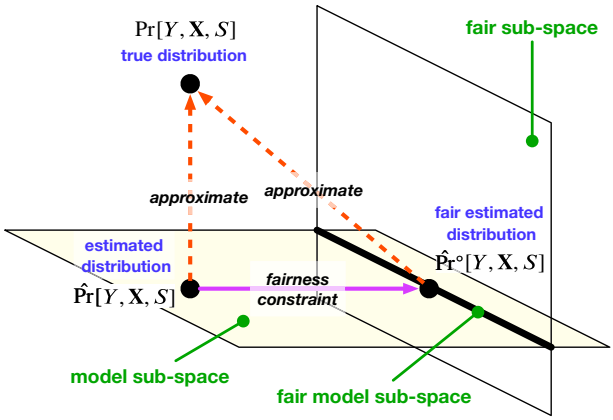


**Fig. 1** Geometrical view of distributions over $(Y, \mathbf{X}, S)$

or favorable outcome, such as denial or approval of a credit request, respectively. $S$ is also restricted to a binary variable. An object whose sensitive value is 1 or 0 is said to be in a *non-protected* or *protected* state, respectively. A protected object represents an individual or entity that should be protected from socially unfair treatment. The group of all objects that correspond to individuals who are in a protected state constitutes a protected group, and the rest of the objects comprise an unprotected group. The above assumptions are rather restrictive in terms of sensitive features, but even in this restricted and simplified case, the problem of accuracy–fairness trade-offs is not fully resolved. In addition, even if a sensitive feature is single and binary, a fairness-aware classifier can be applied to follow a specific regulation, such as the EU Racial Equality Directive. The extension to cases in which a sensitive feature is multivariate and/or continuous is a problem for future discussion.

We next define notations of probability distributions over the space $(Y, \mathbf{X}, S)$. Figure 1 depicts a geometrical view of the distributions. We first introduce distributions that are also managed in a standard machine learning process. These distributions are depicted in the left half of Fig. 1. Each object is represented by a pair of instances, $(\mathbf{x}, s)$, which are generated from a true distribution. Given the object, the corresponding class instance value, $y$, is generated from a true distribution, $\Pr[Y|\mathbf{X}{=}\mathbf{x}, S{=}s]$. It should be noted that this true distribution may lead to a potentially unfair decision that depends on a sensitive feature, $S$. The true joint distribution, $\Pr[Y, \mathbf{X}, S]$, is in a family of all distributions over $(Y, \mathbf{X}, S)$, which corresponds to the entirety of Fig. 1. We cannot know the true distribution itself, but we can observe data sampled from the true distribution. These data comprise a (training) dataset, $\mathcal{D} = \{(y_i, \mathbf{x}_i, s_i)\}$, $i = 1, \ldots, n$. We additionally define $\mathcal{D}_s$ as a subset that consists of all the data in $\mathcal{D}$ whose sensitive value is $s$. A family of model distributions, $\hat{\Pr}[Y, \mathbf{X}, S]$, is also given. Joint model distributions are on a model sub-space, depicted by a horizontal plane in Fig. 1. Examples of model distributions are naive Bayes or logistic regression. Note that because the true distribution is not on the model sub-space in general, the problem of model bias arises, as we will discuss in Sect. 3.2.1. Given a training dataset, the goal of the standard classification problem is to specify the model distribution that would best

approximate a true distribution among all candidate model distributions on the model sub-space.

Next, we turn to distributions that are particularly required to maintain fairness in classification. A fairness constraint is assumed to be formally specified, and a set of all distributions satisfying the fairness constraint constitutes a fair sub-space, $\Pr^{\circ}[Y, \mathbf{X}, S]$, depicted by a vertical plane in Fig. 1. In this paper, we mainly discuss a fairness constraint formalized as unconditional independence between a class variable, $Y$, and a sensitive feature, $S$, as in the next Sect. 2.2. In this case, a fair sub-space is equivalent to a set of all distributions satisfying the independence condition. The intersection of the fair sub-space and a model sub-space is a fair model sub-space, which consists of all candidate estimated fair distributions, $\hat{\Pr}^{\circ}[Y, \mathbf{X}, S]$, as depicted by a thick line in Fig. 1. Given a training dataset, the goal of fairness-aware classification is to find the fair model distribution that would best approximate a true distribution among all candidate distributions on the fair model sub-space.

### 2.2 Fairness in classification

Here we review formal definitions of fairness in classification. Though many types of fairness have been proposed, we will highlight a few representative examples. First, conditional independence, $Y \perp\!\!\!\perp S \mid \mathbf{X}$, corresponds to the simple elimination of a sensitive feature. Note that $A \perp\!\!\!\perp B$ denotes the (unconditional) independence between variables $A$ and $B$, and $A \perp\!\!\!\perp B \mid C$ denotes the conditional independence between $A$ and $B$ given $C$. The simple elimination of a sensitive feature from prediction models is insufficient for avoiding an inappropriate determination process because of the indirect influence of sensitive information. Such a phenomenon is called a *red-lining effect* (Calders and Verwer 2010). An example of a red-lining effect in online ad delivery has been reported (Sweeney 2013). When a full name is used as a query for a Web search engine, online ads with words indicating arrest records will be more frequently displayed for first names that are more common among individuals of African descent than individuals of European descent. In this delivery system, no information about the race or actual first name of users is exploited intentionally. Rather, the online ads are unfairly delivered as the result of automatic optimization of the click-through rate based on the feedback of users.

We next focus on unconditional independence, $Y \perp\!\!\!\perp S$. This condition must be satisfied to avoid the red-lining effect, as shown below. Consider a simple regression case such that $Y = X + \epsilon_X$ and $S = X + \epsilon_S$, where $\epsilon_X$ and $\epsilon_S$ are mutually independent Gaussian noises. A condition $Y \perp\!\!\!\perp S \mid X$ is satisfied because Gaussian noises, $\epsilon_X$ and $\epsilon_S$, are independent if $X$ is observed. However, the red-lining effect is caused because both variables, $Y$ and $S$, depend on a common variable, $X$. As observed in this example, $Y$ and $S$ must not depend on any common variables, and thus unconditional independence $Y \perp\!\!\!\perp S$ must be satisfied, to avoid the red-lining effect. We would like to note that this fairness condition implies the assumption that class labels of a training dataset may be unfair or unreliable due to unfavorable decisions that have been made for people in a protected group. Fairness conditions which assume that training labels are fair have been discussed in Hardt et al. (2016), Zafar et al. (2017).

To represent a fairness constraint in formulae, a fairness index to measure the degree of fairness, such as $Y \perp\!\!\!\perp S$, is introduced. Many types of fairness indices have been proposed: discrimination score (Calders and Verwer 2010), mutual information (Kamishima et al. 2012), $\chi^2$-statistics (Berendt and Preibusch 2012; Sweeney 2013), $\eta$-neutrality (Fukuchi et al. 2013), neutrality risk (Fukuchi and Sakuma 2014), and a combination of statistical parity and the Lipschitz condition (Dwork et al. 2012; Zemel et al. 2013). Note that a previously published tutorial (Hajian et al. 2016) provides a good survey of these indices. If these fairness indices are worse than a pre-specified level, the corresponding decisions are considered unfair.

## 3 Analysis of fairness in classification

We first review the CV2NB method, which achieves a better accuracy–fairness trade-off, as shown in experimental Sect. 6.2. We then hypothesize that this superiority is due to the effects of model bias and a deterministic decision rule being taken into account. Based on this hypothesis, we here formalize the notions of model-based independence and actual independence.

### 3.1 Calders and verwer's two-naive-bayes

We introduce Calders and Verwer's two-naive-Bayes method (CV2NB) (Calders and Verwer 2010), which achieves better trade-offs between accuracy and fairness than other fairness-aware classifiers. The generative model of this method is

$$\hat{\Pr}[Y, \mathbf{X}, S] = \hat{\Pr}[Y|S]\,\hat{\Pr}[S]\prod_k \hat{\Pr}[X^{(k)}|Y, S]. \tag{1}$$

In a standard naive Bayes model, each $X^{(k)}$ depends only on $Y$; in the CV2NB model, it also depends on $S$. Note that this method was named "two-naive-Bayes" because it is as if a distinct naive Bayes classifier is learned for each sensitive value. To make classification fair, a joint distribution $\hat{\Pr}[Y, S] = \hat{\Pr}[Y|S]\,\hat{\Pr}[S]$ is modified by the post-processing algorithm shown in Algorithm 1, and the modified distribution is denoted by $\hat{\Pr}^\circ[Y, S]$. After the algorithm is stopped, a model parameter $\hat{\Pr}^\circ[y, s]$ can be induced from $N(y, s)$, $y, s \in \{0, 1\}$, which are the virtual counts of data of $Y=y$ and $S=s$. A fair model distribution can be obtained by replacing $\hat{\Pr}[Y|S]\,\hat{\Pr}[S]$ in Eq. (1) with the distribution $\hat{\Pr}^\circ[Y, S]$.

This post-processing algorithm was designed to modify the original model so as to satisfy two conditions: (a) fairness in classification, and (b) preservation of a class distribution. First, to satisfy the fairness condition, the post-processing algorithm adopts Calders-Verwer's discrimination score (CVS) as a fairness index. This score is defined by subtracting the probability that protected objects will get favorable treatment from the probability that unprotected objects will:

$$\mathrm{CVS}(Y, S) = \hat{\Pr}[Y{=}1|S{=}1] - \hat{\Pr}[Y{=}1|S{=}0]. \tag{2}$$

**Algorithm 1** A post-processing algorithm for a CV2NB model

**Require:** $N(Y, S)$ (the counts of samples of $Y{=}y$ and $S{=}s$ in training data)
1: $disc \leftarrow$ a CVS of the predicted classes by the current model
2: **while** $disc > 0$ **do**
3: $numpos \leftarrow$ the number of positively classified samples by the current model
4: **if** $numpos <$ the number of positive samples in $\mathcal{D}$ **then**
5: $N(Y{=}1, S{=}0) \leftarrow N(Y{=}1, S{=}0) + \Delta N(Y{=}0, S{=}1)$
6: $N(Y{=}0, S{=}0) \leftarrow N(Y{=}0, S{=}0) - \Delta N(Y{=}0, S{=}1)$
7: **else**
8: $N(Y{=}0, S{=}1) \leftarrow N(Y{=}0, S{=}1) + \Delta N(Y{=}1, S{=}0)$
9: $N(Y{=}1, S{=}1) \leftarrow N(Y{=}1, S{=}1) - \Delta N(Y{=}1, S{=}0)$
10: **if** Any entry of $N(Y, S)$ is negative **then**
11: cancel the previous update of $N(Y, S)$ and abort
12: Recalculate $\hat{\Pr}[Y|S]$, a CVS, and $disc$, based on updated $N(Y, S)$

NOTE: $\Delta$ is a small positive parameter and was set to 0.01 as in the original paper. We slightly modified the original algorithm by adding lines 10–11, which guarantees that $N(Y, S)$ will be non-negative.

Note that $\hat{\Pr}[Y{=}1|S{=}s]$ is obtained by marginalizing $\hat{\Pr}[Y{=}1|\mathbf{X}, S{=}s]\hat{\Pr}[\mathbf{X}|S{=}s]$ over $\mathbf{X}$. It is easy to show that when both $Y$ and $S$ are binary, the zero CVS implies that $Y$ and $S$ are unconditionally independent, $Y \perp\!\!\!\perp S$. Lines 5–6 and 8–9 in Algorithm 1 are designed so that the CVS of the resulting distribution approaches zero. The main loop of this algorithm exits at line 2 if the resultant CVS is closer to zero than a small threshold. Therefore, the resulting distribution $\hat{\Pr}^{\circ}[Y, S]$ satisfies the independence condition between $Y$ and $S$. In terms of the second condition, the modified class distribution is kept close to the original one, i.e., $\hat{\Pr}^{\circ}[Y] \approx \hat{\Pr}[Y]$ in line 4. However, because the marginal distribution of $Y$ is not considered in the stopping criterion in line 2, the resultant distribution of $Y$ does not always equal the sample distribution of $Y$.

As proved in our experimental Sect. 6, the CV2NB method is highly efficient; that is to say, this classifier can precisely and fairly predict class labels. We next discuss the reason for this superiority.

### 3.2 Why is the CV2NB method superior?

CV2NB tends to achieve better trade-offs between accuracy and fairness, even though the other models explicitly impose fairness constraints. We hypothesized two reasons for this. The first is model bias, which makes an estimated distribution different from a true distribution. The second reason is a deterministic decision rule. Though class labels are in fact chosen according to a deterministic decision rule, non-CV2NB methods assume that the labels are probabilistically generated.

#### *3.2.1 Model bias*

We first analyze how model bias damages fairness. In the non-CV2NB cases, class labels are predicted based on an *estimated* distribution, $\hat{\Pr}[Y|\mathbf{X}, S]$, while the objects to be classified are generated according to a *true* distribution, $\Pr[\mathbf{X}, S]$. The estimated

distribution is generally different from the true distribution because the estimated distribution must lie in the model sub-space; this restriction is not relevant to a true distribution. When learning models, random variables following *estimated* distributions, $Y$ and $S$, are constrained to be independent, and a joint distribution over $(Y, \mathbf{X}, S)$ becomes $\hat{\Pr}[Y]\,\hat{\Pr}[S]\,\hat{\Pr}[\mathbf{X}|Y, S]$. Hence, the joint distributions over $(Y, \mathbf{X}, S)$ disagree between the case of learning models and that of making a prediction as follows:

$$\hat{\Pr}[Y|\mathbf{X}, S]\Pr[\mathbf{X}, S] \neq \hat{\Pr}[Y]\,\hat{\Pr}[S]\,\hat{\Pr}[\mathbf{X}|Y, S]. \tag{3}$$

On the other hand, in the CV2NB case, the distribution of class labels is approximated by a sample mean. Specifically, in Algorithm 1, line 12, an empirical distribution, which approximates a true distribution, is adopted as a joint distribution of $Y$ and $S$. Therefore, the CV2NB method can avoid the effect of model bias on its fairness.

### *3.2.2 A deterministic decision rule*

We next discuss the effect of a deterministic decision rule on the choice of class labels. Independence between a class variable and a sensitive feature is satisfied if the distribution of actual class labels equals that induced from a probabilistic model. In other words, labels are assumed to be chosen probabilistically. However, this assumption is not the case because actual labels, $\tilde{y}$, are deterministically chosen by the following decision rule:

$$\tilde{y} = \arg\max_y \hat{\Pr}[Y = y|\mathbf{X} = x, S = s]. \tag{4}$$

We next examine how greatly the distribution of actual class labels determined by a decision rule diverges from that of labels probabilistically generated by a prediction model. We here consider a very simple model with a binary class variable, $Y$, and one binary feature variable, $X$. The class prior distribution follows a discrete uniform distribution, i.e., $\hat{\Pr}[Y{=}1] = 0.5$. Two other parameters, $\hat{\Pr}[X{=}1|Y{=}0]$ and $\hat{\Pr}[X{=}1|Y{=}1]$, are required to represent the joint distribution of $X$ and $Y$. In this case, $\mathrm{E}[Y]$ becomes a constant, 0.5, if $Y$ follows the distribution induced from this model. We then consider the variable $\tilde{Y}$ to represent actual labels determined by Eq. (4). In Fig. 2, we depict the variation of the expectation $\mathrm{E}[\tilde{Y}]$ according to the changes of $\hat{\Pr}[X{=}1|Y{=}0]$ and $\hat{\Pr}[X{=}1|Y{=}1]$. Surprisingly, the condition $\mathrm{E}[Y] = \mathrm{E}[\tilde{Y}]$ is satisfied only if $\hat{\Pr}[X{=}1|Y{=}0] + \hat{\Pr}[X{=}1|Y{=}1] = 1$ (depicted by the thick broken line in Fig. 2). As a result, the two variables $Y$ and $\tilde{Y}$ behave differently at almost every point.

We next demonstrate how heavily the difference between $Y$ and $\tilde{Y}$ worsens fairness in classification. To this end, we evaluate the degrees of two kinds of independence, $Y \perp\!\!\!\perp S$ and $\tilde{Y} \perp\!\!\!\perp S$. We use another simple generative model with a single non-sensitive variable:

$$\Pr[Y, X, S] = \Pr[X|Y, S]\Pr[Y]\Pr[S]. \tag{5}$$

Clearly, $Y$ and $S$ are mutually independent. All variables are binary, and we fixed the parameters: $\Pr[S{=}1] = 0.9$, and

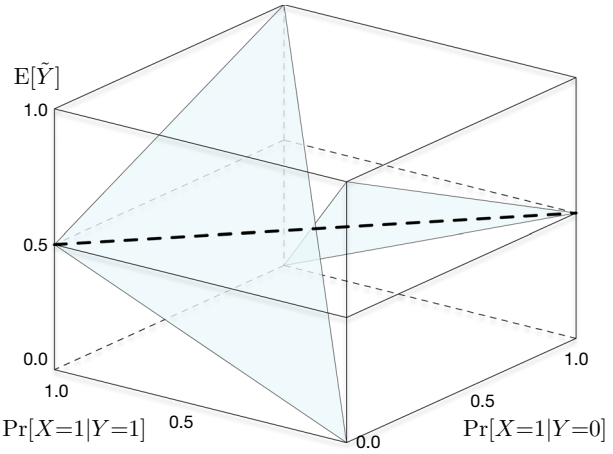


**Fig. 2** Changes in the expectation of actual labels, E[$\tilde{Y}$]

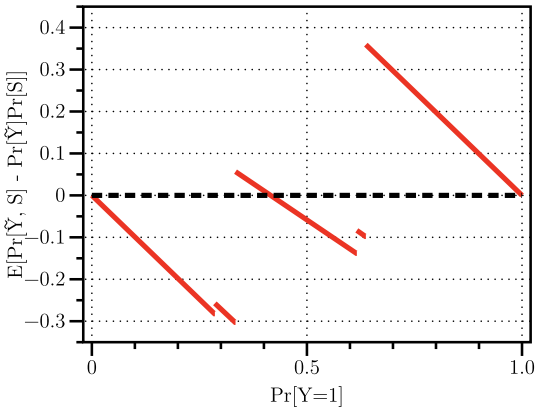


**Fig. 3** Degrees of independence according to the changes of Pr[$Y$=1]

$$\Pr[X{=}1|Y{=}0,\, S{=}0] = 0.2 \qquad \Pr[X{=}1|Y{=}0,\, S{=}1] = 0.3$$
$$\Pr[X{=}1|Y{=}1,\, S{=}0] = 0.5 \qquad \Pr[X{=}1|Y{=}1,\, S{=}1] = 0.4.$$

The last parameter, Pr[$Y$=1], was changed from 0 to 1. The expectation of differences, E[Pr[$Y$, $S$] − Pr[$Y$] Pr[$S$]], is used to evaluate the degree of independence between $S$ and $Y$, a probabilistically generated class. The expectation is constantly zero due to the independence property between $Y$ and $S$, irrespective of the value of Pr[$Y$=1]. We next examine the independence between $S$ and $\tilde{Y}$, which represents a class label obtained by the application of Eq. (4); the expectation E[Pr[$\tilde{Y}$, $S$] − Pr[$\tilde{Y}$] Pr[$S$]] is plotted in Fig. 3. This figure shows that $\tilde{Y}$ is independent of $S$ at only three points. This is in strong contrast to the stationary independence between $Y$ and $S$ when class labels are probabilistically generated. This example proves that considering the influence of a deterministic decision rule is essential for fairness in classification.

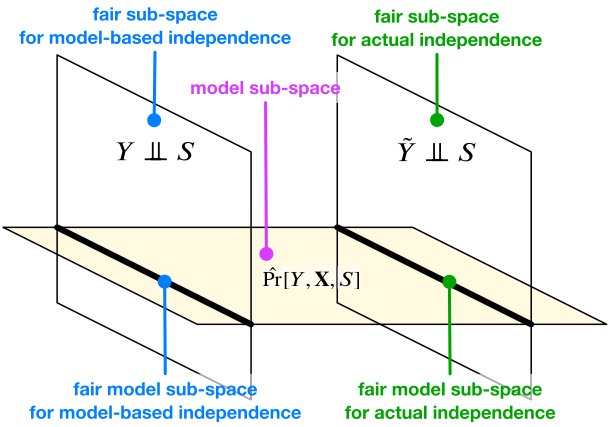


**Fig. 4** Geometrical view of two types of independence

Non-CV2NB methods adopt an assumption that class labels are probabilistically generated, and the effect of a decision rule is ignored. In the CV2NB method, fair labels are determined and maintained based on the independence of sensitive features from actual labels, which is not possible for labels induced from a probabilistic prediction model.

### 3.3 Model-based independence and actual independence

Based on the above discussion of the influences of model bias and a deterministic decision rule, we here formalize the notions of model-based independence and actual independence. Figure 4 shows the sub-spaces required for these two types of independence. A common model sub-space, $\hat{\Pr}[Y, \mathbf{X}, S]$, depicted by the horizontal plane in the figure, is shared in both types of independence. On the other hand, as depicted by the two vertical planes in the figure, there are two distinct fair sub-spaces. The two fair sub-spaces are the same from the standpoint that they satisfy unconditional independence between a class variable and a sensitive feature, but their distributions generating class labels differ. In the case of model-based independence, class labels are directly generated from a distribution on the model-subspace. However, in the case of actual independence, class labels are generated from a distribution induced by taking into account the influence of model bias and a decision rule in the real world. For each type of independence, we provide a procedure to derive the distributions generating class labels in cases of classifiers with a generative model and a discriminative model (Bishop 2006, Sect. 1.5.4).

#### *3.3.1 Model-based independence*

The constraint of *model-based independence* is defined as independence between a class variable and a sensitive feature, and class labels are generated from a model distribution on a model sub-space. Formally, the constraint is defined as

$$Y \perp\!\!\!\perp S, \text{ where } (Y, S) \sim \hat{\Pr}^{\circ}[Y, S]. \tag{6}$$

$\hat{\Pr}^{\circ}[Y, S]$ is directly induced by marginalizing a model distribution, $\hat{\Pr}[Y, \mathbf{X}, S]$, over $\mathbf{X}$. We show the details of this marginalization process for the cases in which a classifier is a generative model or a discriminative model.

We first show the case in which the classifier is a generative model, whose joint distribution of a class and features, $\hat{\Pr}[Y, \mathbf{X}, S]$, is given. Non-sensitive features, $\mathbf{X}$, are marginalized by integrating out from the joint distribution, and we get

$$\hat{\Pr}^{\circ}[Y, S] = \int_{\mathbf{x} \in \mathrm{dom}(\mathbf{X})} \hat{\Pr}[Y, \mathbf{x}, S] d\mathbf{x}. \tag{7}$$

In this generative case, the influence of model bias and a deterministic rule is not considered, as it was in Sect. 3.2.

We next turn to a discriminative model, in which a conditional distribution, $\hat{\Pr}[Y|\mathbf{X}, S]$, is directly parameterized. We want to obtain a joint distribution, $\hat{\Pr}[Y, \mathbf{X}, S]$, but this is impossible due to the lack of a model for the distribution of $\mathbf{X}$ and $S$. Hence, a sample mean is used for approximating the expectation over $\mathbf{X}$,

$$\hat{\Pr}^{\circ}[Y, S] \approx \frac{|\mathcal{D}_s|}{n} \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x} \in \mathcal{D}_s} \hat{\Pr}[Y|\mathbf{X}{=}\mathbf{x}, S{=}s] = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{D}_s} \hat{\Pr}[Y|\mathbf{X}{=}\mathbf{x}, S{=}s]. \tag{8}$$

Because we use a sample mean approximating the true distribution in this discriminative case, the model bias is removed, and only the influence of a decision rule is ignored.

As we will show in Sect. 6, classifiers satisfying this model-based independence are poor in fairness evaluation indexes; this is due to unrealistic assumptions. Model-based independence can be considered as a valid fairness constraint. However, the assumptions adopted in this constraint don't match the practical use of classifiers. Specifically, this constraint is assumed to ignore the influences of model bias and a deterministic decision rule, as discussed in the previous section. Therefore, we introduce another constraint based on a more realistic assumption.

### *3.3.2 Actual independence*

The constraint of *actual independence* is the same as that of a model-based independence in the respect that they are both independence constraints between a class variable and a sensitive feature. The key difference lies in the distributions used to generate class labels. Specifically, class labels are generated not from a model distribution, $\hat{\Pr}[Y, \mathbf{X}, S]$, but from another distribution induced from the model distribution. The induced distribution is designed by taking into account the influences of model bias and a decision rule in the real world. The constraint of actual independence is formally defined as:

$$\tilde{Y} \perp\!\!\!\perp S, \text{ where } (\tilde{Y}, S) \sim \hat{\Pr}^{\circ}[\tilde{Y}, S]. \tag{9}$$

A deterministic class variable, $\tilde{Y}$, is generated from a distribution, $\hat{\Pr}^{\circ}[\tilde{Y}, S]$, which is induced from a model distribution. Below, we describe the details of the method used to induce the distribution, $\hat{\Pr}^{\circ}[\tilde{Y}, S]$, in the cases of a generative model and a discriminative model.

We begin with the case of a generative model. We design $\hat{\Pr}^{\circ}[\tilde{Y}, S]$ so that it can consider the influence of model bias and a decision rule. $\hat{\Pr}^{\circ}[\tilde{Y}, S]$ is derived from $\hat{\Pr}[\tilde{Y}, \mathbf{X}, S]$. To remove the model bias, we avoid the use of a given model distribution, $\hat{\Pr}[Y, \mathbf{X}, S]$. As discussed in Sect. 3.2.1, model bias is problematic because of the difference between the distributions used in the learning and prediction stages. Hence, we adopt a distribution used in the prediction stage, $\hat{\Pr}[\tilde{Y}, \mathbf{X}, S] = \hat{\Pr}[\tilde{Y}|\mathbf{X}, S] \Pr[\mathbf{X}, S]$, which is the left-hand side of Eq. (3). Expectation over the true distribution of $\mathbf{X}$ is approximated by a sample mean as in Eq. (8):

$$\hat{\Pr}^{\circ}[\tilde{Y}, S] = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{D}_s} \hat{\Pr}[\tilde{Y}|\mathbf{X}{=}\mathbf{x}, S{=}s]. \tag{10}$$

All that we have to do is induce the distribution, $\hat{\Pr}[\tilde{Y}|\mathbf{X}, S]$, to generate deterministic class labels from a model distribution. Here, because we have to remove the influence of a decision rule, this distribution is obtained by applying a decision rule:

$$\begin{cases} \hat{\Pr}[\tilde{Y}{=}1|\mathbf{x}, s] = \begin{cases} 1, & \text{if } \hat{\Pr}[Y{=}1, \mathbf{x}, s] \geq \hat{\Pr}[Y{=}0, \mathbf{x}, s] \\ 0, & \text{otherwise} \end{cases} \\ \hat{\Pr}[\tilde{Y}{=}0|\mathbf{x}, s] = 1 - \hat{\Pr}[\tilde{Y}{=}1|\mathbf{x}, s] \end{cases}, \tag{11}$$

where $\hat{\Pr}[Y, \mathbf{X}, S]$ is a generative model on a model sub-space.

In the case of a discriminative model, the derivation procedure of $\hat{\Pr}^{\circ}[\tilde{Y}, S]$ is the same as for the generative model, except that $\hat{\Pr}[\tilde{Y}|\mathbf{X}, S]$ is not obtained by Eq. (11). The distribution is again derived by applying a decision rule:

$$\begin{cases} \hat{\Pr}[\tilde{Y}{=}1|\mathbf{x}, s] = \begin{cases} 1, & \text{if } \hat{\Pr}[Y{=}1|\mathbf{x}, s] \geq \hat{\Pr}[Y{=}0|\mathbf{x}, s] \\ 0, & \text{otherwise} \end{cases} \\ \hat{\Pr}[\tilde{Y}{=}0|\mathbf{x}, s] = 1 - \hat{\Pr}[\tilde{Y}{=}1|\mathbf{x}, s] \end{cases}, \tag{12}$$

where $\hat{\Pr}[Y|\mathbf{X}, S]$ is a discriminative model on a model sub-space. Note that the distributions including $\tilde{Y}$ are not members of a fair sub-space, but these distributions exist somewhere in a space represented by Fig. 4. These distributions are merely exploited to examine the independence between $\tilde{Y}$ and $S$. A fair sub-space for actual independence consists of all distributions over $(Y, \mathbf{X}, S)$ that are used to induce $\hat{\Pr}[Y|\mathbf{X}, S]$ in Eqs. (11) and (12), and the induced distributions satisfy the condition (9).

As described above, the key difference between the two types of fairness constraints, model-based independence and actual independence, is the difference in the distributions to generate class labels. In order to show that the difference of these fairness constraints is important for fairness-aware classification, we then modify two existing fairness-aware classifiers so as to satisfy these fairness constraints.

## 4 A prejudice remover regularizer

We introduce a prejudice remover regularizer that constrains a model-based independence condition. This term is then modified so as to satisfy an actual independence constraint.

We first describe an original form of *logistic regression with a prejudice remover regularizer* (Kamishima et al. 2012) (a PR method, for short). An objective function of this method is derived by adding a constraint term enhancing the fairness to an objective function of logistic regression. Logistic regression is a prediction model:

$$\hat{\Pr}[y|\mathbf{x};\mathbf{w}] = y\,\mathrm{sig}(\mathbf{x}^\top\mathbf{w}) + (1-y)(1-\mathrm{sig}(\mathbf{w}^\top\mathbf{x})), \tag{13}$$

where $\mathrm{sig}(\cdot)$ is a sigmoid function and $\mathbf{w}$ is a weight parameter vector. To develop a prediction model that is dependent on a sensitive feature, a logistic regression model is used for each value of the sensitive feature:

$$\hat{\Pr}[y|\mathbf{x},s] = \hat{\Pr}[y|\mathbf{x};\mathbf{w}^{(s)}].$$

Weight parameters are required for each sensitive value, $\mathbf{w}^{(s)},\ s \in \{0,1\}$. In the PR method, two types of regularizers are adopted. The first regularizer is an $L_2$ regularizer, $\|\mathbf{\Theta}\|_2^2$, to avoid over-fitting. The second regularizer, $\mathrm{R}_{\mathsf{PR}}(Y,S)$, is introduced to enforce fairness. By adding these two regularizers to a negative log-likelihood, the objective function to minimize is obtained:

$$\mathrm{loss}(\{\mathbf{w}^{(s)}\};\mathcal{D}) = -\mathcal{L}(\mathcal{D}) + \eta\,\mathrm{R}_{\mathsf{PR}}(Y,S) + \frac{\lambda}{2}\sum_s \|\mathbf{w}^{(s)}\|_2^2, \tag{14}$$

where $\lambda$ and $\eta$ are positive regularization parameters, and a log-likelihood function is

$$\mathcal{L}(\mathcal{D}) = \textstyle\sum_{(y_i,\mathbf{x}_i,s_i)\in\mathcal{D}} \ln \hat{\Pr}^\circ[y_i|\mathbf{x}_i,s_i].$$

In the case of the original PR method that is designed to satisfy a model-based independence, mutual information between $Y$ and $S$ is used as a prejudice remover regularizer, because the smaller mutual information indicates a higher level of independence. An original prejudice remover is defined as

$$\mathrm{R}_{\mathsf{PR\text{-}MI}}(Y,S) = \sum_{Y,S} \hat{\Pr}^\circ[Y,S] \ln \frac{\hat{\Pr}^\circ[Y,S]}{\hat{\Pr}^\circ[Y]\,\hat{\Pr}^\circ[S]}. \tag{15}$$

Because logistic regression is a discriminative model and we are now trying to satisfy a model-based condition, we use Eq. (8) as $\hat{\Pr}^\circ[Y,S]$. The other distributions, $\hat{\Pr}^\circ[Y]$ and $\hat{\Pr}^\circ[S]$, can be derived from $\hat{\Pr}^\circ[Y,S]$. This regularizer is analytically differentiable, and we used a conjugate gradient method for optimizing an objective function (14).

We then modify this original prejudice remover so as to satisfy an actual independence constraint. For this purpose, we consider the independence between $Y$ and $S$ following $\hat{\Pr}^\circ[\tilde{Y},S]$. The modified prejudice remover is defined as

$$\mathrm{R}_{\mathsf{PR\text{-}AI}}(Y, S) = \sum_{Y,S} \hat{\Pr}^{\circ}[\tilde{Y}, S] \ln \frac{\hat{\Pr}^{\circ}[\tilde{Y}, S]}{\hat{\Pr}[\tilde{Y}]\hat{\Pr}[S]}. \tag{16}$$

A joint distribution $\hat{\Pr}^{\circ}[\tilde{Y}, S]$ can be derived by Eqs. (10) and (12). As we will demonstrate in experimental Sect. 6, this small modification is helpful for realizing a drastic improvement in fairness. Unfortunately, this modified prejudice remover is not differentiable due to a discrete transformation in Eq. (12). Therefore, to optimize the objective function, we used a Powell method, which is applicable without computing gradients. The original and modified method are abbreviated as PR-MI and PR-AI, respectively.

## 5 Reject-option-based classification

Kamiran et al. proposed a method, *reject option-based classification* (ROC), to change decision thresholds for making fairer classification (Kamiran et al. 2012). After reviewing the original ROC method, we show how to select decision thresholds to satisfy model-based and actual independence for a naive Bayes case. We then extend our method so as to make it applicable to classifiers other than those with a generative model.

### 5.1 The original ROC method

Kamiran et al. discussed a theory for determining class labels based on a class posterior distribution so that a fairness constraint was satisfied (Kamiran et al. 2012). In standard classification, objects are classified to class 1 if the class posteriors satisfy the inequality $\hat{\Pr}[Y{=}1|\mathbf{X}] \geq \hat{\Pr}[Y{=}0|\mathbf{X}]$, which is equivalent to $\hat{\Pr}[Y{=}1|\mathbf{X}] \geq 1/2$. The threshold $1/2$ is referred to as a decision threshold, and it is modified to make the decisions fair. Given a threshold parameter, $1 > \tau \geq 1/2$, objects such that $S{=}0$ are classified to class 1 if $\hat{\Pr}[Y{=}1|\mathbf{X}, S{=}0] \geq 1 - \tau$. Inversely, objects such that $S{=}1$ are classified to class 1 if $\hat{\Pr}[Y{=}1|\mathbf{X}, S{=}1] \geq \tau$.

The authors pointed out the connection between this decision rule and cost-sensitive learning (Elkan 2001). The goal of cost-sensitive learning is to classify objects so that their misclassification costs are minimized. When classifying an object, a misclassification cost is a penalty that is added when an estimated class of the object is different from its true class. We turn to the ROC case. For objects such that $S{=}0$, the costs of misclassifying objects whose true classes are 0 are held to 1, but those of misclassifying objects whose true classes are 1 are increased to $\tau/(1-\tau)$. Non-protected objects are treated inversely. This connection between a ROC method and cost-sensitive learning reveals that changing a decision threshold is equivalent to changing the prior distributions. Elkan's theorem 2 in Elkan (2001) asserts the following relation. Given a Bayesian classifier whose prior is $b'$ and whose decision threshold is $p'$, when this prior is changed to $b$, how should we choose a new decision threshold, $p$, so as to make these two classifiers indicate the same decision? Elkan's theorem describes the relation as

$$p' = \frac{b'p(1-b)}{b - pb + b'p - bb'}. \tag{17}$$

According to this theorem, we can discuss adjusting priors instead of thresholds.

In the following subsections, we slightly generalize the original ROC method. Decision thresholds are changed symmetrically in the original method, but we relax this limitation. Specifically, the thresholds are changed to $\tau_0 \in (0, 1)$ for an $S = 0$ group, while they are changed to $\tau_1 \in (0, 1)$ for an $S = 1$ group.

### 5.2 A ROC method satisfying model-based independence

We here describe how to select priors for achieving model-based independence when targeting a naive Bayes classifier. We first define a naive Bayes model satisfying a model-based independence constraint, and parameters of the model are estimated by maximizing a likelihood. We then show that this method corresponds to a special case of the ROC method.

We modify a mixture of two-naive-Bayes models to satisfy a model-based independence constraint, and estimate its parameters. This is the mixture model, which is equivalent to Eq. (1):

$$\hat{\Pr}[Y, \mathbf{X}, S] = \hat{\Pr}[Y|S]\,\hat{\Pr}[S]\,\hat{\Pr}[\mathbf{X}|Y, S]. \tag{18}$$

To satisfy a model-based independence constraint Eq. (6), we replace a class prior so as to make a class variable independent from a sensitive feature, and we get:

$$\hat{\Pr}^{\circ}[Y, \mathbf{X}, S] = \hat{\Pr}^{\circ}[Y]\,\hat{\Pr}^{\circ}[S]\,\hat{\Pr}^{\circ}[\mathbf{X}|Y, S]. \tag{19}$$

It is very easy to derive the maximum likelihood estimators of a model (19) from a training dataset $\mathcal{D}$ if both $Y$ and $S$ are binary, by simply counting the data in a training dataset. Note that we adopt a Laplace smoothing technique to avoid the zero-counting problem in later experiments. We abbreviate this method as ROCNB-MI.

We then clarify that this method is a special case of the ROC method. Equation (18) can be interpreted as a mixture of two naive Bayes models, each of which is learned separately for the respective sensitive value. Furthermore, because only a class prior is changed between models (18) and (19), the remaining parameters are unchanged:

$$\hat{\Pr}^{\circ}[S{=}s] = \hat{\Pr}[S{=}s],\ \hat{\Pr}^{\circ}[X^{(k)}|Y, S{=}s] = \hat{\Pr}[X^{(k)}|Y, S{=}s],\ s \in \{0, 1\},\ k \in \{1, \ldots, m\}.$$

As a result, model (19) can be obtained from model (18) by replacing priors $\hat{\Pr}[Y|S]$ with $\hat{\Pr}^{\circ}[Y]$. Note that, as in related work Sect. 7, we are generally required to assume that a fair class is determined independently from $\mathbf{X}$ in a post-process case such as this ROC method, but these equalities automatically hold because parameters other than priors are unchanged. According to Elkan's theorem Eq. (17), this is equivalent to changing decision thresholds $1/2$ to

**Algorithm 2** A ROC naive Bayes method to satisfy an actual independence constraint

**Require:** Learned parameters of standard classifiers: $\hat{\Pr}[Y|S]$, $\hat{\Pr}[S]$, $\hat{\Pr}[\mathbf{X}|Y, S]$, training dataset: $\mathcal{D}$
1: $\hat{\Pr}^\circ[S] \leftarrow \hat{\Pr}[S]$, $\hat{\Pr}^\circ[\mathbf{X}|\tilde{Y}, S] \leftarrow \hat{\Pr}[\mathbf{X}|Y, S]$ ▷ copy unchanging parameters
2: $bestLikelihood \leftarrow -\infty$
3: **for all** $\hat{\Pr}^\circ[\tilde{Y}]' \in$ {candidate values for $\hat{\Pr}^\circ[\tilde{Y}]$} **do**
4: Find values of parameters, $\hat{\Pr}^\circ[\tilde{Y}|s]'$, to satisfy $\hat{\Pr}^\circ[\tilde{Y}|s]' \approx \hat{\Pr}^\circ[\tilde{Y}]'$ for the respective sensitive value
5: $likelihood \leftarrow$ likelihood of a temporal model, $\hat{\Pr}^\circ[\tilde{Y}|S]' \hat{\Pr}^\circ[S] \hat{\Pr}^\circ[\mathbf{X}|\tilde{Y}, S]$, over $\mathcal{D}$
6: **if** $likelihood \geq bestLikelihood$ **then**
7: $bestLikelihood \leftarrow likelihood$
8: $\hat{\Pr}^\circ[\tilde{Y}|S] \leftarrow \hat{\Pr}^\circ[\tilde{Y}|S]'$ ▷ update best parameter
9: Output parameters of a classifier: $\hat{\Pr}^\circ[\tilde{Y}|S]$, $\hat{\Pr}^\circ[S]$, $\hat{\Pr}^\circ[\mathbf{X}|\tilde{Y}, S]$

$$\tau_s = \frac{\hat{\Pr}[Y{=}1|s](1 - \hat{\Pr}^\circ[Y{=}1])}{\hat{\Pr}^\circ[Y{=}1] + \hat{\Pr}[Y{=}1|s] - 2\,\hat{\Pr}^\circ[Y{=}1]\,\hat{\Pr}[Y{=}1|s]}, \quad s \in \{0, 1\}. \tag{20}$$

It is concluded that this method can be considered as a special case of the ROC method.

### 5.3 A ROC method satisfying actual independence

We next present an approach for finding decision thresholds to achieve actual independence. As in the case of the above ROCNB-MI method, two naive-Bayes-classifiers are trained for each sensitive value, and we search for new priors that maximize likelihood under an actual independence constraint.

Algorithm 2 shows the outline of a ROC naive Bayes method for satisfying an actual independence constraint (a ROCNB-AI method). Fundamentally, this algorithm is designed to find the best parameters by a grid search under an actual independence constraint. Because only priors are changed, all parameters other than priors are copied (line 1). The distribution of a class label obtained by applying a deterministic decision rule, $\hat{\Pr}^\circ[\tilde{Y}]$, is temporally fixed (line 3). For the distribution, priors of naive Bayes, $\hat{\Pr}^\circ[\tilde{Y}|s]'$, are adjusted to satisfy the actual independence constraint Eq. (9) (line 4). Using the adjusted priors, the temporal likelihood is calculated (line 5) and is compared with the current best (line 6), and this algorithm finally outputs the best parameters (line 9).

We then give the details of the step for finding appropriate priors in line 4. To satisfy the condition specified by Eq. (9), for each sensitive value $s$, we must find a prior $\hat{\Pr}^\circ[\tilde{Y}|S{=}s]'$ so that an induced distribution from the prior well-approximates a given $\hat{\Pr}^\circ[\tilde{Y}]'$. This task is formalized as an optimization problem:

$$\hat{\Pr}^\circ[\tilde{Y}{=}1|S{=}s]' = \min_{\hat{\Pr}^\circ[Y=1|S=s]''} \left(\hat{\Pr}^\circ[\tilde{Y}{=}1|S{=}s]'' - \hat{\Pr}^\circ[\tilde{Y}{=}1]'\right)^2, \text{ for } s \in \{0, 1\}. \tag{21}$$

$\hat{\Pr}^\circ[\tilde{Y}{=}1|S{=}s]$ can be computed from a joint distribution $\hat{\Pr}^\circ[\tilde{Y}, S]$, which can be derived by Eqs. (10) and (11) in Sect. 3.3.2. Here, we use $\hat{\Pr}^\circ[Y|S]'' \hat{\Pr}^\circ[S] \hat{\Pr}^\circ[\mathbf{X}|\tilde{Y}, S]$ as a joint model distribution in Eq. (11). Note that the procedure of finding optimal priors was the same as that used in an actual fair-factorization in our preliminary work (Kamishima et al. 2013).

Finally, we should comment on the complexity of Algorithm 2. We begin with the complexity of the optimization task in line 4. If data are sorted according to the value,

$$\hat{\Pr}^{\circ}[Y{=}1|S]'' \, \hat{\Pr}^{\circ}[\mathbf{X}|\tilde{Y}{=}1, S] - \hat{\Pr}^{\circ}[Y{=}0|S]'' \, \hat{\Pr}^{\circ}[\mathbf{X}|\tilde{Y}{=}0, S],$$

in $O(n \log n)$ time at the beginning, the optimal priors can be found in constant time. The complexity of the main loop in line 3 depends on the size of a candidate set. The set is composed of values from 0 to 1 at intervals $1/n$, and the size of the set is $O(n)$. Putting all these facts together, the total complexity of Algorithm 2 becomes $O(n \log n + n) = O(n \log n)$.

### 5.4 A universal ROC method

Next, we will extend the applicable target of the concept of actual independence. There are three types of classifiers: a generative model, a discriminative model, and a discriminant function (Bishop 2006, Sect. 1.5.4). However, the approach in the previous section is only applicable to a classifier with a generative model. To relax this restriction, we developed a procedure, which we call the universal ROC method, to make the approach applicable to all three types of classifiers.

Before explaining this method, we must first show the concept of a classifier with a discriminant function. Decisions of classifiers depend on the sign of a discriminant function, $f(\mathbf{x})$. A classifier with a discriminative model, such as a logistic regression, directly expresses the posterior class probabilities. It determines a predicted class based on the sign of the discriminant function:

$$f(\mathbf{x}) = \hat{\Pr}[Y{=}1|\mathbf{X}{=}\mathbf{x}] - \hat{\Pr}[Y{=}0|\mathbf{X}{=}\mathbf{x}]. \tag{22}$$

The other type is a classifier with a discriminant function that maps each input directly onto a class label, such as a support vector machine. This type also determines its predicted class based on the sign of the discriminant function, $f(\mathbf{x})$. Much as in the case of a ROC method for a generative model, we employ a pair of discriminant functions $f_s(\mathbf{x})$, one for each sensitive value $s \in \{0, 1\}$.

We can now consider an actual independence constraint for classifiers with a discriminant function. To derive this condition, we exploit an actual independence constraint for a discriminative model in Sect. 3.3.2. We here rewrite Eq. (12) by using discriminant function (22):

$$\begin{cases} \hat{\Pr}^{\circ}[\tilde{Y}{=}1|\mathbf{X}{=}\mathbf{x}, S{=}s] = \begin{cases} 1, & \text{if } f_s^{\circ}(\mathbf{x}) \geq 0 \\ 0, & \text{otherwise} \end{cases} \\ \hat{\Pr}^{\circ}[\tilde{Y}{=}0|\mathbf{X}{=}\mathbf{x}, S{=}s] = 1 - \hat{\Pr}^{\circ}[Y{=}1|\mathbf{X}{=}\mathbf{x}, S{=}s] \end{cases}, \tag{23}$$

where $f_s^{\circ}(\mathbf{x})$ is a discriminant function used for predicting a class of objects whose sensitive value is $s$. Now, even for a classifier with a discriminant function, we can compute a fair model distribution, $\hat{\Pr}^{\circ}[\tilde{Y}, S]$, from Eqs. (10) and (23). Note that model-based independence can be defined for classifiers with a discriminative model, but it

cannot be defined for those with a discriminant function, because a joint distribution is not explicitly modeled.

We then modify Algorithm 2 to render it applicable to a classifier with a discriminant function. Two functions, $f_s(\mathbf{x}),\ s \in \{0, 1\}$, are learned, one from each of the datasets, $\mathcal{D}_0$ and $\mathcal{D}_1$, and bias parameters, $b_s,\ s \in \{0, 1\}$, are introduced. We define a pair of fair discriminant functions as

$$f_s^\circ(\mathbf{x}) = f_s(\mathbf{x}) + b_s, \text{ for } s \in \{0, 1\}. \tag{24}$$

Parameters to optimize $\hat{\Pr}^\circ[y|s]''$ in Eq. (21) are replaced with these bias parameters, $b_s$. $\hat{\Pr}^\circ[\tilde{Y}|\mathbf{X}, S]$ is calculated by Eq. (23) and is applied in the step for finding appropriate parameters in line 4. In addition, likelihood is derived based on discriminant functions in line 5. We applied this modified algorithm to logistic regression and a linear SVM, and call these fairness-aware classifiers the ROCLR-AI and ROCSVM-AI methods, respectively.

It should be noted that this framework covers the approach for a classifier with a generative model; that is, we focus on the inequality appearing in Eq. (11):

$$\hat{\Pr}[Y{=}1, \mathbf{x}, s] \geq \hat{\Pr}[Y{=}0, \mathbf{x}, s].$$

We decompose these joint distributions as if an independent generative model is learned for each sensitive value:

$$\hat{\Pr}[Y, \mathbf{X}, S] = \hat{\Pr}[\mathbf{X}|Y, S]\,\hat{\Pr}[Y|S]\,\hat{\Pr}[S]$$

After taking a logarithm of each side of this inequality, the fair discriminant function can be derived by subtracting the right-hand side from the left-hand side:

$$f_s^\circ(\mathbf{x}) = \Big\{\log \hat{\Pr}[Y{=}1|\mathbf{X}{=}\mathbf{x}, S{=}s] - \log \hat{\Pr}[Y{=}0|\mathbf{X}{=}\mathbf{x}, S{=}s]\Big\}$$

$$+\Big\{\log \hat{\Pr}^\circ[Y{=}1|S{=}s] - \log \hat{\Pr}^\circ[Y{=}0|S{=}s]\Big\}.$$

The first and second terms surrounded by curly braces correspond to $f_s(\mathbf{x})$ and $b_s$ in Eq. (24), respectively. This fact indicates that the universal ROC method can change all types of classifiers so as to satisfy an actual independence constraint.

## 6 Experiment

We implemented fairness-aware classifiers satisfying model-based independence and actual independence, and empirically compared these classifiers on real benchmark datasets and a synthetic dataset. This comparison revealed the importance of an actual independence condition, which takes the effects of model bias and a deterministic decision rule into account.

### 6.1 Experimental conditions

Before showing the experimental results, we will describe the experimental conditions. We performed five-fold cross-validation, and calculated the evaluation indices. To evaluate the performance of fairness-aware classifiers, we had to examine how strictly a fairness constraint was satisfied, as well as how accurately class labels were predicted. We used an accuracy measure (Acc), which is the ratio of correctly labeled samples, to evaluate the prediction accuracy. The larger the accuracy is, the more accurately classes are predicted. We supplementally showed Precision and Recall. Precision is the ratio of correctly labeled positive data to the all positively labeled data, and Recall is the ratio of correctly labeled positive data to the all true positive data. We used two metrics for the evaluation of fairness: Calders and Verwer's score (CVS) and normalized mutual information (NMI). CVS is defined by Eq. (2). As CVS approaches zero, fairer decisions are made. Mutual information is a non-negative index to measure the quantity of information shared between random variables. As mutual information between $Y$ and $S$ is linearly decreased, the probability that the value of $Y$ can be inferred given the state of $S$ is exponentially decreased. NMI is defined by normalizing the mutual information into a range [0, 1]:

$$\mathrm{NMI}(Y, S) = \mathrm{I}(Y; S)/\sqrt{\mathrm{H}(Y)\mathrm{H}(S)}, \tag{25}$$

where $\mathrm{I}(\cdot;\cdot)$ and $\mathrm{H}(\cdot)$ denote mutual information and entropy, respectively. This is a geometric mean of $\mathrm{I}(Y; S)/\mathrm{H}(Y)$ and $\mathrm{I}(Y; S)/\mathrm{H}(S)$. Intuitively, while the former is a ratio of information of $S$ misused for prediction, the latter is interpreted as a ratio of information of $S$ leaked by observing predictions. The smaller NMI is, the fairer are the decisions. We round NMI to two significant figures and round the other indexes off to three decimal places.

We examined classifiers as described below.[1] As baselines, we tested standard classifiers trained by using only non-sensitive features. These were three types of classifiers, naive Bayes, logistic regression, and a linear SVM (respectively, NB, LR, and SVM), which were implemented in the scikit-learn (Pedregosa et al. 2011) packages. Note that these classifiers may make potentially unfair decisions. Fairness-aware classifiers were variants of these three classifiers. Variants of naive Bayes classifiers were Calders & Verwer's two-naive-Bayes (CV2NB) in Sect. 3.1, the ROC method satisfying model-based independence (ROCNB-MI) in Sect. 5.2, and that satisfying actual independence (ROCNB-AI) in Sect. 5.3. Regarding logistic regression, we adopted the prejudice remover regularizers satisfying model-based and actual independence conditions (respectively, PR-MI and PR-AI) in Sect. 4, and the universal ROC (ROCLR-AI) in Sect. 5.4. Finally, we tested a universal ROC using the linear SVM (ROCSVM-AI) in Sect. 5.4.

[1] Our implementations of these methods are available at http://www.kamishima.net/faclass/.

**Table 1** Accuracy and fairness indexes for the Adult dataset

| Model-based independence | | | | Actual independence | | | |
|---|---|---|---|---|---|---|---|
| Methods | Acc | NMI | CVS | Methods | Acc | NMI | CVS |
| NB | 0.820 | $1.11 \times 10^{-01}$ | 0.348 | CV2NB | 0.825 | $7.94 \times 10^{-10}$ | 0.000 |
| ROCNB-MI | 0.820 | $2.25 \times 10^{-02}$ | 0.160 | ROCNB-AI | 0.837 | $8.59 \times 10^{-06}$ | 0.002 |
| LR | 0.862 | $4.51 \times 10^{-02}$ | 0.172 | PR-AI | 0.828 | $7.08 \times 10^{-05}$ | 0.008 |
| PR-MI | 0.814 | $2.57 \times 10^{-02}$ | 0.056 | ROCLR-AI | 0.840 | $1.14 \times 10^{-09}$ | −0.000 |
| SVM | 0.862 | $4.32 \times 10^{-02}$ | 0.160 | ROCSVM-AI | 0.839 | $1.09 \times 10^{-07}$ | −0.000 |

**Table 2** Accuracy and fairness indexes for the Dutch dataset

| Model-based independence | | | | Actual independence | | | |
|---|---|---|---|---|---|---|---|
| Methods | Acc | NMI | CVS | Methods | Acc | NMI | CVS |
| NB | 0.787 | $1.83 \times 10^{-02}$ | 0.159 | CV2NB | 0.757 | $8.47 \times 10^{-06}$ | −0.003 |
| ROCNB-MI | 0.808 | $8.82 \times 10^{-02}$ | 0.346 | ROCNB-AI | 0.765 | $1.80 \times 10^{-06}$ | −0.002 |
| LR | 0.819 | $2.19 \times 10^{-02}$ | 0.171 | PR-AI | 0.716 | $4.63 \times 10^{-07}$ | 0.001 |
| PR-MI | 0.790 | $2.28 \times 10^{-02}$ | 0.161 | ROCLR-AI | 0.778 | $5.68 \times 10^{-07}$ | −0.001 |
| SVM | 0.817 | $1.89 \times 10^{-02}$ | 0.158 | ROCSVM-AI | 0.777 | $2.35 \times 10^{-07}$ | −0.001 |

### 6.2 Results on real benchmark datasets

We first tested fairness-aware classifiers on the two benchmark datasets[2] used in Kamiran et al. (2013). The first was an adult dataset (a.k.a., the census income dataset) originally distributed at the UCI repository (Frank and Asuncion 2010). We refer to this dataset as Adult. Its class variable represented whether an individual's income was high or low, and its sensitive feature represented the individual's gender. The size of the dataset was 15, 696, and the number of non-sensitive features was 12. The second dataset was the Dutch census dataset, which we refer to as Dutch. Its class variable represented whether an individual's profession was high income or low income, and its sensitive feature represented the individual's gender. The size of the dataset was 60, 420, and the number of non-sensitive features was 10. Note that all features were categorical and were transformed into multiple binary features by a 1-of-$K$ scheme.

We present our experimental results for the Adult dataset in Table 1 and those for the Dutch dataset in Table 2. For each dataset and each classifier, we computed three evaluation measures: accuracy (Acc), normalized mutual information (NMI), and the Caldars & Verwer score (CVS). We show the results obtained by baseline methods or methods to satisfy model-based independence in the left half of each table, and those obtained by methods to satisfy actual inde-

[2] https://sites.google.com/site/conditionaldiscrimination/.

pendence in the right half of each table. For PR-MI and PR-AI methods, we chose $3\times10^{1}$ and $1\times10^{4}$ as an independence regularization parameter, $\eta$, respectively.

We evaluated the accuracy and fairness of classifiers on these datasets in order to examine the following two questions. First, is the difference between model-based independence and actual independence essential to improve the trade-offs between accuracy and fairness? This validates the importance of the effects of model bias and deterministic decision as analyzed in Sect. 3.2. Second, can the universal ROC methods in Sect. 5.4 improve fairness effectively?

We begin with the first question: is the difference between model-based independence and actual independence essential for the performance in fairness-aware classification? To answer this, we compared the results in the left half of the tables with those in the right half. Comparing the fairness-aware classifiers with their corresponding baseline methods, the relative losses in accuracy by satisfying actual independence were at most about 5% except for the Dutch PR-AI case (12.5%). Moreover, the prediction accuracy was improved in some cases, e.g., the ROCNB-AI for the Adult dataset. In terms of fairness, the improvements were drastic. The NMIs and CVSes of the baselines were worse than $1\times10^{-02}$ and 0.1, respectively. On the other hand, the methods satisfying actual independence achieved better performance than the order of $10^{-04}$ in NMIs and than 0.01 in CVS.

We then compared methods satisfying actual independence with those satisfying model-based independence, which are aligned in the same row in the tables. Specifically, the ROCNB-AI was compared with the ROCNB-MI, and the PR-AI was compared with the PR-MI. The performances in accuracy appeared to be comparable. Each of the PR-AI and the ROCNB-AI methods won in two cases and lost in two cases. Note that the differences were all significant at the level of 1%. In terms of fairness, methods satisfying actual independence again achieved drastic improvements. While the NMIs obtained by the ROCNB-MI and PR-MI methods were worse than $10^{-02}$, those obtained by ROCNB-AI and PR-AI were better than $10^{-04}$. In terms of CVS, the methods satisfying actual independence could achieve scores of nearly zero, but methods satisfying model-based independence could not. From the above results, we can conclude that satisfying a constraint of actual independence, rather than a constraint of model-based independence, improved fairness while minimizing the loss of accuracy.

We now turn to the second question: can the universal ROC methods in Sect. 5.4 improve fairness effectively? As observed in Tables 1, 2, the ROCLR-AI and ROCSVM-AI methods achieved a much higher level of fairness. Because this approach to the universal ROC method can be applied to any type of classifier, users can choose any type of classifiers as bases of fairness-aware classifiers.

We now show the results of the supplemental examination of the effects of an independence parameter $\eta$ of prejudice removers, the PR-MI and the PR-AI, to adjust the balance between accuracy and fairness. Figure 5 shows the change of performance in accuracy and fairness depending on the parameter $\eta$. The increase of $\eta$ generally worsened accuracy and improved fairness as we intended. The PR-MI method failed if $\eta > 10^{2}$ because all the data were classified into one class, while the PR-AI method worked relatively stably even for larger $\eta$. Therefore, we chose $\eta{=}3\times10^{1}$, at the point

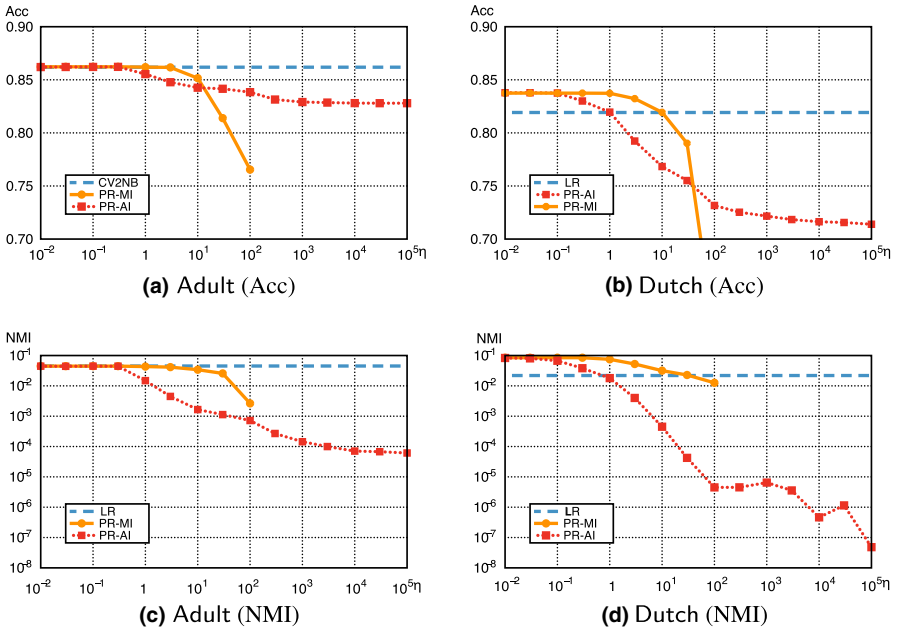


**Fig. 5** The change in accuracy and NMI according to $\eta$ **a** Adult (Acc). **b** Dutch (Acc). **c** Adult (NMI). **d** Dutch (NMI). *Note*: *Horizontal axes* represent the parameter $\eta$, and *vertical axes* represent statistics in each subtitle. *Blue* broken lines, range solid lines with circles, and *red* dotted lines with squares indicate the statistics of LR, PR-MI, and PR-AI, respectively. Larger Acc indicates better performance in accuracy, and smaller MNI indicates better performance in fairness

where just before the accuracy started to fall, for the PR-MI, and chose $\eta$=$10^4$, at which Acc and NMI became saturated, for the PR-AI. Note that NMIs were unstable for large $\eta$ because the non-convexity of a prejudice remover regularizer made it difficult to optimize the objective function.

Finally, we will comment on the effect of changing a class ratio, $\hat{\Pr}^{\circ}[Y]$. As pointed out in Žliobaitė (2015), this ratio affects the realizable degree of fairness. In addition, the ratio cannot be changed, such as in the case that the number of successful candidates is fixed in a university admittance. In the ROC method, the ratio can be controlled, and we set the ratio, $\hat{\Pr}^{\circ}[Y]$, to that observed in a training dataset. If this constraint and an actual independence condition are simultaneously satisfied, the method corresponds to our preliminary method (Kamishima et al. 2013). We denote these ROCNB-AI, ROCLR-AI, and ROCSVM-AI variants by ROCNB-FF, ROCLR-FF, and ROCSVM-FF, respectively. Tables 3, 4 showed accuracy indexes, Acc, Precision, and Recall for the Adult and Dutch datasets, respectively. The estimated positive ratio (EPR) is the ratio of positively estimated data to the whole dataset. Note that the ratios of positive data in the training dataset were 0.235 for the Adult and 0.476 for the Dutch. The EPRs could diverge from these ratios, if they were not constrained. In particular, the PR-MI method largely diverged. Additionally, in cases in which the EPRs were constrained, Precision and Recall tended to have similar values. However, when the EPRs were not

**Table 3** Additional accuracy indexes and estimated positive ratios for the Adult dataset

| Model-based independence | | | | | Actual independence | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Methods | EPR | Acc | Precision | Reccall | Methods | EPR | Acc | Precision | Recall |
| NB | 0.323 | 0.820 | 0.584 | 0.803 | CV2NB | 0.234 | 0.825 | 0.628 | 0.627 |
| ROCNB-MI | 0.304 | 0.820 | 0.591 | 0.763 | ROCNB-AI | 0.162 | 0.837 | 0.722 | 0.497 |
| | | | | | ROCNB-FF | 0.235 | 0.825 | 0.628 | 0.629 |
| LR | 0.185 | 0.862 | 0.762 | 0.599 | PR-AI | 0.241 | 0.828 | 0.631 | 0.646 |
| PR-MI | 0.050 | 0.814 | 0.986 | 0.211 | ROCLR-AI | 0.169 | 0.840 | 0.722 | 0.520 |
| | | | | | ROCLR-FF | 0.235 | 0.833 | 0.44 | 0.645 |
| SVM | 0.170 | 0.862 | 0.786 | 0.568 | ROCSVM-AI | 0.174 | 0.839 | 0.711 | 0.528 |
| | | | | | ROCSVM-FF | 0.236 | 0.832 | 0.641 | 0.644 |

**Table 4** Additional accuracy indexes and estimated positive ratios for the Dutch dataset

| Model-based independence | | | | | Actual independence | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Methods | EPR | Acc | Precision | Reccall | Methods | EPR | Acc | Precision | Recall |
| NB | 0.496 | 0.787 | 0.765 | 0.797 | CV2NB | 0.474 | 0.757 | 0.746 | 0.742 |
| ROCNB-MI | 0.493 | 0.808 | 0.788 | 0.815 | ROCNB-AI | 0.401 | 0.765 | 0.801 | 0.674 |
| | | | | | ROCNB-FF | 0.477 | 0.758 | 0.745 | 0.746 |
| LR | 0.422 | 0.819 | 0.850 | 0.753 | PR-AI | 0.635 | 0.716 | 0.652 | 0.869 |
| PR-MI | 0.320 | 0.790 | 0.916 | 0.616 | ROCLR-AI | 0.433 | 0.778 | 0.793 | 0.721 |
| | | | | | ROCLR-FF | 0.476 | 0.774 | 0.763 | 0.763 |
| SVM | 0.404 | 0.817 | 0.863 | 0.733 | ROCSVM-AI | 0.436 | 0.777 | 0.790 | 0.724 |
| | | | | | ROCSVM-FF | 0.477 | 0.774 | 0.762 | 0.763 |

constrained, Precision and Recall could deviate; this was especially true in the PR-MI method. Regarding fairness, NMI were $4.50\times10^{-08}$ (Adult) and $2.43\times10^{-12}$ (Dutch) for the ROCNB-FF method. Compared with the ROCNB-MI method, this method showed better fairness. A overall trend in the comparison of the ROCNB-FF method with ROCNB-AI method; the former was better in fairness, but worse in accuracy. This is because the EPR was changed to optimize accuracy in ROCNB-AI.

We can summarize the above experimental results as follows:

- Fairness could be drastically improved with less sacrifice in accuracy by satisfying actual independence instead of model-based independence. This implies the importance of the effects of model bias and a deterministic decision rule in terms of fairness.
- The universal ROC method worked as well as the other fairness-aware classifiers, and any type of classifier could be modified to a fairness-aware classifier.

**Table 5** Accuracy and fairness indexes for a synthetic dataset

| Model-based independence | | | | Actual independence | | | |
|---|---|---|---|---|---|---|---|
| Methods | FAcc | UAcc | CVS | Methods | FAcc | UAcc | CVS |
| NB | 0.891 | 0.965 | 0.183 | CV2NB | 0.932 | 0.908 | 0.014 |
| ROCNB-MI | 0.909 | 0.945 | 0.126 | ROCNB-AI | 0.925 | 0.911 | 0.013 |
| LR | 0.894 | 0.984 | 0.189 | PR-AI | 0.906 | 0.924 | 0.020 |
| PR-MI | 0.920 | 0.928 | 0.033 | ROCLR-AI | 0.922 | 0.933 | 0.012 |
| SVM | 0.893 | 0.980 | 0.189 | ROCSVM-AI | 0.922 | 0.929 | 0.012 |

### 6.3 Results for a synthetic dataset

We here investigate whether class labels generated by distributions on a fair-subspace can be estimated by fairness-aware classifiers. In the previous section, we examined accuracy to evaluate how correctly unfair labels were predicted. However, we really want to evaluate how correctly fair labels were predicted. Because such fair labels cannot be observed in real datasets, we will use a synthetic dataset to test accuracy for the fair labels.

We generated a synthetic dataset so that it satisfied fairness-constraints. We generated $n$ non-sensitive feature vectors, $\mathbf{x}_i$, $i = 1, \ldots, n$. Each vector consisted of 20 binary features, which were uniformly-randomly generated. Vectors $\{\mathbf{x}_i\}$ were divided into 18 and 2 features, which were denoted by $\{\mathbf{x}_i^{(L)}\}$ and $\{\mathbf{x}_i^{(S)}\}$, respectively. We generated 20 weights, $\mathbf{w}$, whose elements followed a distribution, Normal(0, 1), and the weight vector was again divided into $\mathbf{w}^{(L)}$ and $\mathbf{w}^{(S)}$. Scores for fair classes were calculated by $f_i^{(L)} = {\mathbf{w}^{(L)}}^\top \mathbf{x}_i^{(L)} + \epsilon$, where $\epsilon \sim$ Normal(0, 0.1) was independent Gaussian noise. We assigned 0 fair labels for the bottom $n\,\Pr^\circ[L{=}0]$ data in the scores, and 1 fair labels for the rest. Scores for sensitive features were calculated by $f_i^{(S)} = {\mathbf{w}^{(S)}}^\top \mathbf{x}_i^{(S)} + \epsilon$, and sensitive features were generated in a similar way. A fair label, $L$, and a sensitive feature, $S$, were unconditionally independent because they did not depend on common non-sensitive features; thus, a fairness constraint, $L \perp\!\!\!\perp S$, was satisfied. Scores for unfair labels were calculated by $f_i^{(Y)} = {\mathbf{w}^{(L)}}^\top \mathbf{x}_i^{(L)} + {\mathbf{w}^{(S)}}^\top \mathbf{x}_i^{(S)} + \epsilon$, and unfair labels, $Y$, were generated in a similar way. Here, because both unfair labels and sensitive features depend on $\mathbf{x}_i^{(S)}$, unfair labels and sensitive features were conditionally independent, but not unconditionally independent. Finally, we show the parameters: $\Pr^\circ[L{=}0] = 0.5$, $\Pr^\circ[S{=}0] = 0.3$, $\Pr^\circ[Y{=}0] = 0.5$, $n{=}10000$.

We tested the same set of classifiers tested in the previous section on synthetic datasets generated by the procedure described above. 100 pairs of datasets were generated: one of each pair was used for training, and the other was used for testing. Note that only unfair labels were used in training. Table 5 shows the mean accuracies over 100 datasets for both fair and unfair labels, denoted by FAcc and UAcc, respectively. Means of absolutes of the fairness indexes between predicted labels and sensitive values, CVS, are also shown. Note that we did not show means of NMI because they are meaningless due to their large variance over 100 datasets.

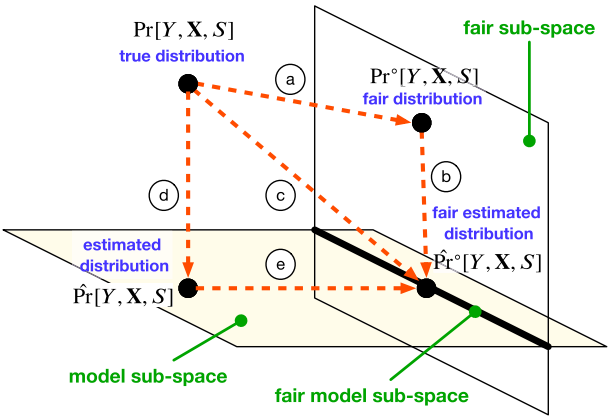


**Fig. 6** A geometrical representation of approaches to fairness-aware classification

We first focus on FAcc and UAcc. All the standard classifiers could successfully predict unfair labels, but performed poorly in predicting fair labels. Inversely, all fairness-aware classifiers could improve the accuracy on fair labels, but worsened the accuracy on unfair labels, compared to their corresponding standard classifiers, e.g., NB for CV2NB. Further, in terms of CV2NB and ROCNB-AI, the accuracies on fair labels were better than those on unfair labels. These results were what we intended, because standard classifiers and fairness-aware classifiers were designed to predict unfair and fair labels, respectively. We next discuss the fairness index, CVS. All fairness-aware classifiers could make fairer decisions than their corresponding standard classifiers, as we intended. In addition, classifiers satisfying actual independence exhibited greater fairness than those satisfying model-based independence. CVS for ROCNB-AI was smaller than that for ROCNB-AI, and PR-AI classified more fairly than PR-MI. This proved the advantage of achieving actual independence.

We can summarize the above experimental results as follows:

- Fairness-aware classifiers performed better than their corresponding standard classifiers in terms of accuracy on fair labels and in fairness indexes.
- Classifiers satisfying actual independence could make fairer decisions than those satisfying model-based independence.

## 7 Related work

This section reviews fairness-aware classifiers. Figure 6 geometrically represents approaches to fairness-aware classification as in Fig. 1. Approaches to fairness-aware classification can be classified into three types (Ruggieri et al. 2010): pre-process, in-process, and post-process. In the pre-process approach, potentially unfair data are mapped onto the fair sub-space (ⓐ in Fig. 6), and the fair model is learned by a standard classifier (ⓑ). Any classifier can in principle be used in this approach, but the development of a mapping method might be difficult without making any assumption on a classifier. In particular, we consider that actual independence will not be satisfied

without specifying a classifier. Massaging is a technique to relabel a dataset based on the predicted probability of class labels (Kamiran and Calders 2012). Hajian and Domingo-Ferrer (2013) changed labels or sensitive features by exploiting frequent pattern mining. Zemel et al. (2013) tried to obtain an intermediate representation that fulfilled three constraints: statistical parity, minimizing the distortion, and maximizing the classification accuracy. Feldman et al. (2015) proposed a method to transform non-sensitive features so that a sensitive feature cannot be predicted from the transformed non-sensitive features.

In the in-process approach, a fair model is learned directly from a potentially unfair dataset as in ⓒ in Fig. 6. This approach can potentially achieve better trade-offs than the other approaches because classifiers are less restricted in their design. However, it is technically difficult to formalize or optimize an objective function. In addition, for each distinct type of classifier, its fair variant must be developed. The prejudice remover in Sect. 4 is categorized into this approach. Kamiran et al. (2010) developed algorithms to learn decision trees for a fairness-aware classification task, in which the labels at leaf nodes were changed so as to decrease the CVS. Fukuchi et al. introduced two constraint terms, *η-neutrality* (Fukuchi et al. 2013) and *neutrality risk* (Fukuchi and Sakuma 2014). Zafar et al. (2015) developed SVMs and logistic regression with constraint terms that make classes uncorrelated (instead of independent) with a sensitive feature. They also proposed a classifier to satisfy a fairness condition that misclassification rates for groups sharing the same sensitive values were equal (Zafar et al. 2017).

In the post-process approach, a standard classifier is first learned (ⓓ), and then the learned classifier is modified to satisfy a fairness constraint (ⓔ). This approach adopts the rather restrictive assumption, *obliviousness* (Hardt et al. 2016), that fair class labels are determined based only on labels of a standard classifier and a sensitive value, and are independent from non-sensitive features. However, this obliviousness assumption makes the development of a fairness-aware classifier easier. Calders & Verwer's two-naive-Bayes method in Sect. 3.1 and the ROC method in Sect. 5.1 are categorized into this approach. Kamiran et al. discussed the re-labeling technique for fairer decisions while considering the effects of confounding variables (Kamiran et al. 2013). Hardt et al. (2016) developed a post-process-style method to match misclassification rates between groups.

Finally, we will review other aspects of fairness-aware classification. Fairness-aware data mining is an emerging research topic and involves many controversial problems. Hajian et al. provide a good tutorial on the relevant literature (Hajian et al. 2016). When using a fairness-aware classifier, a sensitive feature may not be provided for various reasons, such as the protection of privacy. To alleviate this problem, Fukuchi et al. (2013) proposed to use a predictor for a sensitive feature, learned from an independent dataset. In Sweeney (2013), to investigate the fairness online of ad delivery, a sensitive feature, race, is predicted from an independent public dataset, the birth records of the state of California. Even if both a class and a sensitive feature depend on a common factor, the use of the factor in classification is legal for various reasons, such as a genuine occupational requirement. In the context of fairness-aware data mining, such a factor is referred to as an explainable variable (Kamiran et al. 2013). Given such an explainable variable, $\mathbf{E}$, a fair constraint can be relaxed from unconditional independence, $Y \perp\!\!\!\perp S$, to conditional independence, $Y \perp\!\!\!\perp S \mid \mathbf{E}$.

Because an explainable variable can be treated as a confounding variable in a causal inference context, a propensity score is used to maintain the effect of a explainable variable (Calders et al. 2013).

## 8 Conclusions

In this paper, we discussed an independence condition in terms of a fairness-aware classifier. We proposed notions of model-based and actual independence, in which the treatments of model bias and a decision rule are different. We then developed two types of pairs of classifiers, one of which achieves model-based independence and the other actual independence. Empirical comparison of these pairs of classifiers validated that the distinction of two types of independence is essential for improving trade-offs between fairness and accuracy. Finally, We extended an approach exploited in the ROC method to make it applicable to any type of classifiers.

Though we can now achieve a higher level of fairness by satisfying an actual independence condition, the time complexity of algorithms must be improved in the future. Due to the discrete property of a deterministic decision rule, the objective function to optimize becomes indifferentiable, and this fact makes it difficult to find optimal parameters. Approximation and relaxation techniques would be helpful for alleviating this problem.

**Acknowledgements** We wish to thank Dr. Sicco Verwer for providing detailed information about his work, Dr. Žliobaitė for providing datasets, and anonymous reviewers for their helpful suggetions to improve the clarity of this paper. This work is supported by MEXT/JSPS KAKENHI Grant Numbers JP24500194, JP15K00327, and JP16H02864.



## References

Berendt B, Preibusch S (2012) Exploring discrimination: A user-centric evaluation of discrimination-aware data mining. In: Proceedings of the IEEE Int'l Workshop on Discrimination and Privacy-Aware Data Mining, pp 344–351

Bishop CM (2006) Pattern Recognition and Machine Learning. Information Science and Statistics. Springer, New York

Calders T, Verwer S (2010) Three naive Bayes approaches for discrimination-free classification. Data Min Knowl Discov 21:277–292

Calders T, Karim A, Kamiran F, Ali W, Zhang X (2013) Controlling attribute effect in linear regression. In: Proceedings of the 13th IEEE Int'l Conference on Data Mining, pp 71–80

Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R (2012) Fairness through awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, pp 214–226

Elkan C (2001) The foundations of cost-sensitive learning. In: Proceedings of the 17th Int'l Joint Conference on Artificial Intelligence, pp 973–978

Feldman M, Friedler SA, Moeller J, Scheidegger C, Venkatasubramanian S (2015) Certifying and removing disparate impact. In: Proceedings of the 21st ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining, pp 259–268

Frank A, Asuncion A (2010) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, http://archive.ics.uci.edu/ml

Fukuchi K, Sakuma J (2014) Neutralized empirical risk minimization with generalization neutrality bound. In: Proceedings of the ECML PKDD 2014, Part I, pp 418–433 [LNCS 8724]

Fukuchi K, Sakuma J, Kamishima T (2013) Prediction with model-based neutrality. In: Proceedings of the ECML PKDD 2013, Part II, pp 499–514 [LNCS 8189]

Hajian S, Domingo-Ferrer J (2013) A methodology for direct and indirect discrimination prevention in data mining. IEEE Trans Knowl Data Eng 25(7):1445–1459

Hajian S, Bonchi F, Castillo C (2016) Algorithmic bias: from discrimination discovery to fairness-aware data mining. The 22nd ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining, Tutorial

Hardt M, Price E, Srebro N (2016) Equality of opportunity in supervised learning. In: Advances in Neural Information Processing Systems 29

Kamiran F, Calders T (2012) Data preprocessing techniques for classification without discrimination. Knowl Inf Syst 33:1–33

Kamiran F, Calders T, Pechenizkiy M (2010) Discrimination aware decision tree learning. In: Proceedings of the 10th IEEE Int'l Conferene on Data Mining, pp 869–874

Kamiran F, Karim A, Zhang X (2012) Decision theory for discrimination-aware classification. In: Proceedings of the 12th IEEE Int'l Conference on Data Mining, pp 924–929

Kamiran F, Žliobaitė I, Calders T (2013) Quantifying explainable discrimination and removing illegal discrimination in automated decision making. Knowl Inf Syst 35:613–644

Kamishima T, Akaho S, Asoh H, Sakuma J (2012) Fairness-aware classifier with prejudice remover regularizer. In: Proceedings of the ECML PKDD 2012, Part II, pp 35–50 [LNCS 7524]

Kamishima T, Akaho S, Asoh H, Sakuma J (2013) The independence of the fairness-aware classifiers. In: Proceedings of the IEEE 13th Int'l Conference on Data Mining Workshops, pp 849–858

Pedregosa F, et al (2011) Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12:2825–2830, http://scikit-learn.org

Pedreschi D, Ruggieri S, Turini F (2008) Discrimination-aware data mining. In: Proceedings of the 14th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining, pp 560–568

Ruggieri S, Pedreschi D, Turini F (2010) Data mining for discrimination discovery. ACM Transactions on Knowledge Discovery from Data 4(2):Article 9

Sweeney L (2013) Discrimination in online ad delivery. Commun ACM 56(5):44–54

Zafar MB, Martinez IV, Rodriguez MG, Gummadi K (2015) Fairness constraints: A mechanism for fair classification. In: ICML2015 Workshop: Fairness, Accountability, and Transparency in Machine Learning

Zafar MB, Valera I, Rogriguez MG, Gummadi KP (2017) Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In: Proceedings of the 26th Int'l Conference on World Wide Web, pp 1171–1180

Zemel R, Wu Y, Swersky K, Pitassi T, Dwork C (2013) Learning fair representations. In: Proceedings of the 30th Int'l Conference on Machine Learning, pp 325–333

Žliobaitė I (2015) On the relation between accuracy and fairness in binary classification. In: ICML2015 Workshop: Fairness, Accountability, and Transparency in Machine Learning