17. Removing Discrimination from Probabilistic Classification

Sicco Verwer

Research and Documentation Centre, Ministry of Security and Justice, the Netherlands

Radboud University Nijmegen, the Netherlands

Toon Calders

Eindhoven University of Technology, the Netherlands

Abstract In this chapter we give three solutions for the discrimination-aware classification problem that are based upon Bayesian classifiers. These classifiers model the complete probability distribution by making strong independence assumptions. First we discuss the necessity of having discrimination-free classification for probabilistic models. Then we will show three ways to adapt a Naive Bayes classifier in order to make it discrimination-free. The first technique is based upon setting different thresholds for the different communities. The second technique will learn two different models for both communities, while the third model describes how we can incorporate our belief of how discrimination was added to the decisions in the training data as a latent variable. By explicitly modeling the discrimination, we can reverse engineer decisions. Since all three models can be seen as ways to introduce positive discrimination, we end the chapter with a reflection on positive discrimination.

17.1 Introduction

The topic of discrimination-aware data mining was first introduced in (Calders et al., 2009; Kamiran & Calders, 2009; Pedreschi et al., 2008), and is motivated by the observation that often training data contains unwanted dependencies between the attributes. Given a labeled dataset and a sensitive attribute; e.g., gender, the goal of our research is to learn a classifier for predicting the class label that does not discriminate with respect to a given sensitive attribute, e.g., for every sex, the probability of being in the positive class should be roughly the same. For a more detailed description of the problem domain and some algorithmic solutions, see Chapters 3 and 12 of this book. This chapter will discuss different techniques of learning and adapting probabilistic classifiers to make them discrimination-free.

Initially, we concentrate on Naive Bayes classifiers, see, e.g., (Bishop, 2006). These are simple probabilistic models with strong independence assumptions. The main benefit of these assumptions is that they make the problem of learning a Naive Bayes classifier easy. Intuitively, a Naive Bayes classifier can be used to compute the probability that a given combination of attribute values (features or characteristics) obtains a positive class value. If this value is larger than a given threshold (typically 50%), the classifier outputs "yes", otherwise it outputs "no". Consider, e.g., the following example of a spam filter.

Example
Suppose that we have a collection of emails, each of which is marked either as a spam mail or a regular mail. In order to learn a predictive model for spam, we have to transform every message into a vector of values. This is typically done by selecting all words that appear in the emails, order them, and transform every email in a sequence of 0-1 where a 1 in the ith position indicates that the ith word appeared in the mail. Otherwise, the ith position is 0. E.g., suppose that the ordered list of words appearing in the collection of emails is:

(of, a, the, man, $, win, price, task).

Then the vector (1,1,1,0,0,1,0,0) would indicate an email in which the words "of", "a", "the", and "win" appear, but not "man", "$", or "price". Based on the data vectors obtained, a model will be learned that can be used to predict for a new, unlabeled email if it is spam or not. For the Naive Bayes classifier, the model essentially corresponds to assigning a positive or negative score to every word, and setting a threshold. The scores for all words present in the email to be classified are added up, and only if the total score exceeds the threshold, the mail will be classified as spam. The scores of the words and the threshold are the parameters of the model. The Naive Bayes classification algorithm learns optimal values for these parameters based upon the data. For example, suppose that the Naive Bayes algorithm learns the following scores: (-0.5,-0.5,-0.5,0.5,2,1.5,2,-3) and threshold 2, then an email with content "win a price" corresponds to the vector (0,1,0,0,0,1,1,0) and gets a score of -0.5+1.5+2 = 3, which exceeds the threshold. Therefore, the mail is classified as spam.

A more exact definition of Bayesian models will be given in Section 2 of this chapter. The decision of a Naive Bayes classifier is based on a given data set, which is used to fit the parameters of the Naive Bayes model, and the strong class-independence assumption. Although this assumption is often violated in practice, even then good results can be obtained using a Naive Bayes approach (Langley et al., 1992).

Example

In our spam-example above, the class-independence assumption says that the occurrence of the different words in the email are independent of each other, given the class. More specifically, in the spam emails, every word has a probability that it occurs, but all words occur independently; if "a" occurs with 20% probability, and "the" with 50% probability in spam mails, the probability that both words occur in the spam email is 20% times 50% = 10%; the only factor that influences the probability of words occurring is if it is a spam email or not. Obviously this assumption will be violated in real emails. Nevertheless, many spam filters successfully use Naive Bayes classifiers even though they are based upon an unrealistic assumption.

As discussed in much detail in Chapter 3, often there is a need to learn classifiers that do not discriminate with respect to certain sensitive attributes, e.g., gender, even though the labels in the training data themselves may represent a discriminatory situation. In Chapters 12 and 13, preprocessing techniques and an adapted decision tree learner for discrimination-aware classification have already been introduced. In this chapter, we provide three methods to make a Naive Bayes model discrimination-free:

1. Use different decision thresholds for every sensitive attribute value; e.g., females need a lower score than man to get the positive label.
2. Learn a different model for every sensitive attribute value and use different decision thresholds.
3. Add an attribute for the actual non-discriminatory class to a specialized Naive Bayes model and try to learn the actual class values of every row in the data-set using the expectation-maximization algorithm.

Note, however, that all of the above methods can be seen as a type of positive discrimination: they assume an equal treatment of every sensitive attribute value and force the predictor to satisfy this assumption, sacrificing predictive accuracy in the process. Thus, although the off-the-shelf classifier considers it more likely for some people to be assigned a positive class, they are forcibly assigned a negative class in order to reduce discrimination, i.e., they are discriminated positively. Since positive discrimination is considered illegal in several countries, these methods should be applied with care. Applying predictive tools untouched, however, should also be done with care since they are very likely to be discriminating: they make use of any correlation in order to improve accuracy, also the correlation between sensitive and class attributes.

Since it is impossible to identify the true cause of being assigned a positive class using data mining, discrimination in data mining cannot be avoided without introducing positive discrimination. When applying data mining, one thus has to make a choice between positive and negative discrimination. In our opinion, using the assumption of equal treatment in a well-thought-out way is a lesser evil than blindly applying a possibly discriminating data mining procedure.

This chapter is organized as follows. We start with an introduction to the Naive Bayes classifier in Section 2. We then use examples to provide arguments in favor of discrimination-aware data-mining in Section 3. Afterwards, we discuss our

discrimination-aware techniques applied to the Naive Bayes classifier in Section 4. In Section 5, we discuss the effects of our techniques on positive discrimination. Section 6 concludes the chapter.

17.2 The Naive Bayes classifier

We already gave an intuitive introduction to the Naive Bayes classifier in the introduction. In this section we will provide a more in-depth discussion of this classifier introducing the necessary background for understanding the proposed adaptations to the model to make it discrimination-free. The Naive Bayes classifier is a simple probabilistic model that assumes independence between all attributes when given the class attribute, see, e.g., (Bishop, 2006). For example, when predicting whether someone has a high or low income (class attribute), the age of a person correlates with the type of position (s)he occupies. A Naive Bayes classifier assumes that once the income is known, these two attributes are independent. For instance, age no longer correlates with position when considering only people with a high (low) income. Formally, a Naive Bayes model computes the following probability function[1]:

$$P(C,A_1,A_2,\ldots,A_n) \propto P(C)P(A_1|C)P(A_2|C)\ldots P(A_n|C)$$

In this formula, C is the class attribute and $A_1,A_2,\ldots,A_n$ are all other attributes. P(C) is a probability function for the different class values, and P(A|C) is a probability function for A's attribute values given the class value. Due to the independence assumption, the total probability function (or model) $P(C,A_1,A_2,\ldots,A_n)$ can be computed simply by multiplying the individual probabilities of the class and of each attribute given the class. We now show using an example how to estimate these probability functions and use them as a classifier.

Example.
Suppose we are given a data-set consisting of 100 people, 40 of which are female and 60 male. We would like to predict whether a new person is likely to have a high or a low income based on this data. In the data-set 20 males and 10 females have a high income. This results in the following probability functions:

P(high income) = 30/100 = 0.3, P(low income) = 0.7
P(male| high income) = 20/30 = 0.67, P(female|high income) = 10/30 = 0.33
P(male| low income) = 40/70 = 0.57, P(female|low income) = 30/70 = 0.43

1 We disregard normalizing constants. Note that this formulation is consistent with the one used in the introduction, as we can easily move from comparing products to sums via the logarithm.

In addition, suppose we also know the education of these people and that this attribute results in the following probability functions:

P(university|high) = 0.5, P(high school|high) = 0.33, P(none|high) = 0.17
P(university|low) = 0.07, P(high school|low) = 0.57, P(none|low.) = 0.36

These functions can all be easily estimated from the data-set by counting how many times each attribute value occurs together with each class attribute value. When we want to determine for instance the probability that a female with high school education receives a high income, we use the total probability function to compute and normalize the probability of these values together with a high and a low income:

P(high income,female,high school) = 0.3•0.33•0.33 = 0.033
P(low income,female,high school) = 0.7•0.43•0.57 = 0.172
P(high income|female,high school) =
P(high income,female,high school)/P(female,high school) =
0.033/(0.033+0.172) = 0.16

Since this is less than 0.5, we estimate that a female with a high school education will not receive a high income. Note that this is estimated based on the assumption that education and gender are independent given the income class.

The above example describes the basic version of a Naive Bayes classifier. Most implementations use Gaussian distributions for continuous attributes and smoothing methods to avoid zero probabilities (Bishop, 2006). In addition, the decision threshold (0.5 in the example) can often be modified. Although using a threshold of 0.5 makes sense intuitively, it is common practice to modify it depending on the situational needs, for instance to increase accuracy, or decrease the number of false positives (Lachiche & Flach, 2003).

17.3 The problem of discrimination in data-mining

In Chapter 3, it is explained how discrimination may occur, even if the training data is non-discriminatory. In this section we will now show specifically for a Naive Bayes classifier how using an off-the-shelf Naive Bayes classifier can lead to discriminatory results.
We motivate our methods using examples of the discriminatory results that are obtained when using a Naive Bayes classifier[2] on the census income data-set[3].

2 We use the Naïve Bayes classifier from the e1071 package in the R statistical toolbox (Dimitriadou et al., 2008).

3 http://archive.ics.uci.edu/ml/datasets/Census+Income

From this data set we try to learn a Naive Bayes classifier that can be used to decide whether a new individual should be classified as having a high or a low income. Historically, this decision has been biased towards the male sex, as can be seen in the following table:

	Low income	High income
Female	9592	1179
Male	15128	6662

Table . *The contingency table of the income and gender attributes.*

This table shows the number of male and female individuals in the high and low income class. About 30% of all male individuals and only about 11% of all female individuals have a high income. Thus, according to the definitions introduced in Chapter 12, the amount of discrimination in the data-set is 30% - 11% = 19%, or 0.19.

Suppose that a bank wants to use such historical information to learn models for predicting the probability that new loan applicants will default their loan. Clearly, the data shows this probability to be dependent on the gender of a person. Nevertheless, from an ethical and legal point of view it is unacceptable to use the gender of a person to deny the loan to him or her, as this would constitute an infringement of the discrimination laws. We now show that this is a serious problem when applying data mining to this type of data.

The problem

If one learns a Naive Bayes classifier from the census income data, the discrimination in the data will be learned as a rule. This can be seen very clearly in the probability tables of the Naive Bayes classifier:

	Low income	High income
Female	0.388	0.150
Male	0.612	0.850

Table . *The P(gender|income) table used in a Naive Bayes classifier.*

The probabilities in this table denote the probability of being male or female, given the income class of an individual. Thus, if a given person has a high income, the probability that that person is male is 0.85. If the person has a low income, this probability is only 0.61. Since the classifier uses this table in its decision whether someone is more likely to have a high or a low income, the discrimination in its predictions is likely to be worse than 0.19. We test this using the test-set (containing unseen data) included in the census income data folder. The amount of discrimination in the class values of this test-set is approximately equal to the amount of discrimination in the data-set:

	Low income	High income
Female	4831	590

Male	7604	3256

Table . *The gender-income contingency table of the test-set.*

This changes however when we use the predictions of the Naive Bayes classifier to determine whether someone has a high or a low income:

	Low income	High income
Female	5094	327
Male	8731	2129

Table . *The gender-predicted income contingency table for the test-set, assigned by a Naive Bayes classifier.*

The amount of discrimination in these predictions is (2129 / (8731+2129)) – (327 / (5094+327)) = 0.20 – 0.06 = 0.14. Thus, surprisingly, the total amount of discrimination has become less. However, notice also that the total positive class probability has dropped from 0.24 to 0.15; I.e., less people get assigned the class label “High income”. This drop artificially lowers the discrimination score. We correct for this drop by lowering the decision threshold of the Naive Bayes classifier until the positive class probability reaches 0.24. This results in the following table:

	Low income	High income
Female	4958	463
Male	7416	3444

Table . *The gender-predicted income contingency table for the test-set, corrected to maintain positive class (high income) probability.*

The positive class probability for females is 0.09, while the positive class probability for males is 0.32, resulting in a total discrimination of 0.32 – 0.09 = 0.23. This is a lot worse than the amount of discrimination in the actual labels of the test-set. One may wonder why this is such a big problem, since the data already told us that females are less likely to have high incomes. Suppose that such a discriminating classifier is used in a decision support system for deciding whether to give a loan to a new applicant. Let us take a look at a part of the decisions made by such a system:

	Low income	High income
Female	319	271
Male	1051	2205

Table . *The corrected gender-predicted income contingency for high income test cases.*

This table shows the labels assigned by the classifier to people in the test-set that actually have a high income. The ones that get assigned a low income in the table are the false negatives. In the banking example, these are the ones that are falsely

denied a loan by the classifier. These false negatives are very important for a decision support system because denying a loan to someone that should actually obtain one can lead to law suits. In fact, when looking at the data, it is obvious that the classifier discriminates females since males have a probability of only 1051 / (1051+2205) = 0.32 to be wrongfully denied a loan, while females have a probability of 319 / (319+271) = 0.54. Using data mining tools unmodified for such decision support systems can thus be considered to be a very dangerous practice.

Removing sensitive information does not help

A commonly used method to avoid potential law suits is to not store any sensitive information such as gender. The idea is that learning a classifier on data without this type of information avoids that the classifier's predictions will be based on the sensitive attribute. This approach, however, does not work. The reason for that is that there may be other attributes that are highly correlated with the sensitive attribute. In such a situation, the classifier will use these correlated attributes and thus discriminate indirectly. This phenomenon was termed the red-lining effect in Chapter 3. In the banking example, e.g., job occupation is correlated with gender. Removing gender will only help a bit, as job occupation can be used as a predictor for this attribute. For example, when we learn a Naive Bayes classifier on the census income data-set without gender information[4] and test it on the test-set with modified threshold, we obtain the following table:

	Low income	High income
Female	4900	521
Male	7474	3386

Table . *The gender-predicted income contingency table for the test-set, assigned by a Naive Bayes classifier learned without gender information.*

This table shows positive class probabilities of 0.10 and 0.31 for respectively females and males, and thus a discrimination of 0.21. This does not improve a lot over the classifier that used the gender information directly. In fact, the false negatives show the same problem as before:

	Low income	High income
Female	301	289
Male	1079	2177

Table . *The no gender information corrected gender-predicted income contingency table for high income test cases.*

Thus, even learning a classifier on a data-set without sensitive information can be dangerous. Removing the sensitive information from a data-set actually makes the situation worse because data-mining tools will still discriminate, but in a much

4 In addition, we replaced “wife” by “husband” in the relationship attribute.

more concealed way, and rectifying this situation using discrimination-aware techniques is extremely difficult without sensitive information.
Obviously, one could also decide to remove all of the attributes that correlate with the sensitive ones from the dataset. Although this would resolve the discrimination problem, in this process a lot of useful information will get lost. In fact, the occupation of a person is a very important decision variable when deciding whether to give a loan or not. The occupation attribute can hence, at the same time, reveal information on gender and give useful, non-discriminatory information on loan defaulting. We provide solutions that make use of all the available information, but in a non-discriminatory way.

17.4 Discrimination-free Naive Bayes classifiers

In this section, we provide three approaches for removing discrimination from a Naive Bayes classifier.

17.4.1 Using different decision thresholds

The most straightforward method for removing discrimination is to modify the decision thresholds differently for the different sensitive values. For instance, we can decide to give a high income label to females if the high income probability is greater than 0.1, but to males if it is greater than 0.6. This instantly reduces discrimination by favoring females. Note that this is a very direct form of positive discrimination since even though the model considers some males more likely to belong to the positive class than some females; it still predicts a negative class for these males and a positive class for the females.
When using different decision thresholds for different sensitive values, an important question to ask is which ones to use, and why. The answer to this question highly depends on the situation. It is well-known that using a different decision threshold influences the number of positives, false positives, negatives, and false negatives. Since the importance of these values differs per application, several analysis techniques like ROC (receiver operator curve) analysis (Lachiche & Flach, 2003) exist to aid in setting this threshold smartly. By using different decision thresholds for different sensitive attribute values, the threshold settings in addition influence the amount of positive and negative discrimination. Ideally, these should be taken into account when performing such an analysis.
In our work, we assume that the amount of people that are assigned a positive class should remain the same. In many applications, keeping this number close to the number or positive labels in the data-set is highly favorable. For instance, in the setting of banks assigning loans to individuals, the bank does not suddenly

want to assign less or more loans. In addition, as explained in Section 3, this assumption makes comparing the different techniques on their discrimination score a lot more fair. We set the decision thresholds using a simple algorithm:

1. Calculate the number of positive class labels P assigned to the data-set.
2. Learn a Naive Bayes classifier on the data-set.
3. Set the decision threshold T_+ and T_- for the favored and discriminated sensitive values to 0.5.
4. Calculate the amount of discrimination in the data-set when using T_+ and T_-.
5. While the discrimination is greater than 0
   6. Calculate the number of positive class labels P' assigned to the data-set.
   7. If P' is greater than P, raise T_+ by 0.01.
   8. If P' is less than or equal to P, lower T_- by 0.01.
   9. Iterate
10. Use the resulting decision thresholds to classify the test-set.

The idea of this algorithm is to lower the threshold for females if the classifier assigns less positive class labels than the number of positive class labels in the data-set. Otherwise, we raise the decision threshold for males. In this way, we try to keep the number of positive class labels intact. One may note that since we want to keep this number intact, it is possible to pre-compute the number of males and females that should get a different class label in order to obtain a discrimination score of 0:

$$m_{change} = m_{assigned} - P(\text{positive class}) \bullet m_{total}$$
$$f_{change} = f_{assigned} - P(\text{positive class}) \bullet f_{total}$$

where m_{change}, $m_{assigned}$ and m_{total} (and f) denote the change in the number of males (females) that receive a positive class label, the number of males (females) initially assigned a positive class, and the total number of males (females), respectively. It is straightforward to set the decision thresholds to values that result in these changes. Although this calculation is more efficient, we prefer using our algorithm since it provides an overview of the different threshold settings possible between the original and discrimination-free models. In addition to changing the decision thresholds, we remove the sensitive attribute from the Naive Bayes model.

17.4.2 Two Naive Bayes models

Using the above method, discrimination can be removed completely from a Naive Bayes classifier. However, it does not actively try to avoid the red-lining effect. Although the resulting classification is discrimination-free, this classification can still depend on the sensitive attribute in an indirect way. In our second approach, we try to avoid this dependence by removing all correlation with the sensitive attribute from the data-set used to train the Naive Bayes classifier.

Removing all correlation with the sensitive attribute from the data set seems difficult, but the solution actually is very simple. We divide the data-set into two sets, each containing people with only one of the sensitive values. Subsequently, we learn two Naive Bayes models from these two data sets. In the banking example, we thus get one model for the male and one for the female population. The model for males still uses attributes correlated to gender for making its decisions, but since it has not been trained using data from females; these decisions are not based on the fact that females are less likely to get positive labels. The predictions made using these models are therefore independent of the sensitive attribute. When classifying new people, we first select the appropriate model, and then use that model to decide on the class label.[5]
Intuitively, this approach makes a lot of sense since it uses different classifiers to classify data that is known to be differently distributed (males are different from females). Since males are still favored, however, the resulting classification can still contain discrimination. We apply the threshold modification algorithm to remove this discrimination.

17.4.3 A latent variable model.

Our third and most sophisticated approach tries to model the discrimination process in order to discover the actual class labels that the data-set should have contained if it would have been discrimination-free. Since they are not observed, these actual class labels are modeled using a latent (or hidden) variable, see, e.g., (Bishop, 2006). Such a latent variable can be seen as an attribute that is known to exist, but its values have not been recorded in the data-set. A well-known example of such a variable is “happiness”. It is very difficult to observe if someone is happy, but since we known how being happy influences one’s actions, we can infer whether someone is happy by observing his or her actions. In our case, we cannot know who should have gotten a positive class label, but we can make assumptions about how this variable depends on the other variables:

1. The actual discrimination-free class label is independent from the sensitive attribute.
2. The observed class label is determined by discriminating the actual labels based on the sensitive attribute uniformly at random.

These two assumptions might not correspond to how discrimination is being applied in practice. For instance, the females close to the decision boundary could have a higher chance of being discriminated. However, because they result in a simple model, they do allow us to study the problem of discrimination-free

5 It has been suggested to swap these models, i.e., use the model learned using males to classify females and vice versa. In our opinion, this makes less sense since this approach uses classifiers to classify data from different distributions. Also, in our experience it produces worse results.

classification in detail. The resulting model is given by the following total probability function:

$$P(C,L,S,A_1,A_2,\ldots,A_n) = P(L)P(S)P(C|L,S)P(A_1|L,S)P(A_2|L,S)\ldots P(A_n|L,S),$$

where C is the class attribute after discrimination, L is the latent variable representing the true class before discrimination, S is the sensitive attribute, and A_1, A_2,…, A_n are all other attributes. The formula is similar to the original Naive Bayes formula in the sense that all attributes A_1, A_2,…, A_n are independent from each other given the class label. Except that in this model, we use the actual latent class label L instead of C. In addition, every value except L is conditioned on the sensitive attribute S. The result is identical to the previous approach that used two separate models; for every value of S, a different set of probability functions are used, thus a different model is used for every value of S. The distribution of L however, is modeled to be independent from S, satisfying the first assumption. The probability function P(C|L,S) satisfies the second assumption: for every combination of an actual latent class label value with a sensitive value, a different probability function is used to determine the observed class label. Thus, the discrimination depends on both the actual class label, and on the sensitive value, but who is being discriminated is decided at random, i.e., independent of the other attribute values. We now show how to find likely latent class labels, i.e., how to discover who is likely being discriminated.

Finding likely latent values

We need to find good values to assign to the latent attribute in every row from the data-set. Essentially, this is a problem of finding two groups (or clusters) of rows: the ones that should have gotten a positive label, and those that should have gotten a negative label. We now briefly describe the standard approach of expectation maximization (EM) that is commonly used in order to find such clusters. The reader is referred to (Bishop, 2006) for a more detailed description of this algorithm.

Given a model M with a latent attribute L, the goal of the expectation maximization algorithm is to set the parameters of M such that they maximize the likelihood of the data-set, i.e., the probability of the data-set given the model. Unfortunately, since L is unobserved, the parameters involving L can be set in many different ways. Searching all of these settings for the most optimal one is a hopeless task. Instead, expectation maximization optimizes these settings by fitting them to the data-set (the M-step), then calculates the expected values of the latent attribute given those settings (the E-step), incorporates these back into the data-set, and iterates. This is a greedy procedure that converges to a local optimum of the likelihood function. Typically, random restarts are applied (randomizing the initial values of the latent variable) in order to find better latent values.

Using prior information

For the problem of finding the actual discrimination-free class labels we can do a lot better than simply running EM and hoping that the found solution corresponds to discrimination-free labels. For starters, it makes no sense to modify the labels of rows with favored sensitive values and negative class labels. The same holds for rows with discriminated sensitive values and positive class labels. Modifying these can only result in more discrimination, so we fix the latent values of these rows to be identical to the class labels in the data-set and remove them from the E-step of the EM algorithm.
Another improvement over blindly applying EM is to incorporate prior knowledge of the distribution P(C | L, S). In fact, since the ultimate goal is to achieve zero discrimination, we can pre-compute this entire distribution. We show how to do this using an example.

Example
Suppose we have a data-set consisting of 100 rows of people, distributed according to the following occurrence counts:

	Low income	High income
Female	30	20
Male	10	40

Clearly, there is some discrimination: the positive class probability of males (0.8) is much bigger than the positive class probability of females (0.4). Initially, we set the distribution over the latent labels to be equivalent to the distribution over the class labels, keeping the discrimination intact:

	Latent positive		Latent negative	
	Low income	High income	Low income	High income
Female	0	20	30	0
Male	0	40	10	0

Next, we rectify this situation by subtracting occurrence counts from the males with positive latent values, and giving these negative latent values. We do the opposite for females. Since we want the number of rows with actual non-discriminatory positive labels to be equal to the number of rows with positive labels in the data, the amount of such changes we need to make is unique and easy to compute. In the example, it is 10, resulting in the following distribution:

	Latent positive		Latent negative	
	Low income	High income	Low income	High income
Female	10	20	20	0
Male	0	30	10	10

In this table, both males and females have a probability of 0.6 to obtain a positive latent value. The latent values are therefore discrimination-free. We use these counts to determine the probability table P(C | L, S) in the latent variable model.

17.4.4 Comparing the three methods

In order to test the three Naive Bayes approaches for discrimination-free classification, we performed tests on both artificial and real-world data (Calders & Verwer, 2010). Here we made use of the latent variable model to generate the artificial data-sets. A big advantage of this artificial data is that we can also generate the actual class labels that should have been assigned to the rows when there is no discrimination. These labels are then used to test the accuracy of the classifiers. When using real-world data, we do not have this luxury of a discrimination-free test-set.
When performing such experiments with discrimination-aware methods, one should test at least the following quantities: the loss in accuracy and the amount of remaining discrimination. One always has to make a trade-off between these two values since discrimination can only be decreased by sacrificing accuracy. The main conclusions from experiments in (Calders & Verwer, 2010) are that our second threshold modifying method performs best, achieving zero discrimination with high accuracy. In addition, the expectation maximization algorithm has problems converging to a good quality solution with zero discrimination. In fact, during later iterations, it often finds solutions that are worse both in terms of discrimination and accuracy than solutions found earlier. This strange behavior of the EM algorithm still has to be further investigated. For a more detailed overview and discussion of these results, the reader is referred to (Calders & Verwer, 2010).

17.5 A note on positive discrimination

Although discrimination-aware data-mining is necessary in our opinion, one should be aware that it not only decreases the accuracy of data-mining, it also has a high probability to introduce positive discrimination. For instance, if we repeat the final analysis from Section 3 to results obtained using our first threshold modifying method (until zero discrimination) on the census income data-set, we obtain the following counts on people that should get a high income according to the test-set:

	Low income	High income
Female	101	489
Male	1763	1493

Table . *The gender-predicted income contingency table for high income test cases, assigned by a Naive Bayes classifier with modified decision thresholds.*

Suddenly, females have a much smaller probability of being falsely denied a high income. This is an example of positive discrimination, and in some countries this

type of discrimination is also considered illegal. These numbers, however, are determined using the discriminatory labels in the test-set. The actual difference in false negatives will be smaller using the true non-discriminatory class values. Unfortunately, since we do not know who is being discriminated, we cannot know exactly how to correct these numbers for this discrimination. We can, however, make an estimated guess based on the assumption that discrimination occurs at random, and that the number of positives should remain intact.

Under these assumptions, 690 females with a negative class label in the test-set should actually have a positive label, and 690 males with a positive label should actually have a negative label. The probability that a female is already assigned a positive label is equal to the false positive probability, which is 0.1683 (813 out of 4018). Thus, 690•0.1683=116 discriminated females get a positive label, and 574 discriminated females remain. Since these should get a positive label, these counts are added to the true and false negatives. For the male counts, some positives should actually be negatives. The false positive probability for males is 0.5415 (1763 out of 3256). Thus, 690•0.5415=374 favored males get a negative label, and 316 favored males remain. Since these counts should actually be negative, we subtract them from the counts in the table. This results in the following table:

	Low income	High income
Female	675	605
Male	1389	1177

Table . *The modified threshold gender-predicted income contingency table for discrimination corrected high income test cases.*

This corresponds to a probability of being denied a loan of 0.53 for females, and 0.54 for males. These probabilities are a lot more reasonable. Although they are based on the not always realistic assumption of equal treatment (and random discrimination), in our opinion, trying to make these false negative probabilities similar for males and females using positive discrimination is a lesser evil than knowingly making them unbalanced by blindly applying a discriminating data mining procedure.

17.6 Concluding remarks

We introduced the Naive Bayes classifier and argued that naively applying such a classifier to a data-set containing information regarding people automatically introduces discrimination with respect to sensitive attributes such as gender, race, and ethnicity. Using data mining tools in a decision support system based on such data can thus be considered very dangerous since it opens the possibility of law suits. We show using an example that the solution of removing this sensitive information from the data-set does not remove this discrimination. Since data-

mining tools use attributes that are correlated with this sensitive information, the decisions made by naively applying data-mining tools will still be discriminating. In fact, removing the sensitive information from a data-set makes the situation worse because data-mining tools will still discriminate, and rectifying this situation without access to sensitive information is extremely difficult. Instead, we introduce three discrimination-aware data-mining methods based on the Naive Bayes classifier that use the sensitive information in order to make non-discriminatory predictions.

In our first method, we use different decision thresholds for different sensitive values. For instance, we can decide to assign a positive class label to females if the positive class probability is greater than 0.1, but to males if it is greater than 0.6. We provide a simple algorithm for making modifications to these thresholds until the resulting classification is discrimination-free.

The second method involves learning two different classifiers; for instance one for all males, and one for all females. This effectively removes all correlation with the sensitive attribute from the data-set used to train the Naive Bayes classifier, thus avoiding that correlated attributes can be used to discriminate. Since there can still be discrimination in the resulting classification, we assign different decision thresholds to them using the algorithm of our first method. Of all three methods, this approach performed best in experiments on artificial and real-world data.

In the third and most involved method we introduced a latent variable reflecting the actual class of a person that should have been assigned if there were no discrimination. This actual non-discriminatory class is assumed to be independent of the sensitive attribute, and the non-discriminatory labels are assumed to be discriminated uniformly at random, resulting in the actual labels in the data-set. The probabilities in this model are learned using the expectation maximization algorithm. We provide ways to incorporate knowledge about the discrimination process into this algorithm. In experiments, this method unfortunately performed poorly due to problems in the behavior of the expectation maximization algorithm.

We ended with a discussion on the positive discrimination introduced by discrimination-aware data-mining and why we believe it is a better option than blindly applying a discriminating off-the-shelf data-mining procedure.

References

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning.* Springer.

Calders, T., Kamiran, F. & Pechenizkiy, M. (2009). Building classifiers with independency constraints. In *IEEE ICDM Workshop on Domain Driven Data Mining*, pp. 13-18, IEEE press.

Calders, T. & Verwer, S. (2010). Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2), 277-292, Springer, 2010.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D. & Weingessel, A. (2008). *e1071: Misc functions of the Department of Statistics*. TU Wien, R package version 1.

Kamiran, F. & Calders, T. (2009). Classifying without discriminating. In *Proc. IEEE International Conference on Computer, Control and Communication (IC4)*, pp. 1-6, IEEE press.

Lachiche, N. & P. Flach. (2003). Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. In *Proc. International Conference on Machine Learning (ICML)*, pp. 416-423, AAAI Press.

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. In *Proc. Conference on Artificial Intelligence (AAAI)*, pp. 223-228.

Pedreschi, D., Ruggieri, S. & Turini, F. (2008). Discrimination-aware data mining. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 560-568.