Building Classifiers with Independency Constraints

Toon Calders Faisal Kamiran Mykola Pechenizkiy
Eindhoven University of Technology, The Netherlands
E-mail: {t.calders,f.kamiran,m.pechenizkiy}@tue.nl

Abstract

In this paper we study the problem of classifier learning where the input data contains unjustified dependencies between some data attributes and the class label. Such cases arise for example when the training data is collected from different sources with different labeling criteria or when the data is generated by a biased decision process. When a classifier is trained directly on such data, these undesirable dependencies will carry over to the classifier's predictions. In order to tackle this problem, we study the classification with independency constraints problem: find an accurate model for which the predictions are independent from a given binary attribute. We propose two solutions for this problem and present an empirical validation.

1. Introduction

Classifier construction is one of the most researched topics within the data mining and machine learning communities. Literally thousands of algorithms have been proposed. The quality of the learned models, however, depends critically on the quality of the training data. No matter which classifier inducer is applied, if the training data is incorrect, poor models will result. In this paper we want to study cases in which the input data contains dependencies between some of its attributes and the class label that are either incorrect, or undesirable. Such cases occur naturally when the decision process leading to the labels was biased, as illustrated by the next example. *Throughout the years, an employment bureau recorded various parameters of job candidates. Based on these parameters, the company wants to learn a model for partially automating the match-making between a job and a job candidate. A match is labeled as successful if the company hires the applicant. It turns out, however, that the historical data is biased; for higher board functions, Caucasian males are being favored systematically. A model learned directly on this data will learn this discriminatory behavior and apply it for future predictions.* In this case it is desirable to have a mean to "tell" the algorithm that its predictions should be independent of the attributes sex and ethnicity. We want to integrate this domain knowledge in the discovery process.

The original idea of requiring independent predictions stems from [5] where it was introduced in the context of discrimination. [7, 6] introduced similar concept but they concentrated on identifying the discriminatory rules that are present in a dataset rather than on learning a classifier with independency constraint for future predictions. Here we will concentrate on the case where a labeled dataset is given, and one boolean attribute B which we do not want the predictions to correlate with. The dependency of the predictions of a classifier C on the attribute B is defined as

$$P\big(C(x) = + \mid x(B) = 1\big) - P\big(C(x) = + \mid x(B) = 0\big)$$

for unseen tuples x. A positive dependency will reflect that a tuple for which B is 1 has a higher chance of being assigned the positive label by C than one where B equals 0. The straightforward solution of the problem is removing the attribute B from the training-set. In most cases, this solution does not solve the problem. For example, removing *sex* and *ethnicity* for the job-matching example might not help, as other attributes such as *residential area* may be correlated with it. Blindly applying an out-of-the-box classifier on the job-matching data without the ethnicity attribute will in such a situation still lead to a model that discriminates indirectly based on residential area. A parallel can be drawn with the practice of *redlining*: denying inhabitants of certain racially determined areas from services such as loans.

The problem of classification with independency constraints is in fact a multi-objective optimization problem; on the one hand the more dependency we allow for, the higher accuracy we can obtain and on the other hand, in general, we can trade in accuracy in order to reduce the dependency. In this paper we propose two methods for incorporating independency constraints into the classifier construction process. The first method *Massaging* the dataset changes some labels in the dataset in order to remove the dependency between the class labels and the attribute B. This method was introduced in [5] and extended here. The second method *Reweighing* assigns weights to tuples instead of changing

978-0-7695-3902-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDMW.2009.83

the labels. These weights are used to balance the original biased training dataset. On this balanced dataset the dependency-free classifier is learned. An empirical study is given in Section 4 which shows promising results.

2. Problem Statement

We formally introduce the notion of an independency constraint.

We assume a set of attributes $\{A_1, \ldots, A_n\}$ and their respective domains $dom(A_i)$, $i = 1 \ldots n$ have been given. A *tuple* over the schema $(A_1, \ldots, A_n)$ is an element of $dom(A_1) \times \ldots \times dom(A_n)$. A dataset over the schema $(A_1, \ldots, A_n)$ is a finite set of such tuples and a labeled dataset is a finite set of tuples over the schema $(A_1, \ldots, A_n, Class)$. Throughout the paper we will assume $dom(Class) = \{-, +\}$.

Let a labeled database D, an attribute B and a value $b \in dom(B)$ be given. The *dependency* between $B = b$ and $Class$ in D, denoted by $dep_{B=b}(D)$, is defined as the difference of the probability of being in the positive class between the tuples having $B = b$ in D and those having $B \neq b$ in D; that is:

$$dep_{B=b}(D) := \frac{|\{x \in D \mid x(B) \neq b, x(Class) = +\}|}{|\{x \in D \mid x(B) \neq b\}|} - \frac{|\{x \in D \mid x(B) = b, x(Class) = +\}|}{|\{x \in D \mid x(B) = b\}|} .$$

When clear from the context we will omit $B = b$ from the subscript in $dep_{B=b}(D)$. A positive dependency means that tuples with $B = b$ are less likely to be in the positive class than tuples with $B \neq b$.

The problem we study in the paper is now as follows: given a labeled dataset D, an attribute B, and a value $b \in dom(B)$, learn a classifier C such that:

(a) the accuracy of C for future predictions is high; and

(b) the dependency between $B = b$ and $Class$ is low.

Clearly there will be a trade-off between the accuracy and the dependency of the classifier. In general, lowering the dependency will result in lowering the accuracy as well and vice versa. In this paper we are making three strong assumptions:

A1 The primary intention is learning the most accurate classifier for which the dependency is 0.

A2 The learned classifier should not use the attribute B to make its predictions.

A3 The total ratio of positive predictions of the learned classifier should be equal to the ratio of positive labels in the dataset D.

We do not claim that other settings where these assumptions are violated are not of interest, but at the current stage our work is restricted to this setting.

3. Solutions

In this section we propose two solutions to learn a classifier with independency constraint that does not use the attribute B to make its predictions. Both solutions are based on removing the dependency from the training dataset. On this cleaned dataset a classifier can be learned. Our hypothesis is that, since the classifier is trained on balanced data, its predictions will be (more) balanced as well. The empirical evaluation in Section 4 will confirm this statement. The first approach we present, called *Massaging the data*, is based upon changing the class labels in order to remove the dependency between $B = b$ and $Class$. A preliminary version of this approach was presented in [5]. The second approach is less intrusive as it does not change the class labels. Instead, the dataset is re-sampled in such a way that the dependency is removed. This approach will be called *Reweighing*. In the experimental section we will also consider a third, very straightforward option: instead of cleaning the training set, the attribute B and its most correlated attributes are removed from the dataset. It will be shown, however, that this method is almost always inferior to the two methods proposed in this section.

3.1. Massaging

In *Massaging*, we want to remove the dependency between B and the class attribute from the dataset. In order to do this, we will change the labels of some objects x with $x(B) = b$ from '$-$' to '$+$', and the same number of objects with $x(B) \neq b$ is changed from '$+$' to '$-$'. Our *Massaging* approach has some similarity to [3] with respect to relabeling the dataset. From the proof of Theorem 1 in [2] we know that in this way we can reduce the dependency with the minimal number of changes to the dataset while keeping the overall positive class ratio constant. The set pr of objects x with $x(B) = b$ and $x(Class) = -$ will be called the *promotion candidates* and the set dem of objects x with $x(B) \neq b$ and $x(Class) = +$ will be called the *demotion candidates*.

We will not randomly pick promotion and demotion candidates to relabel. Instead a ranker will be used to select the best tuples as follows. On the training data, a ranker R for ranking the objects according to their positive class probability is learned; i.e., the higher on the ranking an object x is, the more probable it is that $x(Class) = +$. With this ranker, the promotion candidates are sorted according to descending rank by R and the demotion candidates according to ascending rank. When selecting promotion and

Sex	Ethnicity	Highest Degree	Job Type	Cl.	Prob
m	native	h. school	board	+	98%
m	native	univ.	board	+	89%
m	native	h. school	board	+	98%
m	**non-nat.**	**h. school**	**healthcare**	**+**	69%
m	non-nat.	univ.	healthcare	-	30%
f	non-nat.	univ.	education	-	2%
f	**native**	**h. school**	**education**	**-**	40%
f	native	none	healthcare	+	76%
f	non-nat.	univ.	education	-	2%
f	native	h. school	board	+	93%

Table 1. Sample job-application relation with positive class probability.

Sex	Ethnicity	Highest Degree	Job Type	Cl.	Prob
f	native	h. school	education	-	40%
f	non-nat.	univ.	education	-	2%
f	non-nat.	univ.	education	-	2%
Sex	Ethnicity	Highest Degree	Job Type	Cl.	Prob
m	non-nat.	h. school	healthcare	+	69%
m	native	univ.	board	+	89%
m	native	h. school	board	+	98%
m	native	h. school	board	+	98%

Table 2. Promotion candidates (negative objects with $Sex = f$ in descending order) and demotion candidates (positive objects with $Sex = m$ in ascending order)

demotion candidates, first the top elements will be chosen. In this way, the objects closest to the decision border are selected first to be relabeled, leading to a minimal effect on the accuracy.

The modification of the training data is continued until the dependency in it becomes zero. The number of modifications M required to make the data dependency-free can be calculated by using the following formula:

$$M = \frac{(b \times \overline{b} \wedge +) - (\overline{b} \times b \wedge +)}{b + \overline{b}}$$

where b and $\overline{b}$ represent respectively the number of objects with $B = b$ and $B \neq b$ while $b \wedge +$ and $\overline{b} \wedge +$ are the number of objects with label '+' and $B = b$ or $B \neq b$ respectively. The formal description of the algorithm can be found in [2, 5].

Example 1. *We consider an example dataset given in Table 1. This dataset contains the Sex, Ethnicity and Highest Degree of* 10 *job applicants, the Job Type they applied for and the Class defining the outcome of the selection procedure. In this dataset, the dependency between* Sex *and Class will be* $dep_{Sex=f}(D) := \frac{4}{5} - \frac{2}{5} = 40\%$ *. In other words, a data object with* $Sex = f$ *will have* 40% *less chance of getting a job than one with* $Sex = m$*. We want to learn a classifier to predict the class of objects for which the predictions are independent of* $Sex = f$*. In this example we rank the objects by their positive class probability given by a* Naive Bayesian *classification model. In Table 1, the positive class probabilities as given by this ranker are added to the table (calculated by using NBS implementation of Weka).*

In the second step, we arrange the data separately for female *applicant with class '−' in descending order and for* male *applicants with class '+' in ascending order with respect to their positive class probability. The ordered promotion and demotion candidates are given in Table 2.*

The number M *of labels of promotion and demotion candidates we need to change equals:*

$$M = \frac{(f \times (m \wedge +)) - (m \times (f \wedge +))}{f + m} = \frac{(5 \times 4) - (5 \times 2)}{5 + 5} = 1$$

So, 1 *change from the promotion candidates list and one from the demotion candidates list will be required to make the data independent. We change the labels of the top promotion and demotion candidates (rows highlighted with the bold font in Table 1). After the labels for these instances are changed, the dependency level will decrease from* 40% *to* 0%*. So, the dataset (which will be used for future classifier learning) becomes dependency-free.* □

The *Massaging* approach is rather intrusive as it changes the labels of the objects. Our second approach does not have this disadvantage.

3.2. Reweighing

Instead of relabeling the objects, the *Reweighing* approach attaches different weights to them. For example, objects with $B = b$ and $Class = +$ will get higher weights than objects with $B = b$ and $Class = -$ and objects with $B \neq b$ and $Class = +$ will get lower weights than objects with $B \neq b$ and $Class = -$. According to these weights the objects will be sampled (with replacement) leading to a dataset without dependency. We will refer to this method as *Reweighing*. Again we will assume that we want to reduce the dependency to 0 while maintaining the overall positive class probability.

We discuss the idea of weight calculation by recalling some basic notions of probability theory with respect to this particular problem setting: If the dataset D is unbiased, in the sense that B and $Class$ are independent of each other, the expected probability $P_{exp}(b \wedge +)$ would be:

$$P_{exp}(b \wedge +) := b \times +$$

where b is the fraction of objects having $B = b$ and '$+$' the fraction of tuples having $Class = +$. In reality, however, the actual probability

$$P_{act}(b \wedge +) := b \wedge +$$

might be different.

If the expected probability is higher than the actual probability value, it shows the bias towards class '$-$' for $B = b$. We will assign weights to b with respect to class '$+$'. The weight will be

$$W(B = b \mid x(Class) \ = \ +) := \frac{P_{exp}(b \wedge +)}{P_{act}(b \wedge +)}.$$

This weight of b for class '$+$' will over-sample objects with $B = b$ for the class '$+$'. The weight of b for class '$-$' will be

$$W(B = b \mid x(Class) \ = \ -) \ := \ \frac{P_{exp}(b \wedge -)}{P_{act}(b \wedge -)}$$

and the weights of $\overline{b}$ for class '$+$' and '$-$' will also be calculated in the similar way.

In this way we assign to every tuple a weight according to its B- and $Class$-values. The balanced dataset is then created by sampling the original training data, with replacement, according to the assigned weights. On this balanced dataset the dependency-free classifier is learned. Our *Reweighing* technique can be seen as an instance of cost-sensitive learning [4] in which, e.g., an object of class '$+$' with $B = b$ gets a higher weight and hence an error for this object becomes more expensive. The pseudocode of the algorithm describing our *Reweighing* approach in detail can be found in [2, 5].

Example 2. *Now we use the* Reweighing *scheme to remove the dependency between Sex attribute and $Class$ attribute from the data of Table 1. We calculate a weight for each data object according to its B- and $Class$-value. We observe that in this particular example where both $B = Sex$ and the $Class$ attribute are binary attributes. Only four combinations between the values of B and the $Class$ attribute are possible, i.e., $B = f$ or $B = m$ can have $Class$-values '$+$' or '$-$'. So weights of these four combinations will be sufficient for the whole data. For instance, we calculate the weight of a data object with $B = f$ and $Class$ '$+$'. We know that 50% objects have $B = f$ and 60% objects have $Class$-value '$+$', so the expected probability of the object should be:*

$$P_{exp}(Sex = f \mid x(Class) \ = \ +) = 0.5 \times 0.6$$

but its actual probability is 20%. So the weight W will be:

$$W(Sex = f \mid x(Class) \ = \ = \ \frac{0.5 \times 0.6}{0.2} = 1.5 \ .$$

Similarly the weights for the other combinations are:

$$\begin{aligned} W(Sex = f \mid x(Class) \ = \ -) \ &= \ 0.67 \\ W(Sex = m \mid x(Class) \ = \ +) \ &= \ 0.75 \\ W(Sex = m \mid x(Class) \ = \ -) \ &= \ 2 \ . \end{aligned}$$

The weight of each individual data object of the Table 1 is given in Table 3.

Sex	Ethnicity	Highest Degree	Job Type	Cl.	Weight
m	native	h. school	board	+	0.75
m	native	univ.	board	+	0.75
m	native	h. school	board	+	0.75
m	non-nat.	h. school	healthcare	+	0.75
m	non-nat.	univ.	healthcare	-	2
f	non-nat.	univ.	education	-	0.67
f	native	h. school	education	-	0.67
f	native	none	healthcare	+	1.5
f	non-nat.	univ.	education	-	0.67
f	native	h. school	board	+	1.5

Table 3. Sample job-application relation with weights.

4. Experiments

In this section we present experiments that support the following **claims**:

1. Due to the red-lining effect it is not enough to just remove the attribute B from the dataset in order to remove the dependency with the class attribute. Also removing B and the attributes that correlate with it does not have the desired effect, as either too much dependency remains or the accuracy is lowered too much.

2. Both proposed solutions get better results in the sense that they more optimally trade accuracy for independency. Especially the *Massaging* approach, if initiated with the right choice of ranker and base learner shows potential.

3. More concretely, when the goal is to reduce the dependency to zero while maintaining a high accuracy, a good ranker with a base learner that is sensitive to small changes in the dataset seems to be the best choice.

Experimental setup. In our experiments we used the Census Income dataset and the German Credit datasets which are both available in the UCI ML-repository [1]. As the results and conclusions we can draw from them are very similar for both datasets, we only present the figures for the Census Income dataset. The results for the German Credit dataset can be found in [5]. Census Income has 48842 instances of which we used only a random sample of 1/3 for reasons of efficiency. Census Income contains demographic information about people and the associated prediction task is to determine whether a person makes over 50K per year or not, i.e., income class *High* or *Low* will be predicted. We will denote income class *High* as '$+$' and income class *Low* as '$-$'. Each data object is described by 14 attributes which include 8 categorical and 6 numerical attributes. We excluded the attribute *fnlwgt* from our experiments (as suggested in the documentation of the dataset). The other attributes in the dataset include: age, type of work, education, years of education, marital status, occupation, type of relationship (husband, wife, not in family), sex, race, native country, capital gain, capital loss and weekly working hours. We use $Sex = f$ as dependent attribute. In our sample of the dataset, 5421 citizens have $Sex = f$ and 10860 have $Sex = m$. This dependency between $Sex = f$ and $Class$ is as high as 19.13%; i.e.,

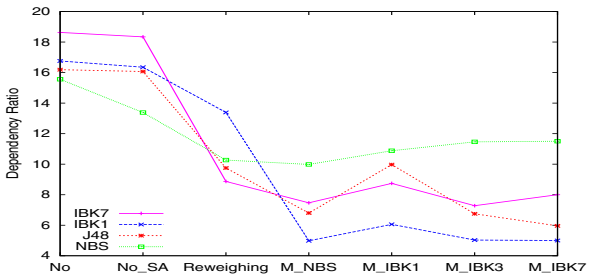
$$P(x(Class) = + \mid x(Sex) = m) - P(x(Class) = + \mid x(Sex) = f) \; = \; 19.13\%$$

The goal is now to learn a classifier that has minimal dependency between $Sex = f$ and its predictions while maintaining a high accuracy. All reported accuracy numbers in the paper were obtained using 10-fold cross-validation and reflect the accuracy; that is, on non-massaged test data.

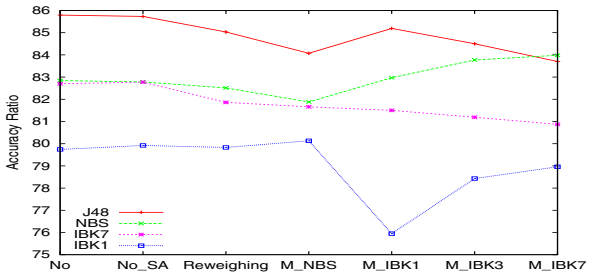
4.1. Testing the Proposed Solutions

We conducted experiments to test our proposed solutions. We compare three different types of algorithms:

1. Two **baseline** approaches: an out-of-the-box classifier was learned on at the one hand the original data (labeled "No" in the graphs to reflect no *Preprocessing* technique was applied) and on the other hand the original data but with the attribute Sex removed (labeled "No_SA" (Sex Attribute) in the graphs).



(a) Baseline dependency=19.3



(b) Baseline accuracy=76.3

Figure 1. The results of 10-fold CV.

2. The **Massaging** approach with different combinations of base learner and ranker. We consider four different rankers: one based on a Naïve Bayesian classifier (M_NBS), and three based on nearest neighbors with respectively 1, 3 and 7 neighbors (M_IBk1, M_IBk3, and M_IBk7). For the base classifiers that are learned on the massaged data, a Naïve Bayes Classifier (NBS) was used, two nearest neighbor classifiers with respectively 1 and 7 neighbors (IBk1 and IBk7), and a decision tree learner: the Weka implementation of the C4.5 classifier (J48). Many more combinations have been tested (including Adaboost and all possible combinations) but we will restrict to these choices as they present a good overview of the obtained results. An overview of all results can be found in [2].

3. The **Reweighing** approach with different base classifiers (labeled "Reweighing" in the graphs).

In Figures 1(a) and 1(b), respectively the dependency and accuracy results for all algorithms under comparison are given. The X-axis shows the names of the data preprocessing techniques which have been applied to the training dataset to remove undesirable dependencies between *Sex* and *Income* class attribute. The dependency of the resultant classifiers learned on this data has been given on the Y-axis of Figure 1(a) and their accuracy on the Y-axis of Figure 1(b). We observe that the classification models with independency constraint produce less dependent results as compared to the baseline algorithms; in Figure 1(a) we see

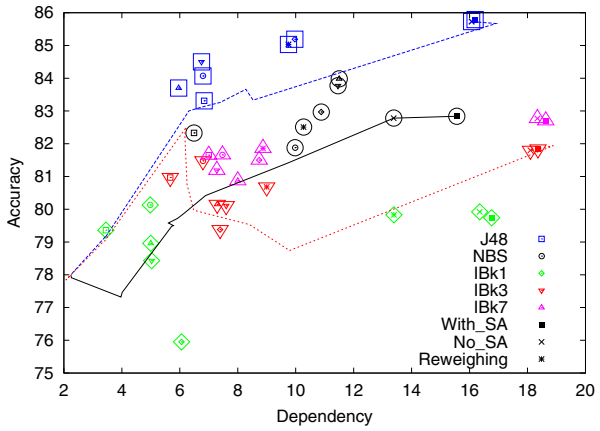


Figure 2. Accuracy-dependency trade-off. *Outer and inner symbol of each data point shows the corresponding base learner and preprocessing technique respectively*.

that IBk1 classifies the future data objects with a dependency level of 16.76% which is lowered only slightly if the Sex attribute is removed. If *Massaging* is applied, however, the dependency level goes down to 4.98%. The dependency level always goes down when we apply our classifiers with independency constraint. Clearly, the choice of base learner and ranker (for *Massaging*) plays a very important role in dependency free classification. The accuracy drops to some extent because our test set contains these undesirable dependencies.

Figure 2 offers a good overview that allows us to quickly assess which of the combinations are dependency-accuracy-optimal in the class of the classifiers learned in our experiments. Each pictogram in this figure represents a particular combination of a classification algorithm (shown by outer symbol) and corresponding preprocessing technique (shown by inner shape of the data point). For *Massaging*, the inner symbol will represent the corresponding ranker. On the X-axis we see the dependency and on the Y-axis, the accuracy. Thus, we can see the trade-off between accuracy and dependency for each combination. The closer we are to the the top left corner the higher accuracy and the lower dependency we obtain. We observe that the top left area in the figure is occupied with the points corresponding to the performance of *Massaging* approach. *Reweighing* approach falls behind *Massaging* but also shows reasonable performance. From Figure 2 we can see that our both approaches compare favorably to the baseline and the simplistic solutions: the three lines in the figure represent three classifiers (J48, NBS and IBk3 from the top to bottom) learned on the original dataset (the most top-right point in each line, denoted with $With\_SA$ symbol), the original dataset with the *Sex* attribute removed (denoted with No_SA symbol), the original dataset with the *Sex* attribute and the one (two, three, and so on) most correlated attribute(s) removed (that typically correspond to the further decrease in both accuracy and dependency). We see that this simplistic solution is dominated by our classification with independency constraints approaches.

Overall, we hypothesize that *Massaging* has more impact on a noise-sensitive classifier, e.g., J48 than noise-tolerant classifiers, e.g., NBS. When we remove dependencies from training data, this effect is transferred to future classification in case of noise-sensitive classifiers and both the dependency level and the accuracy goes down more than for a noise-tolerant classifier. So, if the minimal dependency is the first priority, a noise-sensitive classifier is the better option and if the high accuracy is the main concern, a noise-tolerant classifier might be more suitable.

5. Conclusion and Discussion

In this paper we presented the classification with independency constraints problem. Two approaches towards the problem were proposed: *Massaging* and *Reweighing* the dataset. Both approaches remove the dependency from the training data and the claim is that a classifier learned on this unbiased data will be less biased itself. Experimental evaluation shows that indeed this approach allows for removing dependency from the dataset more effectively than simple methods such as, e.g., removing the dependent attribute from the training data. All methods have in common that to some extent accuracy must be traded-off for lowering the dependency.

To conclude, we believe that classification with independency constraints is a new and exciting area of research addressing a societally relevant problem.

References

[1] A. Asuncion and D. Newman. UCI machine learning repository. 2007.

[2] T. Calders, F. Kamiran, and M. Pechenizkiy. Building classifiers with independency constraints. technical report. 2009.

[3] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *KDD*, pages 155–164, 1999.

[4] C. Elkan. The foundations of cost-sensitive learning. In *Proc. IJCAI'01*, pages 973–978, 2001.

[5] F. Kamiran and T. Calders. Classifying without discriminating. In *Proc. IC4'09*. IEEE press.

[6] D. Pedreschi, S. Ruggieri, and F. Turini. Measuring discrimination in socially-sensitive decision records. In *Proc. SIAM SDM'09*.

[7] D. Pedreschi, S. Ruggieri, and F. Turini. Discrimination-aware data mining. In *Proc. ACM SIGKDD'08*, 2008.