Learning with Actionable Attributes: Attention – Boundary Cases!

Indrė Žliobaitė and Mykola Pechenizkiy
Eindhoven University of Technology, Eindhoven, the Netherlands
Email: (i.zliobaite,m.pechenizkiy)@tue.nl

***Abstract*—Traditional supervised learning assumes that instances are described by observable attributes. The goal is to learn to predict the labels for unseen instances. In many real world applications the values of some attributes are not only observable, but can be proactively chosen by a decision maker. Furthermore, in some of such applications the decision maker is interested not only to generate accurate predictions, but to maximize the probability of the desired outcome. For example, a direct marketing manager can choose the color of an envelope (actionable attribute), in which the offer is sent to a client, hoping that the right choice will result in a positive response with a higher probability. We study how to learn to choose the value of an actionable attribute in order to maximize the probability of a desired outcome in supervised learning settings. We emphasize that not all instances are equally sensitive to change in actions. Accurate choice of an action is essential for those instances, which are on a borderline (e.g. do not have a strong opinion). We formulate three supervised learning approaches to select the value of an actionable attribute at an instance level. We focus the learning process to the borderline cases. The potential of the underlying ideas is demonstrated with synthetic examples and a case study with a real dataset.**

***Keywords*-actionable attributes; supervised learning; change**

I. Introduction

In supervised learning it is typically assumed that the values of attributes in the attribute space are given and the task is to predict the values of the target attribute (e.g. class labels). However, in a number of predictive analytics applications the attribute space consists not only of observable but also of actionable attributes. Consider a direct marketing task: given an individual to be provided with an information about cable TV subscription, the task is to predict how likely it is that (s)he will subscribe it. Given a historical dataset with known ground truth (which individuals responded positively, which did not), the task is to support business decisions, to which customers what offer to send, and to whom not to send any offer at all.

In the considered case, the attribute space may include the attributes, the values of which can be influenced by a decision maker (*actionable* attributes). For a simple example, a marketer may choose whether to send an offer in a blue or yellow envelope when thinking how to maximize the probability of the desired outcome (in this case subscription).

The subject of this study is *learning to select* the value of the actionable attribute for a given instance, to maximize the probability of a desired outcome. That is, in the direct marketing example, given an individual, we learn to choose, what envelope to use (actionable attribute), in order to maximize the probability of a positive response (target attribute).

This formulation is motivated by an observation, that customers are sensitive to the actionable attributes in different ways. Some customers will certainly not accept the offer, and others will certainly accept it no matter whether yellow or blue envelope is used. Yet, there will be some group of customers, for whom the final decision may be affected by the color of an envelope. An important assumption implicitly made here is that there is no globally better action. We assume that for *some* customers yellow, and for *some* customers blue collor is more attractive, while *others* do not care, see Figure 1.

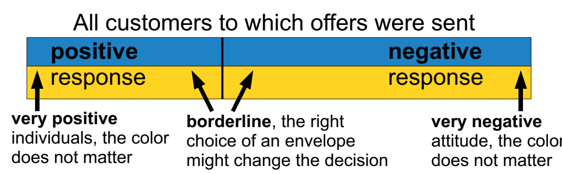


Figure 1. Illustration of the set up, a direct marketing example.

Accurate choice of an action is essential for those instances, which are on the borderline, including, but not limited (as we show in the figure) to a decision boundary between positive and negative instances.

The task of learning a decision rule is to decide, what value of an actionable attribute should be selected for a customer in consideration. It is not trivial, since in the historical data the information whether the customer accepted the offer because of being affected by the correct action (envelope color) or for some other reason typically is not available. Thus, the training sample for this learning task is hidden and needs to be constructed from the data. In this work we study how to focus learning process to the borderline cases.

The two main contributions of this study are: (1) formulation and experimental exploration of supervised learning with actionable attributes specialized in borderline cases; (2) introducing three heuristic approaches for learning action rules with focused training process.

The paper is organized as follows. In Section II we define and discuss the problem and assumptions taken in this study. In Section III we introduce three supervised learning approaches to select the value of an actionable attribute at an instance level. Section IV overviews related work. Section V presents our experimental case study. In Section VI we discuss complementary and further research scenarios and challenges. Section VII concludes.

II. Problem set up

Consider the following set up: given a set of instances with labels $(\mathbf{Z}, \mathbf{y})$ there is a mapping $\mathcal{L}: Z \rightarrow y$ that maps any instance Z in p-dimensional space to its true label y in one-dimensional space. Consider also, that some value of the label is desired to be achieved, denote it y^*.

In predictive analytics there are situations when some attributes are under control of a decision maker. Let $Z = \{XA\} = \{x^{(1)}x^{(2)} \ldots x^{(m)}a^{(1)} \ldots a^{(k)}\}$, is an instance in p-dimensional input space, where there are m observable attributes and k actionable attributes, $m + k = p$.

Actionable attribute $a^{(j)}$ is an attribute under control, the value of which can be changed by actions of a decision maker prior to predicting the label. The values of an *observable attribute* $x^{(i)}$ cannot be changed.

For example, a type of treatment for a particular patient can be decided by a doctor, while the age of the patient is only observable. Examples of actionable attributes in predictive analytics:

- credit amount in credit scoring;
- complementary products, discounts or a form of an offer in direct marketing;
- the type of a recommendation approach or actual shown items in recommender systems;
- promotion activities in sales quantity prediction;
- bidding amount in the sponsored search advertisement.

In this study we assume that there is only one actionable attribute a. The task is to learn a decision rule $\mathcal{H}: X \rightarrow a$, where a is an actionable attribute and X is an instance in the input space without the actionable attribute.

The task is not trivial, as the true outcomes for alternative actions with the same instance are not known. Conceptually, we would need to observe the results of different actions to identical instances. Therefore, there is no ground truth, what happens, if the same individual is offered an alternative proposal. The effects of actions to the outcome need to be learned from the data.

At best, to collect such kind of training data one can employ A\B testing procedure.[1] In marketing A\B testing is realized by stratified sampling. That is, alternative samples are distributed to customer strata (for example, stratified w.r.t. age). If one value of the actionable attribute is globally better for achieving the desired outcome y^*, that will be reflected in the proportions of the collected response labels.

This approach would not reveal the subgroups, which react to actions in different ways. It might be the case that for a particular group of people yellow envelope leads to a better response rate, while for other group blue leads to a better response rate. We are interested to identify and learn local dependencies w.r.t. the actionable attribute. We illustrate this further with a toy dataset.

[1]A\B test refers to a direct comparison between two design alternatives in a controlled experiment.

Table I
A summary of strategies and responses for the toy example.

strategy	$P(+\|M)$	$P(+\|F)$	$P(+)$	expected '+'
random	0.2	0.4	0.3	6
all yellow	0.4	0.2	0.3	6
all blue	0	0.6	0.3	6
selective	0.4	0.6	**0.5**	**10**

Consider as a toy example a direct marketing case, where offers can be sent either in blue (B) or in yellow (Y) envelope. Offers were sent to 20 persons (10 male (M) and 10 female (F)) choosing the envelope color at random. The color of an envelope is the actionable attribute.

The marketer got *six* positive responses ($P(+) = 0.3$), three from blue and three from yellow. thus, there was no globally better envelope, $P(+|Y) = P(+|B) = 0.3$. However, there were subgroups, which responded to the colors differently. Female responded better to blue envelopes, while male respond better to yellow envelopes. The following response probabilities were observed: $P(+|F, Y) = 0.2$, $P(+|F, B) = 0.6$, $P(+|M, Y) = 0.4$ and $P(+|M, B) = 0$.

Thus, the marketer instead of random can send only blue envelopes to female and only yellow envelopes to male customers. In this case, the expected the number of positive responses would increase from *six* to *ten*. A summary of the strategies and responses is provided in Table I, assuming a new sample of 20 individuals with equal proportions of males and females.

Knowing the sensitivity of different subgroups the marketer can optimize the strategy further. The probability is maximized if an offer is sent *only* to females in blue envelopes, then $P(+) = 0.6$. Sending to several (not all) individuals might be preferred if sending has high costs.

Can the marketer do even better than that? Likely, if we can rank the instances based on their sensitivity. An example will be presented in the next section.

III. Three approaches to choose the action

We formulate and analyze three heuristic approaches to decide upon the value of an actionable attribute in a supervised learning manner. The first approach uses a classification rule to choose the action (try-and-see). The second approach reverses the task, including the outcome into the input space. The third one constructs a training set, treating the outcome as a context. Employing these approaches we will experimentally explore a focused learning technique for learning to choose the actions.

input: labeled training data $(\mathbf{X}, \mathbf{a}, \mathbf{y})$.
output: action rule $\mathcal{H}_{wrapper} : X \rightarrow a$.

1) Learn a mapping $\mathcal{L} : (Xa) \rightarrow y$ using $(\mathbf{X}, \mathbf{a}, \mathbf{y})$.
2) For $i \in$ domain(a) output $\hat{y}_i = (X, a_i)$.
3) Action $\mathcal{H}_{wrapper} : a = \arg\min_i(\text{distance}(y^*, \hat{y}_i))$.

Figure 2. Try-and-see wrapper approach.

input: labeled training data $(\mathbf{X}, \mathbf{a}, \mathbf{y})$.
output: action rule $\mathcal{H}_{twist} : X \rightarrow a$.

1) Learn a mapping $\mathcal{H}_{twist} : (Xy) \rightarrow a$ using $(\mathbf{X}, \mathbf{a}, \mathbf{y})$.
2) Action $\mathcal{H}_{twist} : (Xy^*) \rightarrow a$,
   where y^* is fixed for all the unseen X.

Figure 3. Twist approach.

A. Try-and-see wrapper approach

Wrapper approach does not learn the action rule $\mathcal{H}$ explicitly. Rather it learns the mapping between the instance and the outcome, where instance includes the actionable attribute $\mathcal{L} : (Xa) \rightarrow y$. The historical experimental data is used for learning as is, no modifications are introduced.

Given an unseen instance different values of an actionable attribute are tried and the respective model outputs are observed. The action, which leads to the output closest to the desired y^* is selected. The approach is summarized in Figure 2.

The approach has several limitations.

1) There is no explicit action rule. Practically the alphabet of possible actions should be limited.
2) Maximum and minimum values of a need to be controlled in case a is numerical.
3) The type of a predictor $\mathcal{L}$ assumes the relationship between a and y. For instance, linear models will learn either positive or negative relation between a and y. Thus, in the marketing example, yellow envelope will be assumed always better or always worse.

We illustrate the approach using the previous toy example with different base classifiers. The results are presented in Table II. It can be seen that linear classifiers do not make a distinction between the envelopes at all, since the decision boundary is non linear, as it is a XOR type classification problem. The desired outcome is $P(+|F, Y) < P(+|F, B)$ and $P(+|M, Y) > P(+|M, B)$, the classifiers that satisfy it are marked in bold. Non linear crisp output classifiers (SVM, kNN) correctly identify the color for female, but not male, since negative response is dominated among males. The regression tree correctly identifies the posterior probabilities, quadratic discriminant and neural network give approximate probabilities, but correct in terms of the desired outcome.

B. Twist approach

This approach adds the outcome to the attribute space and treats the actionable attribute as a label. The task now is to learn a mapping $\mathcal{H}_{twist} : (X, y) \rightarrow a$.

For an unseen instance we do not know, what the outcome will be, but we need to augment the input space. Thus we add the desired value of the outcome $y^u = y^*$ instead. The approach is summarized in Figure 3.

The approach has a potential limitation due to the possibility to create non-existing or very unlikely instances. When augmenting the input space it assumes that for every instance X^u the desired outcome $y^u = y^*$ is possible, which might not be true. It is not a problem as long as the remaining outcomes $(\neg y^*)$ are equally bad (or we are dealing with binary classification).

We illustrate the approach using the previous toy example with different base classifiers. Now we train the classifiers to output the color of the envelope, given a gender and a response using the original historical data. When testing, we always input the value of a desired response, which is '+'. The results are presented in Table III. Here the desired outcome is $P(Y|F, +) < P(B|F, +)$ and $P(Y|M, +) > P(B|M, +)$, the methods that satisfy it are marked in bold. Again, linear classifiers are not able to capture the subgroups.

C. Contextual approach

In this approach we treat the outcome as a context. The customers, who responded positively ('+'), form one context. The ones, who did not respond ('–'), form another context.

We want to *push* a given instance into the desired context by manipulating the value of the actionable attribute. For that we construct a training set to learn the action rule $\mathcal{H}_{context}$, defining the labels in a specific way. For a binary actionable attribute we make the following assumptions:

Table II
WRAPPER APPROACH RESULTS WITH THE TOY EXAMPLE.

classifier	$P(+\|F, Y)$	$P(+\|F, B)$	$P(+\|M, Y)$	$P(+\|M, B)$
Naive Bayes	0.39	0.39	0.21	0.21
log. regression	0.40	0.40	0.20	0.20
lin. discriminant	0.60	0.60	0.38	0.38
quad. discriminant	0.27	**0.79**	**0.63**	0.04
SVM (rbf)	0	**1**	0	0
regression tree	0.2	**0.6**	**0.4**	0
kNN (k=5)	0	**1**	0	0
neural net (4)	0.11	**0.55**	**0.66**	0
TRUE	0.2	0.6	0.4	0

Table III
TWISTING APPROACH RESULTS WITH THE TOY DATASET.

classifier	$P(Y\|F, +)$	$P(B\|F, +)$	$P(Y\|M, +)$	$P(B\|M, +)$
Naive Bayes	0.5	0.5	0.5	0.5
log. regression	0.5	0.5	0.5	0.5
lin. discriminant	0.5	0.5	0.5	0.5
quad. discriminant	0.27	**0.75**	**0.65**	0.06
SVM (rbf)	0	**1**	0	0
regression tree	0.33	**0.75**	**0.63**	0
kNN (k=5)	0	**1**	0	0
neural net (4)	0.25	**0.74**	0.50	0.63
TRUE	0.25	0.75	1	0

input: labeled training data $(\mathbf{X}, \mathbf{a}, \mathbf{y})$.
output: action rule $\mathcal{H}_{context} : X \rightarrow a$.

1) Construct a training set $(\mathbf{X}, \bar{\mathbf{a}})$, where for each training instance i $\bar{a}_i := a_i$ if $y_i =$ '+', otherwise $\bar{a}_i := \neg a_i$.
2) Learn a mapping $\mathcal{H}_{context} : X \rightarrow \bar{a}$.
3) Action $\mathcal{H}_{context} : X \rightarrow a$.

Figure 4. Contextual approach.

- customers responded positively (+) either because they were certain (does not matter, which envelope) or because the *right* color was sent to them;
- customers responded negatively (-) either being certain about their negative attitude (independently of the envelope) or because of the *wrong* color.

Thus to all positive customers (+) we assign the labels according to the original color of envelopes, which were sent to them. To the negative customers (-) we assign the *opposite* colors as it was sent, hoping that another color might have changed their decision.

Using the training set $(\mathbf{X}, \bar{\mathbf{a}})$, constructed as described, we learn a classifier $\mathcal{H}_{context} : X \rightarrow \bar{a}$. The classifier $\mathcal{H}_{context}$ outputs which color to use. The final decision whether to send an offer or not is made using the predictor $\mathcal{L} : (X, a) \rightarrow y$. The approach is summarized in Figure 4.

The approach is limited to binary actionable attributes, but not limited to binary outcomes. It is noteworthy, that by modifying the training set we introduce a sample selection bias in the training process.

Contextual approach has an advantage over the other two. If a learning problem is formulated in this way, the testing labels are in fact available. We again use the toy marketing example for illustration. In Table IV we present simulation results. Here the desired outcome is $P(Y|F) < P(B|F)$ and $P(Y|M) > P(B|M)$, the methods that satisfy it are marked in bold. All the methods except kNN do well.

D. Focusing on sensitive cases

Up to now we discussed the approaches assuming that the contribution of an actionable attribute to the label is uniform for all instances. In fact, this is a simplified scenario. For more complex data the impact of an action can be different for every instance. Possible outcomes of changing action are illustrated in Figure 5.

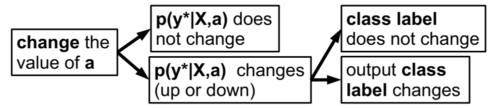


Figure 5. Effects of a change in an actionable attribute (a).

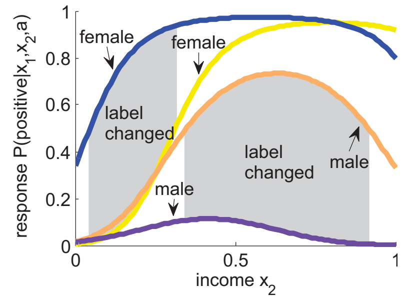


Figure 6. Changes in response on the toy dataset.

To explore this issue, we expand our toy dataset with a continuous (not actionable) attribute. It can be assumed to represent personal income. The data is shown in Table V.

We train a quadratic discriminant classifier using the wrapper approach. The effects within different groups are plotted in Figure 6. Recall from Section II that the probability of positive response was maximized (0.6) when sending only to females only in blue envelopes. Here sending any envelopes for a female with income over 0.25 a marketer could expect the response probability close to one $P(+|F, x_2 > 0.25) \approx 1$. Sending yellow envelopes for males with income around 0.75 also gives higher than 0.6 probability of response $P(+|M, x_2 > 0.75) > 0.6$.

Overall, blue envelopes are still globally better for female, while yellow are better for male as before. However, only in the grey areas the label changes as a result of change in the envelope color. This toy example illustrates a simplified situation. In real data the room for focused learning is to specialize in these cases where the outcome (label) actually changes as a result of different action. That is, to concentrate

Table IV
CONTEXTUAL APPROACH RESULTS WITH THE TOY DATASET.

classifier	$P(Y\|F)$	$P(B\|F)$	$P(Y\|M)$	$P(B\|M)$
Naive Bayes	0.3	**0.7**	**0.7**	0.3
log. regression	0.3	**0.7**	**0.7**	0.3
lin. discriminant	0.3	**0.7**	**0.7**	0.3
quad. discriminant	0.3	**0.7**	**0.7**	0.3
SVM (rbf)	0	**1**	**1**	0
regression tree	0.3	**0.7**	**0.7**	0.3
kNN (k=5)	0	**1**	0	1
neural net (4)	0.17	**0.83**	**0.83**	0.17
TRUE	0.3	0.7	0.7	0.3

Table V
TOY DATASET EXPANDED WITH INCOME ATTRIBUTE.

x_1 gender	x_2 income	a envelope	y response	x_1 gender	x_2 income	a envelope	y response
F	0.00	Y	-	M	0.55	Y	-
F	0.05	Y	-	M	0.65	Y	+
F	0.20	Y	-	M	0.85	Y	+
F	0.30	Y	-	M	0.90	Y	-
F	0.35	Y	+	M	0.95	Y	-
F	0.00	B	-	M	0.60	B	-
F	0.10	B	+	M	0.70	B	-
F	0.15	B	-	M	0.75	B	-
F	0.25	B	+	M	0.80	B	-
F	0.40	B	+	M	1.00	B	-

Table VI
THREE APPROACHES FOR CHOOSING THE ACTION.

name	type	a	y	labels for testing
Wrapper	wrapper	categorical	any	no
Twisting	filter	any	any	no
Contextual	filter	binary	any	yes

learning in the grey areas in the figure.

Table VI presents a summary of the three approaches.

IV. Related work

First we overview actionable knowledge discovery and discovering cause and effect relationship in the data. These are the research domains our work falls into. Then we discuss the techniques related to focused supervised learning, which remotely relates to the specific contribution of our paper. We focus the learning to the boundary instances, for which changing the action has high impact to the label. The related techniques come from different domains, rather than action rule mining. We are not aware of works directly related to the set up we explore in this paper.

A reader is refereed to a survey on mining actionable knowledge [1], which discusses data mining in a light of follow up decision making (actions). The role of domain experts is emphasized in producing action rules from data mining results. The implications of data mining to the actions have been under discussion for more than a decade [2]; however, they focus on different aspects of knowledge discovery process. There actions are not in the data mining rules themselves, follow up actions as a result of discovering an interesting rule are considered. We concentrate on the approaches, which mine action rules directly.

Discovering action rules either require prior extraction of classification rules or are built directly. In our study the Wrapper approach is an example of the first type, while Twist and Contextual are of the second type.

Three mainstream paradigms for discovering action rules can be distinguished in relation to desired outcome: association rules, rough sets and classification (decision trees).

Mining action rules without prior classification is often formulated as the *association rule discovery* problem [3], [4]. The focus is how to form the interestingness measures to discover actionable knowledge: conditional lift (conditioned on the actionable attribute) [4], support and confidence with alternative actions [3]. The latter work also uses cost constraints for action.

A series of works by Ras and colleagues [5], [6] mine action rules using *rough set* techniques. In these works the action rules are the end goal for a decision making, the predictive analytics task is not explicitly associated.

In [7] actionable knowledge is extracted from *decision trees*, which is intuitively close to the Wrapper approach, formulated in our study. In [7] the tree is inspected from a perspective, what attributes should be changed in order to get to the leaf with higher probability of the desired outcome. Instead of assuming actionable and observable attributes, the authors introduce a cost matrix for changing the value of each attribute. Impossible actions are marked with infinite costs (like changing from an adult to a child).

One of the earliest works related to choosing actions [8] builds reasoning about actions from causality perspective.

From focused learning perspective there are remote relations to boosting [9], classification with reject option [10] and evaluating classifier competence [11]. Boosting goes over a loop of training, each step putting more emphasis on learning the cases which were previously misclassified. Classification with reject option evaluates the regions of competence and is allowed not to output the decision if confidence is low.

V. Experiments

The case study dataset comes from MastersPortal.eu. The webportal provides information about Master study programmes in Europe. It was launched in 2007. In March of 2010, MastersPortal.eu received nearly 1000000 monthly visits and contained over 15000 programmes. The revenue model for the MastersPortal.eu website is based on selling web-based advertising campaigns and providing sponsored search-results. Thus, it is vital to increase both the number of visitors and the number of pages viewed by each visitor.

Throughout the existence of the portal, the bounce rate of visitors was in the range of 55-70%. However, visitors referred by Google had a much higher bounce-rate, about 90%. They were referred directly to one of the programmes, and if not interested in that particular programme would go back to Google rather than exploring the portal.

In response to this situation, it was decided to build a recommender system that would help visitors finding relevant information and thus also promote staying longer within the portal. The results of the master thesis [12] showed that with content-based or collaborative recommenders (but not a baseline recommender) it is possible to reduce the bounce-rate for visitors referred by Google from 90% to 82%. We use the data from one of the latest live experiments for exploring the potential of learning with actionable attributes.

The dataset contains information of 39477 *human* visitors coming from Google. The visits were filtered out of 90650 visits recorded over two weeks live testing (March 17 - March 30, 2010). Each visitor was directed into one of the two experimentation groups using a round-robin assignment approach w.r.t. their country of origin. The experiment resulted in 48.3% of visits shown content based recommendations and 51.7% collaborative. The respective positive outcome rates (not bounced) were 17.6% and 18.3%. Content based recommendations performed slightly better, but the difference is not essential.

We use the following attributes for learning: (a) recommender type (collaborative or content based), (x_1) session count (returning visitor or not), (x_2) continent, (x_3) screen resolution, (x_4) browser, (x_5) referrer site, (x_6) country, $(x_{7\ldots12})$ query related attributes, and (x_13) outcome (label). In total there are 14 attributes, from which the first one is actionable (recommender type), 12 are descriptive, and the last one is the outcome, whether the visitor bounced or not (not bounced is the desired outcome). The priors for the label are 17.9% (not bounced) and 82.1% (bounced).

The task is to learn how to choose the optimal action (recommender type), so that the probability of not bouncing is maximized for a given user. Note, that we do not use the information about the overlap in recommendations generated by content-based and collaborative approaches, while they might have provided the same recommendation in certain cases.

The goal of the experiments is to analyze how we can learn to choose the type of recommender (actionable attribute) in order to increase the expected rate of positive outcome. We aim to focus the learning of an action rule to the borderline cases, the ones which are the most sensitive to the optimal choice of the recommender.

A. Evaluation Issues

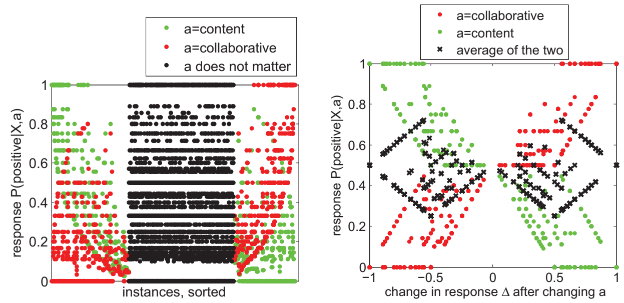
Offline evaluation is tricky, as the ground truth for the alternative action with a given training instance is not available. One of the key aspects of this study is to produce an action rule if the outcome of an alternative action is not known and not feasible to obtain.

Given an unseen instance (X, a, y), where X was defined as a set of observable attributes, a is the actionable attribute and y is the original label obtained in the real online experiment, an action rule $\mathcal{H} : X \rightarrow \hat{a}$ suggests an optimal action for that instance, to maximize the posterior probability of the desired outcome (y^*). Note, that the original y and a of the testing instance are not input to the model. Four situations are possible as a result:

1) $y = y^*$ (positive instance) and $a = \hat{a}$ (the suggested value of the actionable attribute is the same as the original one);
2) $y \neq y^*$ and $a = \hat{a}$;
3) $y = y^*$ and $a \neq \hat{a}$;
4) $y \neq y^*$ and $a \neq \hat{a}$.

The first case can be counted as correct. The other three are challenging. In the second case we do not know if $y = y^*$ is possible at all for the given instance. If it is possible, then the model made a mistake. Otherwise, it is not a mistake. In the third case we do not know if $y = y^*$ is possible with other value of the actionable attribute $(\neg a)$. If it is not possible, then this is a mistake, otherwise not. In the fourth case it is not clear if the suggested action have changed the outcome or not, but it is not a mistake anyway.

To overcome these challenges a live online testing would be needed, which is out of the scope of this study. Thus



(a) all instances (b) sensitive

Figure 7. The effects of changing action.

we concentrate on exploratory analysis of the experimental results to identify the effects of focused learning.

B. Experiments

Experimental evaluation on MastersPortal.eu data consists of two parts. In the first we explore where the boundary cases in the data are. In the second, we compare focused learning with learning using all data.

1) Where are the boundary cases?: We investigate how the change of a recommender type affects the expected probability of not bouncing. We divide the data into training and test sets at random (equal halves). We construct two alternative test sets: one with collaborative recommender $Test_A = (\mathbf{X}\mathbf{a}_{collaborative})$ and another with content based recommender $Test_B = (\mathbf{X}\mathbf{a}_{content})$.

We train the decision rules using $Train$ and analyze the outputs produced on $Test_A$ and $Test_B$ datasets. We use a decision tree with continuous outputs as the base classifier. Note, that the test error on the test set with the original values of the actionable attribute is 24.4% and the area under ROC curve is 0.56, the prediction task itself is difficult.

In Figure 7(a) we plot the posterior probabilities for $Test_A$ and $Test_B$, sorted from the most negative to the most positive difference between the two (Δ). Let $\Delta = P(+|X, a_{collaborative}) - P(+|X, a_{content})$ is the difference in posterior estimate, resulting from a change in the value of the actionable attribute. The black instances in the middle indicate $\Delta = 0$, which means that a change in a makes no change in the posterior. The figure shows that the choice of action makes a difference to the prediction, while not necessarily being close to the *decision* boundary.

In Figure 7(b) we plot only the instances for which the *predicted* class label changes as a result of changing the value of a. We also plot the average of the two posteriors. In total 2218 labels change from the total 19738 test instances as a result of changing a. These 11% represent the sensitive boundary cases to which we aim to focus the training.

From this analysis a conclusion follows. Sensitivity of the output to the actionable attribute is not the same as closeness to the decision boundary. The actionable attribute might give a large impact to the posterior even being far from the decision boundary and vice versus. We want to focus our action rules to the cases when the value actionable attribute makes high impact to the final outcome.

2) Learning focused on the boundary cases: The goal of the second experiment is to compare focused action rule learning with pooled learning using all the training data. We choose to compare the results grouped based on the continents from which visitors are coming.

We divide the data at random into three parts: pre-training, training and testing. For the test set we construct two alternative sets: one with collaborative recommender $Test_A = (\mathbf{Xa}_{collaborative})$ and another with content based recommender $Test_B = (\mathbf{Xa}_{content})$ as in the previous experiment. We use $PreTrain$ set to build a filter to select the instances for *focused* learning. The filter identifies the instances which change the label as a result of change in action, the way it was within the grey areas in Figure 6. We use regression tree as a filter here. In Figure 8 the difference between pooled and focused training is illustrated.

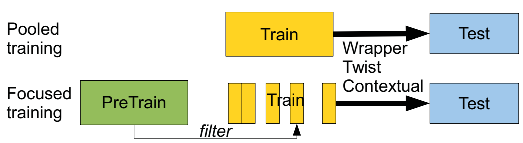


Figure 8. Polled and focused training.

In Figure 9 we analyze the effect of actions to the visitors within different continents. The *focused approach* suggests choosing collaborative recommender for Oceania and South America and content based recommender for the rest. In contrast, *pooled approach* suggests that content based recommender is always better on average (that we could already see from the label statistics without learning). Focused approach demonstrates specialization.

Similarly, we inspect the results of Twist approach (Figure 9) for pooled and focused training. Note, that here the plot shows the posterior of one action. If there was no difference across the continents, we would expect a flat line, parallel to the horisontal axis. Using Twist approach both pooled and focused training give similar results; they make a similar distinction between the continents. Thus, both demonstrate promising performance.

Finally, let us see what happens with the Contextual approach (Figure 9). Here, in the focused approach, the continents differ, while in the full approach the line is almost flat, thus we gain no information about the choice of action. Clearly in this case focused training gives much more specialized results, that is what we aim for.

The experiment demonstrates the potential of the formulated set up and approaches. This comparative experiment shows that focused learning leads to identifying desired actions in a more distinct way, especially with Wrapper and Contextual approaches.

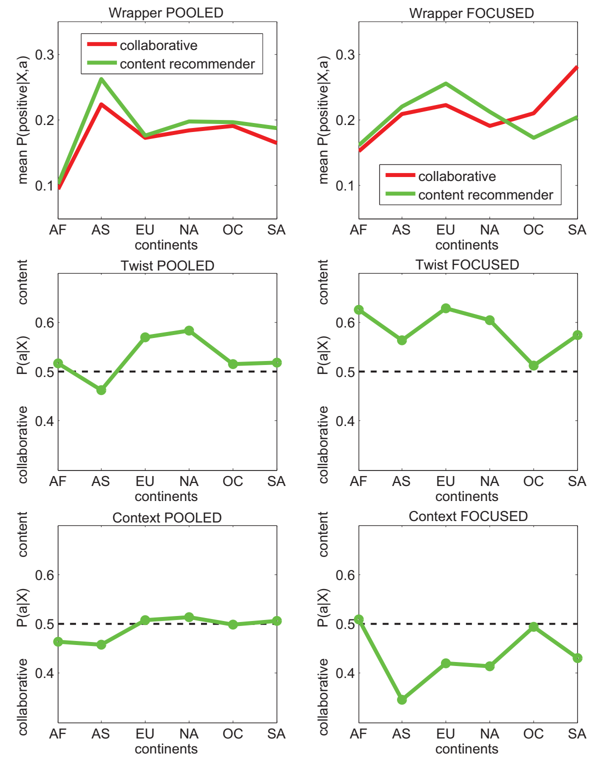


Figure 9. Effects of focused learning.

VI. Discussion and Future Work

This study opens a potential for direct extensions and follow up studies for learning with actionable attributes, which we discuss in this final section.

First, in this work we assumed that there is only one actionable attribute and that it is categorial. Actionable attributes can also be ordered or real valued (e.g. a discount for the next purchase). In these cases a simple enumeration of actions and wrapper like approaches are not appropriate.

Existence of more than one actionable attribute brings more interesting challenges and opportunities. For example, the problem of recommender system itself can be formulated as learning with actionable attributes. Every user is described with a set of observed features (e.g. demographics and already rated items) and actionable attributes, i.e. items that can be recommended for this user. The task can be formulated as selecting the most promising actions such that the user would be interested or would like at least one of the recommendations.

Second, in this work we assumed that the costs associated with each value of the actionable attribute are the same or are zero. In some applications costs can be different for each action. One straightforward approach to account for costs is to integrate them into the definition of the boundary cases. In general, if we deal with more than one actionable attribute, an optimization problem can be formulated: given a budget, decide on optimal strategy (including multiple actions) leading to the highest profit. Another interesting addition is related to the fact that costs of an action is not necessary the same for all the instances (e.g. the price of sending an envelope to remote areas).

Besides costs, additional constrains can be introduced. Some actions might be impossible or not allowed to perform together, while others can be complementary. Impossible or not allowed actions or combination of actions can be simply marked with infinite costs, e.g. giving simultaneously two different medications, which cannot be used together.

Third, in this work we assumed that there is no direct relation between the action and the target (e.g. the color of an envelope does not increase the value or reduces the price of the product). However, if we consider a mortgage application or placing a product 'on sale' in a supermarket the action contributes to the outcome directly. This assumption potentially can be incorporated in the learning process.

Cost-sensitive supervised learning, in which obtaining the value of each attribute for an instance is associated with a particular cost (e.g. the cost of a medical test), may be also seen as close setup. Indeed, a doctor may decide which test to perform to better diagnose a patient. However, the type of a diagnostic test is not expected to cure a patient (i.e. change the label from positive to negative). But the type of treatment might affect negatively. Thus, there are two interesting variations possible with our problem formulation. In the above settings the doctor is willing to make an accurate diagnosis, not (yet) to give the best treatment to save patient's life. And the doctor is not affecting the result of the test, i.e. the doctor decides, which attribute information is important to know. All these variations lead to the formulation of interesting optimization problems, subject to our further research.

VII. Conclusion

The paper defines a set up of focused supervised learning with actionable attributes and further research directions. The problem we formulated is not completely new. It has been studied from different perspectives in different areas of data mining. However, the viewpoint presented in this paper is new. We emphasize 1) supervised learning setup, motivated by predictive analytics domain, 2) maximizing the probability of the desired class rather than learning to separate two classes, and 3) focused learning, motivated by the notion that different instances are sensitive to the value of the sensitive attribute in different ways.

We demonstrated with artificial and real data that an accurate choice of action is essential for those instances, which are on a borderline, and that this borderline is not limited to the decision boundary between positive and negative instances, i.e. it is not sufficient to reduce this supervised learning problem to predicting with a reject option.

We further explored the potential of the focused learning approach with the real-world dataset. Two recommender types, having almost equally good performance globally can be used yet more efficiently in a focused approach, where the learnt model is tailored to the most sensitive cases.

We identified several immediate future research steps. First, we plan to generalize supervised leaning with actionable attributes to continuous attributes, and, second, make the learning process cost-sensitive allowing to utilize this setup also in other scenarios. For example, a marketer can get help in deciding at individual customer level whether to give a discount or send an offer without it.

Acknowledgment

The authors thank Thijs Putman for providing the dataset used in the experimental study.

References

[1] Z. He, X. Xu, and S. Deng, "Data mining for actionable knowledge: A survey," Harbin Institute of Technology China, Technical report, 2005.

[2] G. Adomavicius and A. Tuzhilin, "Discovery of actionable patterns in databases: The action hierarchy approach," in *Proc. of the 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD'97)*, 1997, pp. 111–114.

[3] Z. He, X. Xu, S. Deng, and R. Ma, "Mining action rules from scratch," *Expert Syst. Appl.*, vol. 29, no. 3, pp. 691–699, 2005.

[4] H. Zhang, Y. Zhao, L. Cao, and C. Zhang, "Combined association rule mining," in *Proc. of the 12th Pacific-Asia Conf. on Advances in knowledge discovery and data mining (PAKDD'08)*, 2008, pp. 1069–1074.

[5] Z. W. Ras and A. Wieczorkowska, "Action-rules: How to increase profit of a company," in *Proc. of the 4th European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD'00)*, 2000, pp. 587–592.

[6] Z. W. Ras, E. Wyrzykowska, and L. S. Tsay, "Action rules mining," in *Encyclopedia of Data Warehousing and Mining*, J. Wang, Ed., 2009, pp. 1–5.

[7] Q. Yang, J. Yin, C. Ling, and R. Pan, "Extracting actionable knowledge from decision trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 43–56, 2007.

[8] H. Geffner and J. Wainer, "Modeling action, knowledge and control," in *Proceedings of the European Conf. on Artificial Intelligence (ECAI'98)*, 1998, pp. 532–536.

[9] Y. Freund, "Boosting a weak learning algorithm by majority," *Inform. and Comput.*, vol. 121, no. 2, pp. 256–285, 1995.

[10] P. L. Bartlett and M. H. Wegkamp, "Classification with a reject option using a hinge loss," *Journal of Machine Learning Research*, vol. 9, pp. 1823–1840, 2008.

[11] M. Li and I. K. Sethi, "Confidence-based classifier design," *Pattern Recognition*, vol. 39, no. 7, pp. 1230–1240, 2006.

[12] T. Putman, "Towards adaptive retrieval and recommendation of higher education programmes," Master's thesis, Eindhoven University of Technology, 2010.