Paper

Cross-selling models for telecommunication services

Szymon Jaroszewicz

Abstract—Cross-selling is a strategy of selling new products to a customer who has made other purchases earlier. Except for the obvious profit from extra products sold, it also increases the dependence of the customer on the vendor and therefore reduces churn. This is especially important in the area of telecommunications, characterized by high volatility and low customer loyalty. The paper presents two cross-selling approaches: one based on classifiers and another one based on Bayesian networks constructed based on interesting association rules. Effectiveness of the methods is validated on synthetic test data.

Keywords— cross-selling, telecommunication service, classifier, association rule, Bayesian network.

1. Introduction

The definition of cross-selling (according to Wikipedia) is:

"*Cross-selling is the strategy of selling other products to a customer who has already purchased (or signaled their intention to purchase) a product from the vendor.*"

Cross-selling offers several advantages. Except for the obvious from the extra products sold, it also increases the dependence of the customer on the vendor and therefore reduces churn. We will now discuss some of the specific aspects of cross-selling in the telecommunication industry, with special focus on cellular network operators.

Telecommunications markets are characterized by high volatility. Customer loyalty is at a very low level in this sector, due to anti-monopoly measures taken by governments, as well as lucrative offers for *new* customers from most service providers.
Cross-selling is thus very important for cellular operator since the more services a user has activated the closer he/she is tied to the company, and the harder it is for him/her to switch to another provider.
In case of telecommunication companies, there exist several marketing communication channels through which a customer can be reached:

- offers made to customer when *he/she* contacts the call-center;
- a phone call to the customer;
- an SMS sent to the customer;
- a standard mail sent to customer (may accompany the monthly bill).

It may seem that some of those channels (especially SMS messages) incur almost no cost, so a large number of offers should be sent. In reality this is not true. The reason for that is the negative reaction of customers to too many offers [3]. Too many SMSes are simply annoying, the users quickly learn to ignore them.
It follows that the amount of cross-selling opportunities is in fact quite limited, and the campaigns have to be carefully targeted such that the probability of a "hit" is maximized.

Let us now briefly discuss related literature and available commercial cross-selling solutions.
A cross-selling application applied in the banking sector is presented in [3]. The system selects customers who would potentially be interested in opening a brokerage account. A classifier (decision tree) is built separately for each service. If a customer who does not have a brokerage account falls into a leaf of the tree where many customers have such an account, it is assumed that the customer is likely to accept the offer. The authors claim that the acceptance rate was much higher than for random offers.
In [15] association rules and statistical models are used to predict purchases based on WWW logs. Association rules are used to generate features which are then used as inputs to a hybrid classifier model.
In [10] the authors present a probabilistic model with hidden variables for predicting customer behavior based on their purchases and questionnaire data. An advantage of such models is high flexibility and possibility of inclusion of hidden variables. A disadvantage is the difficulty of detecting relationships not included in the model. In this work this problem has been solved through the use of association patterns to discover new relationships.
Wong and Fu [18] present a method of selecting a subset of services which should be promoted in order to maximize overall profit. The influence of popularity of some services on the popularity of others is taken into account. The analysis of dependencies between products is achieved through market basket analysis (association rules). It has been shown that selecting an optimal set of products in NP-complete, thus an approximate algorithm has been presented.
A number of companies offer cross-selling products, some of them targeted specifically at telecommunication market. We will briefly describe two such products.
Single attachment station (SAS) offers a telecommunication cross-selling solution [14]. Detailed information is not available, however, the company does say that it is based on market basket analysis [2]. Association rules are used to analyze typical paths of customers' development, e.g., be-

ginning with a single phone line and later moving to a few phone lines plus an Internet connection. This allows for identification of customers who can be interested in purchasing new services. The system is custom built by SAS specialists and requires the purchase of SAS licence.
IBM offers `IBM Guided Selling & Active Advisor` a complete cross-selling solution targeted primarily towards retail sales. No description is available of methods and algorithms used.
Related to cross-selling are so called *recommender systems* [1], which offer suggestions to customers based on the similarity of their purchase histories to histories of other customers. Probably the best known example is the webpage of the `www.amazon.com` online bookstore displaying an information "customers who bought this book also bought...". Such systems are dedicated to retail stores with thousands of products. Telecommunication markets are quite different in this respect since the number of services is much smaller. Also, more data about customers such as sex, calling history, etc., are available, which is not the case for recommender systems. Such systems are thus not very useful for cross-selling in the telecommunication industry.

This paper presents an analysis of two approaches to cross-selling in a telecommunications setting.
The first approach is based on constructing a Bayesian network representing customer's behavior and using this network to predict which customers are most likely to pick each service offered. This gives not only a cross-selling model, but also allows the analyst to gain insight into the behavior of customers. The Bayesian network is constructed using a method based on author's previous work. The method starts with a (possibly empty) network representing users background knowledge. At each iteration patterns are found whose probabilities in customer data diverge most from what the network predicts. The analyst then explains those discrepancies by updating the network.
The second approach uses a separate classifier model for each service offered. Each model predicts, which customers are most likely to buy a specific product. Each customer is then offered a service the classifier of which gives the highest probability of acceptance (among the services which the customer does not yet use). The method does not give any insight into customer behavior but is fully automatic.

2. Test data and experimental setting

Unfortunately the author was not able to perform the experiments on real customer data. Instead, a synthetic data generator developed at the National Institute of Telecommunications was used. Efforts have been made to ensure that the simulation is realistic. To ensure objectivity, the data generator was created by a different person then the one doing the experiments. The experiments revealed that the method based on Bayesian networks achieved lower cross-selling accuracy than the classifier based method, but offered valuable insight into customer behavior (e.g., it was able to reconstruct much of the data generator's internal logic). Below we describe the experimental setting and the data generator.
It has been assumed that the cross-selling action targets three optional services allowing the customer to lower connection cost. The services are described below:

- **RL: cheap local calls** lower price for calls made within customer's local area;
- **TPG: cheap late calls** lower price for calls made after 6 p.m.;
- **TPWS: cheap call within the network** lower price for calls to other users of our network.

The goal is to design a system which for a given customer will suggest one of the above services, which the customer is likely to accept.
Data generator. The generator works in three stages. First customer billing data are generated. Based on those data and a set of rules, active services are chosen for each customer. Data is then aggregated to obtain the format used in data warehouses, e.g., one record of the aggregated dataset corresponds to one customer.

Table 1
Characteristics of customer profiles

Profiles	Characteristics
1	More SMS-type services 60–70% of all uses, short connection time – most connections take just few minutes
2	Most connections during peak hours, few connections outside peak hours
3	Many peak hours connections both within the network and to land lines, less evening connections
4	Most calls during peak hours to land lines, 1 or 2 area codes
5	Most calls to just a few selected users, most calls withing the network

To represent customer diversity, the simulated customers have been split into several profiles. Each profile has different calling habits. Table 1 shows brief characteristics of the profiles.
For each customer we also select at random (taking into account customer's profile) one of six calling plans. In general, the longer a customer talks, the higher plan he is assigned (meaning higher monthly payment but lower cost per call).
Based on customer's profile his/her billing data are generated. Data is then aggregated into a data warehouse format, and services used by each customer chosen based on probabilistic rules. Attributes of aggregated data are given in Table 2.

Table 2
Attributes of the aggregated database table (data warehouse)

Attribute	Description
user-id	User identifier
czas-polaczen	Total connection time
ilosc-polaczen	Number of connections made
sr-dlug-pol	Average connection length
pd_*y*-il-pol	Number of connections during part of day *y*
pd_*y*-czas-pol	Total connection time during part of day *y*
usl_*y*-czas-pol	Total connection time for service *y*
usl_*y*-il-pol	Number of connections for service *y*
taryfa	Customer's calling plan
rl	Active "cheap local calls"
tpg	Active "cheap late calls"
tpws	Active "cheap call within the network"

The following types of calls are available:

- SMS,
- connection within the network,
- connection to another cellular network,
- connection to a land line.

Each day is split into the following three "parts of day": 8:00 – 17:59, 18:00 – 23:59, 0:00 – 7:59.
Numerical variables have been discretized using the equal weight method. About five thousand data records have been generated. The data has been split into the training (4000 records) and testing (1000 records) sets. Models are built on the training set, and their accuracy is verified on the test set. This minimizes the risk of overfitting where the model "learns" the training data but cannot generalize to new examples.

3. Association rules based approach to cross-selling

In this section we describe the association rules based approach to cross-selling. Association rules have first been introduced by Rakesh Agrawal and his team [2] and used to analyze supermarket purchase data. Thus the approach is also known as *market basket analysis*.
Initially association rules have been defined for binary tables, where each attribute corresponded to an item and each record to a transaction. The attribute was set to 1 in a given record if the corresponding item was purchased in the corresponding transaction.

Let $H = \{A_1, A_2, \ldots, A_n\}$ be the set of attributes. Take any subset $I = \{A_{i_1}, A_{i_2}, \ldots, A_{i_k}\} \subseteq H$. The *support* of the set of attributes I in a database table D is defined as

$$\text{support}_D(I) = \frac{|\{t \in D : t[I] = (1,1,\ldots,1)\}|}{|D|}, \qquad (1)$$

that is, as the fraction of records in which all attributes in I are simultaneously 1.
If $I, J \subset H$ and $I \cap J = \emptyset$, we can define an *association rule* $I \rightarrow J$. For such a rule we define two quantities which assess its quality: *support* and *confidence*, given by the following formulas:

$$\text{support}_D(I \rightarrow J) = \text{support}_D(I \cup J), \qquad (2)$$

$$\text{confidence}_D(I \rightarrow J) = \frac{\text{support}_D(I \cup J)}{\text{support}_D(I)}. \qquad (3)$$

Support tells us what proportion of transactions in the database contain all items in $I \cup J$, and confidence tells us how likely it is that a transaction containing all items in I also contains all items in J.
In [2] the Apriori algorithm has been presented, which discovers all rules with given minimum support and confidence. Minimum support ensures that discovered rules pertain frequently occurring situations, and minimum confidence ensures high predictive value.
Association rules can easily be generalized to multivalued and numerical (through discretization) attributes.
An advantage of association rules is that existing algorithms allow for finding all rules with given parameters, allowing for discovery of high level correlations. A drawback is that usually too many rules are discovered which creates a secondary analysis problem of finding rules which are interesting to the user. One of such filtering methods (developed by the author) has been applied here to the cross-selling problem.

3.1. Finding interesting association rules

As it has been said above, application of association rules requires methods of selecting interesting rules. One of the methods for achieving this task has been developed by the author of this paper (in cooperation with others) and published in [7, 8].
The method is based on taking into account users knowledge of the analyzed problem. The knowledge is represented using a formal model (Bayesian network). Association rules discovered in data which do not agree with what users knowledge predicts are considered interesting. Such rules are than used by the user to update the model, and the algorithm is applied again to find new interesting rules.
User's knowledge is represented using Bayesian networks [6, 9, 12]. Bayesian networks are directed acyclic graphs depicting direct causal relationships between attributes. Vertices correspond to attributes, and edges to direct causal links. Additionally every vertex is labelled with a conditional probability distribution. A Bayesian network completely determines a joint probability distribution over the attributes it described, allowing for inferences based

on that distribution. Figure 1 shows and example Bayesian network.

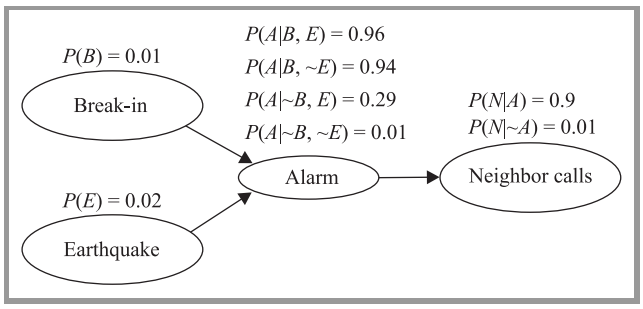

Fig. 1. An example Bayesian network describing simple probabilistic relationships.

One of the main advantages of Bayesian networks is their intelligibility. The dependencies between attributes are shown simply as edges in a graph. Bayesian networks are easy to build, it suffices to connect vertices with appropriate edges. This is usually easy, since humans can easily identify causal relationships [12]. Once the graph has been constructed, conditional probabilities are simply estimated from data. An additional advantage of Bayesian networks is that they determine joint distribution over their attributes, so the description they provide is complete.

Let E be a probabilistic event. The *interestingness* of this event is defined as [8]

$$inter(E) = |P^{BN}(E) - P^{D}(E)|, \tag{4}$$

that is, as the absolute difference between the probability of that event obtained from data and predicted based on the Bayesian network.

Events analyzed in [8] have the form

$$attribute_1 = value_1 \wedge attribute_2 = value_2 \wedge \ldots \wedge attribute_k = value_k, \tag{5}$$

corresponding to sets of attributes in market basket analysis.

The algorithm in [8] finds all such events with given minimum level of interestingness.

One of the main problems related to Bayesian networks is high computational complexity of computing marginal probabilities needed in Eq. (4). Bayesian network inference is NP-complete, and during the course of the algorithm such inference is repeated thousands of times. In Eq. (4) the problem has been addressed by computing larger marginal distributions from the network, and marginalizing several smaller distributions directly from larger ones. This allowed for use of networks of up to 60 attributes. In [7] an approximate, probabilistic algorithm has been given, which works even for huge Bayesian networks, and provides guarantees on the accuracy of discovered patterns.

Detailed description of those algorithms is beyond the scope of this work and can be found in [7, 8].

An important advantage of the approach is that its result is a full probabilistic model, not just a set of rules. The model can then be used for probabilistic inference. Bayesian networks are so flexible, that practically any parameter of the model can be computed from them. This has been used below to estimate the probability of acceptance of a given product by a customer during a cross-selling action.

Adaptations needed for the cross-selling problem. The algorithms described above required certain modifications to work for the given application. Problems occurred when to many edges were directed towards a single node, causing an exponential growth of the conditional probability table associated with the vertex. This caused two types of problems.

The first one was big memory consumption. The second, difficulties in reliable estimation of distribution parameters. The first problem was solved by only keeping nonzero probabilities, the second by using so called Laplace correction to estimate the probabilities. Laplace correction smoothes probability estimates by using a uniform prior distribution.

3.2. Building the Bayesian network

We will now describe the process of building the Bayesian network based on the training set.

Before the first application of the algorithm, edges corresponding to trivial, well known dependencies have been added to the network. These were primarily the consequence of how the attributes were aggregated. Table 3 shows edges in the initial network.

Table 3
Edges corresponding to trivial apriori known dependencies following from the way the data were aggregated

From	To	Justification
pd1-il-pol	ilosc-polaczen	Number of connections is the sum over all parts of day
pd2-il-pol	ilosc-polaczen	
pd3-il-pol	ilosc-polaczen	
usl1-il-pol	ilosc-polaczen	Number of connections is the sum over all services
usl2-il-pol	ilosc-polaczen	
usl3-il-pol	ilosc-polaczen	
usl4-il-pol	ilosc-polaczen	
pd1-czas-pol	czas-polaczen	Total connection time is the sum over all parts of day
pd2-czas-pol	czas-polaczen	
pd3-czas-pol	czas-polaczen	
usl2-czas-pol	czas-polaczen	Total connection time is the sum over all services
usl3-czas-pol	czas-polaczen	
usl4-czas-pol	czas-polaczen	
usl2-il-pol	usl2-czas-pol	Number of connections influences connection time
usl3-il-pol	usl3-czas-pol	
usl4-il-pol	usl4-czas-pol	
pd1-il-pol	pd1-czas-pol	
pd2-il-pol	pd2-czas-pol	
pd3-il-pol	pd3-czas-pol	
ilosc-polaczen	sr-dlug-pol	Average length is computed from total time and number of connections
czas-polaczen	sr-dlug-pol	

Table 4

Results of repeated application of interesting association rule discovery algorithm to the cross-selling problem

Most interesting events					
attributes	values	inter.	P^{BN}	P^{D}	conclusions
First application of the algorithm					
ilosc-polaczen, rl, tpg, tpws	2,N,N,N	0.200	0.1839	0.3845	Number of connections influences additional services used by customers. Customers who make few calls don't use those services. **Added edges** from ilosc-polaczen to rl, tpg, tpws
Second application of the algorithm					
pd2-czas-pol, taryfa, rl, tpg, tpws	1,2,N,N,N	0.179	0.0362	0.2153	Relation between those services seems intuitive. In order to better understand the nature of those relationships, most interesting pairs of attributes were examined
sr-dlug-pol, rl	4,N	0.153	0.2215	0.068	Customers making long calls more ofter use the "cheap local calls" service. The conclusion was considered plausible and **edge has been added** from sr-dlug-pol to rl
sr-dlug-pol, rl	3,N	0.140	0.2077	0.3478	
Third application of the algorithm					
pd2-czas-pol, taryfa, rl, tpg, tpws	1,2,N,N,N	0.18011	0.0351	0.2153	The pattern was still the most interesting one, pairs of attributes were examined again
usl4-il-pol, rl	2,T	0.15	0.1680	0.0183	The influence of the number of calls to land lines on "cheap local calls" is plausible. **Added edge** from usl4-il-pol to rl
taryfa, pd2-czas-pol	1,2	0.1425	0.0727	0.2153	Dependency between calling time during the day and calling plan. **Added edge** pd2-czas-pol to taryfa
Fourth application of the algorithm					
taryfa, rl, tpg, tpws	1,N,N,N	0.151	0.1549	0.3058	Calling plan influences services used. Customers with a cheap plan use their phone infrequently, and thus don't activate extra services. **Added edges** from taryfa to rl, tpg and tpws
Fifth application of the algorithm					
sr-dlug-pol, tpg, tpws	4,N,N	0.149	0.3134	0.1648	Customers making long calls usually have at least one of tpg or tpws active. **Added edges** from sr-dlug-pol to tpg and tpws
Sixth application of the algorithm					
ilosc-polaczen, pd2-il-pol, rl, tpg, tpws	2,2,N,N,N	0.167	0.1068	0.2738	Day time 2 means day time connections so it tells a lot about customer's profile. **Added edges** from pd2-il-pol to rl, tpg and tpws

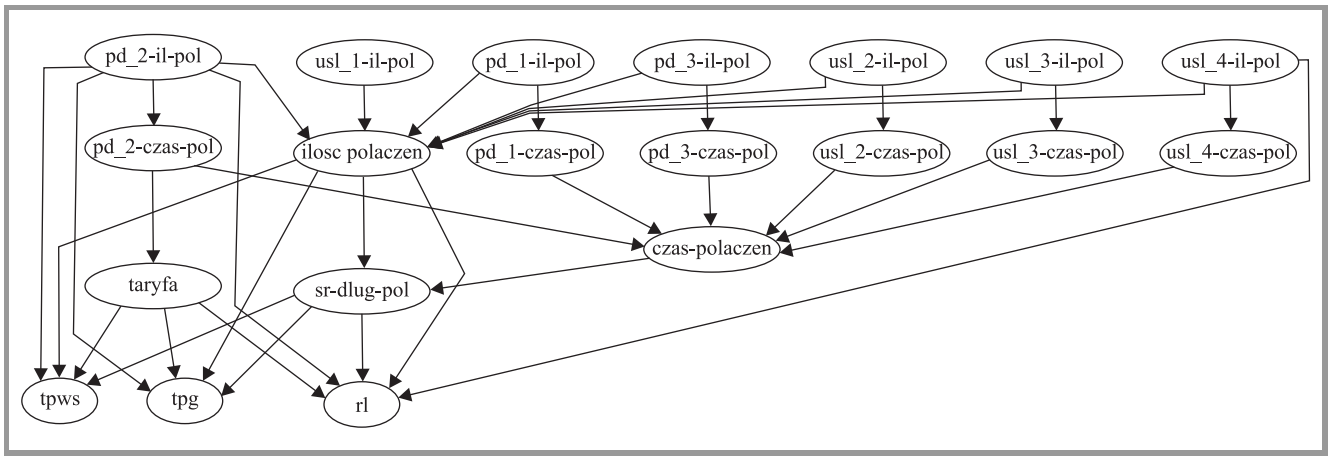

Fig. 2. Final Bayesian network built by analyzing customer behavior data.

Note that the Bayesian network models those dependencies well. For example, connection times in various parts of day are independent of each other. But when the total connection time is known, they become dependent, just as the network predicts.

Table 4 illustrates the process of building the Bayesian network describing customer behavior.

Each subtable shows a new run of the algorithm and the most interesting (in the sense described above) events discovered. The events are conjunctions given in Eq. (5).

The columns of Table 4 are described below:

- attributes: attributes of the interesting event,
- values: values of attributes in the event,
- inter.: interestingness value (Eq. (4)),
- P^{BN}: probability of the event in the Bayesian network,
- P^{D}: probability of the event in the data,
- conclusions: interpretation and explanation of the event, modifications applied to the Bayesian network.

The final Bayesian network is shown in Fig. 2.

3.3. Testing of the model

Reliable testing of a cross-sell solution in off-line conditions is difficult. A real test should involve sending offers based on the analyzed model to a test group of customers, and checking how many of them responded. The procedure should be repeated for another group with random offers. The results for both groups should then be compared.

Such a test was not possible in this work. A simulation of such a test has thus been conducted. We assume that the set of services of a user in the dataset consists of services the user would activate had they been offered (let us call this set A).

For each customer a random subset of those services has been removed (every service was removed with probability 50%). Thus obtained set B was assumed to be the set of services which were active before the marketing campaign. Thus the services in $A \setminus B$ were the services the user would accept if offered.

Then, for each user, based on the Bayesian network the probability of each service not in B was computed, and the offer was made for the service with highest such probability. If the offered service was in $A \setminus B$, the offer was assumed to be accepted. For comparison we also picked a random offer (from those not in B).

The percentage of accepted offers is:

- Bayesian network: 22.84%,
- random offer: 12.83%.

It can be seen that the Bayesian network achieved almost twice as high efficiency as random offers. It should also be noted that in the test set 53.59% of customers did not have any active offer, which in our test prevented them from accepting *any* offer. Since over 50% of offers must have been rejected anyway, 23% accuracy should be considered very high.

4. Classifier based cross-selling approach

In this section we present the second approach which is based on classification models. For each service we want to sell, a classifier is built which assesses the probability that a given customer uses the service. To select which service to offer to a customer we feed his/her data to each of the classifiers and pick the one with highest predicted probability (out of the offers the user does not already have).

As it was mentioned above, this is not an optimal solution. Potential new customers may not resemble current users of

a service. Ideally one should send a pilot offer to a random sample of customers and build classifiers based on the results of that offer. In the current work (and in many real life marketing campaigns) such an approach was not possible.
We have used four classification algorithms implemented in the Weka package [17]: naive Bayesian classifier, decision trees (J4.8), boosted decision trees (AdaBoostM1) and support vector machines. The algorithms have been briefly characterized below, full description is beyond the scope of this work and can be found, e.g., in [17].

Naive Bayesian classifier. Despite being one of the simplest classifier models, this approach often gives results comparable to or even better than other more complicated models [11, 17].
Suppose we want to predict class Y based on attributes $X_1, X_2, \ldots, X_n$. From Bayes theorem we have

$$P(Y = y_i | X_1 = x_{i_1} \wedge \ldots \wedge X_n = x_{i_n}) = \frac{P(X_1 = x_{i_1} \wedge \ldots \wedge X_n = x_{i_n} | Y = y_i) \cdot P(Y = y_i)}{P(X_1 = x_{i_1} \wedge \ldots \wedge X_n = x_{i_n})}. \quad (6)$$

Note that the denominator can be omitted since the probabilities over all y have to add up to one, and we can just rescale the probabilities after classification.
We then use the so called "naive assumption" which says that $X_1, \ldots, X_n$ are independent conditioned on Y, which gives

$$P(Y = y_i | X_1 = x_{i_1} \wedge \ldots \wedge X_n = x_{i_n}) \propto P(X_1 = x_{i_1} | Y = y_i) \cdots P(X_n = x_{i_n} | Y = y_i) \cdot P(Y = y_i). \quad (7)$$

If continuous variables are present, discretization or kernel estimation of conditional distributions is used [11, 17].

Decision trees. Another frequently used classifier model is a *decision tree*. Decision trees [13, 17] are a graphical representation of a decision taking algorithm.
We begin at the root of the tree. In every node we perform an appropriate test and based on its outcome pick the left or right branch of the tree. The procedure is repeated recursively until we reach a leaf of the tree which contains the final decision.
Several decision tree learning algorithms are available in literature [13, 17]. In general the algorithms proceed by picking the test to be placed in the root of the tree, then splitting the dataset in two parts based on the outcome of the test, and then repeating the procedure recursively on each part. After that, the tree is "pruned" to prevent overfitting.
We used the J4.8 algorithm which an improved version of Ross Quinlan's C4.5 method [13].

Boosting. *Boosting* is a method of improving accuracy of other classification models [5, 17]. The idea is based on the fact that classifiers' error can be decomposed into *bias* and *variance* parts. Bias represents classifiers inability to represent complex relationships in data, and variance represents inaccuracies in estimating classifier parameters. In general the lower the bias of an algorithm, the higher its variance.
There exist methods to decrease variance of classifiers. *Bagging* takes several (even hundreds) samples from the training set and builds a classifier on each of them. All those models are then averaged which results in variance reduction.
A better method of variance reduction, which also has the potential to reduce bias is *boosting* [5]. The method works by training a classifier and then reweighting the training set, such that misclassified examples are given higher weights. A new classifier is built on the reweighted data. The process is repeated several times, and all resulting classifiers participate in the final decision. Detailed analysis of the method can be found in [5].
In this work the AdaBoostM1 [5, 17] algorithm was used with J4.8 tree as the base classifier.

Support vector machines. The last classification method is the newest of all four. Support vector machines [4, 16] allow for classification of nonlinear problems while providing guarantees on generalization accuracy for previously unseen cases.
Linear support vector machines construct a linear hyperplane separating both classes. It is constructed in such a way that a large margin between the separating plane and examples from both classes is maintained, which allows for a theoretical guarantee on generalization accuracy.
In order to classify nonlinear problems, original coordinates are transformed in a nonlinear fashion. In the new space the problem may become linear. In order to achieve high efficiency, the transformation is not done explicitly, but achieved through the use of appropriate kernels; see [4] for an excellent introduction.

Experimental results. The classifier based method has been tested the same way as the Bayesian network based model. From among the services the user does not have, we select the one whose classifier gives the highest probability. Effectiveness is estimated exactly as in the previous section. The percentage of accepted offers is:

- naive Bayes: 20.54%,
- decision tree (J4.8): 27.93%,
- AdaBoostM1 (J4.8): 26.19%,
- support vector machine: 28.15%,
- random offer: 12.83%.

It can be seen the naive Bayesian classifier, the simplest classification method, gave results significantly worse than other methods. The remaining three classifiers achieved comparable accuracy, although decision trees have been slightly worse than boosted decision trees and support vector machines.
Almost 30% of offers have been accepted, which means very high effectiveness.

5. Conclusions and further research

It is apparent that classifier based methods achieved (except for the naive Bayesian classifier) higher effectiveness than the Bayesian network.

It should be noted however that such methods do not provide models which are understandable to humans. This is the case even for decision trees, where large size of the tree and potential variable correlations make it difficult to understand the underlying causal structure.

Classifier models can thus be useful for selecting customers and services which should be targeted, but not to explain *why* particular customers prefer particular services. Such knowledge could of course result in a better marketing campaign.

The Bayesian networks based method offers lower accuracy but gives full insight into dependencies between attributes in the data. While building the network we "learn" the data, and eventually get a model describing not just the correlations, but also causal relationships between all variables. We can thus understand how changing one of the parameters will influence probability distributions of other parameters.

The first direction of future research will be improving the Bayesian network implementation such that conditional probability distributions can be represented using classifiers. This should allow the Bayesian network method to achieve accuracy comparable with classifier based methods.

In a longer perspective it would be interesting to create a model which would describe general aspects of customer behavior. It would thus become possible to predict the demand for a service before it was even rolled out to the market. A Bayesian network could form a basis of such model. It would also be useful to couple such a model with customer's lifetime value prediction module.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions", *IEEE Trans. Knowl. Data Eng. (TKDE)*, vol. 17, no. 6, pp. 734–749, 2005.

[2] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases", in *ACM SIGMOD Conf. Manage. Data*, Washington, USA, 1993, pp. 207–216.

[3] M. Berry and G. Linoff, *Mastering Data Mining*. New York: Wiley, 2000.

[4] N. Christianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge: Cambridge University Press, 2003.

[5] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting", Tech. Rep., Dept. of Statistics, Stanford University, 1998.

[6] D. Heckerman, "A tutorial on learning with Bayesian networks", Tech. Rep. MSR-TR-95-06, Microsoft Research, Redmond, 1995.

[7] S. Jaroszewicz and T. Scheffer, "Fast discovery of unexpected patterns in data, relative to a Bayesian network", in *11th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. KDD 2005*, Chicago, USA, 2005, pp. 118–127.

[8] S. Jaroszewicz and D. Simovici, "Interestingness of frequent itemsets using Bayesian networks as background knowledge", in *10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. KDD 2004*, Seattle, USA, 2004, pp. 178–186.

[9] F. V. Jensen, *Bayesian Networks and Decision Graphs*. New York: Springer, 2001.

[10] W. Kamakura, M. Wedel, F. de Rosa, and J. Mazzon, "Cross-selling through database marketing: a mixed data factor analyzer for data augmentation and prediction", *Int. J. Res. Market.*, vol. 20, pp. 45–65, 2003.

[11] T. Mitchell, *Machine Learning*. New York: McGraw Hill, 1997.

[12] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. Los Altos: Morgan Kaufmann, 1998.

[13] J. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo: Morgan Kaufmann, 1993.

[14] "Sas cross-sell and up-sell for telecommunications", http://www.sas.com/industry/telco/sell/brochure.pdf

[15] E. Suh, S. Lim, H. Hwang, and S. Kim, "A prediction model for the purchase probability of anonymous customers to support real time marketing: a case study", *Expert Syst. Appl.*, vol. 27, pp. 245–255, 2004.

[16] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[17] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo: Morgan Kaufmann, 2005.

[18] R. C.-W. Wong and A. W.-C. Fu, "ISM: item selection for marketing with cross-selling considerations", in *Eights Pacific-Asia Conf. Knowl. Discov. Data Min. PAKDD*, Sydney, Australia, 2004, pp. 431–440.

Szymon Jaroszewicz received his M.Sc. degree in 1998 from Szczecin University of Technology, Poland, and his Ph.D. degree in computer science in 2003 from the University of Massachusetts at Boston, USA. He is currently an Assistant Professor at the National Institute of Telecommunications in Warsaw, Poland. His research interests include data mining and knowledge discovery, primarily discovering interesting patterns in large databases.

e-mail: s.jaroszewicz@itl.waw.pl
National Institute of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland