Modeling the Behavior of Bank Clients

Achim Lewandowski and Peter Protzel

Chemnitz University of Technology
Dept. of Electrical Engineering and Information Technology
Institute for Automation
09107 Chemnitz, Germany
achim.lewandowski@e-technik.tu-chemnitz.de
peter.protzel@e-technik.tu-chemnitz.de

Abstract. One goal of Customers Relationship Management is to recognize non-active bank clients which possibly could switch to another bank. We developed an approach to judge the state of a bank client, so that all clients can be addressed with suitable actions. The data was provided by the Eunite 2002 competition. Given the data of 12000 clients with known state (active or non-active), the task was to predict the states of another 12000 clients.

We use neural nets to model the probability of a client to be active. As 36 inputs were given and the data sets were quite large, we used a procedure based on the Naive Bayes Classifier to reduce the number of inputs.

The data was given in a form, that was not directly applicable for fitting neural nets. Most time was used to recode the variables to make the learning phase easier.

1 Introduction

The main task of the Eunite 2002 competition is to model the Customer Intelligence in a Bank. Banks need to analyze the client behavior to recognize a possible tendency of a client to switch to another bank.

In this framework the customer is described by a 36-dimensional vector. 6 of the inputs are nominal and 30 real. The inputs were transformed and the real meaning of the inputs was unknown to the participants of the competition.

The bank provided the data of 24 000 clients. 12 000 of them are marked as "active" and 12 000 are marked as "non-active" clients, but only for the 12000 persons which are in the training set the true values are known. As we have 6000 active and therefore 6000 non-active clients in the training set, we can already conclude that we will have also 6000 active and 6000 non-active clients in the test set.

Before we could fit neural nets to model the probability of a client to be "active", we examined, whether the data was already in a suitable form.

2 Data analysis

We first investigated the structure of each input. The original inputs are not known, because each input had been given transformed, with a new minimum of 0 and a new maximum of 1. Unfortunately the data is not evenly distributed for all inputs. For example, the distribution of input 11 is heavily skewed, with a empirical mean of 0.0006 and nearly 40 percent of all data points are smaller than 0.0001.

For every variable we calculated the number of different values and also the number of different values we need to cover 99 percent of all clients. Furthermore we calculated the number of cases in the test set with values smaller (larger) than the minimum (maximum) values of the training set to judge whether extrapolation could be a problem. We suspect, that training and test set are *not* randomly drawn out of the same set. For the variables 11 to 36 we tested the hypothesis, that the distribution functions are the same by the aid of the Kolmogoroff-Smirnov-Test. For the other variables we used a Chi-Square-Test (collapsing some cells with small expected values). We used a significance level of $\alpha = 0.01$.

V	#	#0.99	#Test<min(Tr)	#Test>max(Tr)	Train=Test?
1	2	2	0	0	
2	8	2	0	0	no
3	54	25	3	0	
4	20	6	4	0	
5	66	8	15	0	
6	29	2	15	0	(no)
7	14	5	0	1	no
8	12	3	0	1	no
9	9	2	0	0	
10	9	3	0	3	
11	157	25	0	0	
12	1597	1357	0	4	
13	179	48	0	0	no
14	1669	1429	0	5	no
15	909	669	0	0	no
16	7018	6778	3	1	no
17	384	144	0	1	
18	2545	2305	0	3	
19	98	17	1	0	
20	2049	1809	0	3	no
21	128	37	0	0	no
22	790	550	2	3	no
23	714	474	2	0	no
24	1177	937	1	0	no
25	105	20	0	0	
26	669	429	4	1	

27	140	40	2	0	no
28	722	482	5	1	no
29	773	533	2	0	no
30	1087	847	4	0	no
31	183	45	1	0	
32	1538	1298	3	3	no
33	223	89	3	0	no
34	1648	1408	3	5	no
35	793	553	1	1	
36	1072	832	3	2	

It seems, that training and test set are not generated by the same law. For example, the empirical distribution function of variable 16 has shifted to the right (see figure 1). For better illustration, we have used the ranks instead of the original values of variable 16.

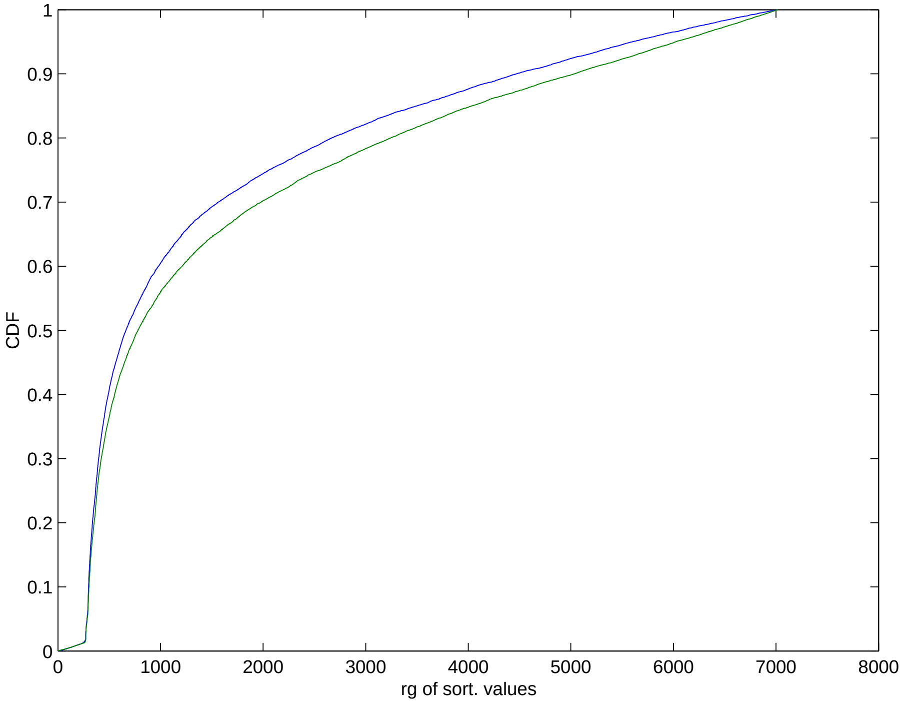

Fig. 1. Example V16: Empirical distribution functions of variable 16 for the training set (above, blue) and the test set (green).

Another example is variable 23. In the following table we show the most frequent values of the original variable 23. Obviously, the differences, especially for the three cases $541 - 543$, cannot be treated as random.

Rank	V23	#Train	#Test
538	0.924029745	358	289
539	0.924030152	260	311
540	0.924030559	335	426
541	0.924030965	3823	3491
542	0.924031372	383	781
543	0.924031779	510	222
544	0.924032186	210	133
545	0.924032593	177	147

If we decide, for example, to declare clients with $V23 = 0.924041779$ as ”active”, we have selected 510 clients in the training set but, only 222 clients in the test set. As we have been told, that there also 6000 ”active” people in the test set, the question would arise, where to find the ”missing” 288 clients. For this ”artificial” example, there are two possibilities: Maybe clients with $V23 = 0.924041779$ are indeed the ”active” clients, but now (for the test set) the distribution has changed and we find only 222 clients with this value, but then we have problems to fill up the missing 288. Or $V23 = 0.924041779$ was just a label (maybe the age of people) and this label has changed, so that we don't know, where to find the new label. Here, a lot of uncertainty arises, as we have no possibility to decide between these two cases.

To make things hopefully clearer, let us assume, that the 12000 cases in the training set are from 2001 (randomly chosen) and the test set data is from 2002 (a realistic setting, because continuous learning was always emphasized). If we discover, based on an analysis of the training set, that all people with an income of more than 1000 are ”active”, and we assume, that income rises every year by 5 percent, we don't know exactly in 2002, whether we should treat people with an income of more than 1000 or 1050 as ”active”. The empirical distribution function of ”income” has shifted to the right. In 2002, there is no ”mathematical” decision between these two cases.

As there are also large gaps between values, we decided to recode most inputs. We left the first 6 nominal inputs unchanged. For all other inputs we calculated the median of the distances between the 24000 sorted values. The sorted values of every input were grouped. We always collected the values from left to right, until a threshold of 200 clients was exceeded.

If there was an original value which was shared by more than 200 clients, a new group was also created. As sometimes large gaps between values appear, we also started a new group, when the original gap was larger than 100000 times the median of the distances. The new input values are equally spaced, and most of the transformed values are shared now by more than 200 clients.

For example, V13 has after the transformation (we omitted a intermediate step, where we reversed the order, only a side effect of an easier way to compute

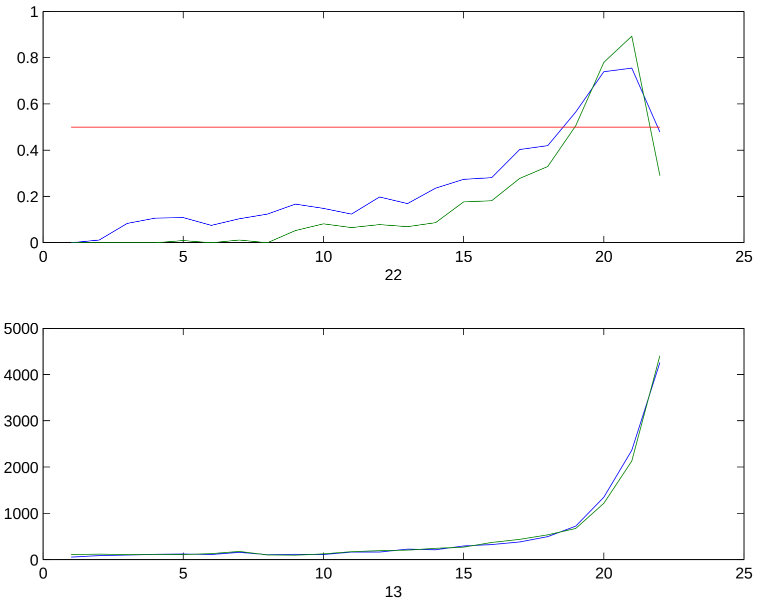

Fig. 2. Example V13: True percentages of active clients in the training set (blue) and estimated percentages (green) in the test set of an early model (above), frequencies in training and test set (bottom)

the transformed values) only 22 different values, whereby the minimal frequency is still 160 for the most left value. Fig. 2 shows the percentages of ”active clients” for the different values of V13, the true values for the training set and the fitted (of an early model) for the test set. The frequencies in the training and test set (bottom) are only roughly similar.

3 Variable selection

To get a feeling for the important variables, we performed a step wise procedure based on the Naive Bayes Classifier. In this contest, the ”active” and ”non-active” clients have the same a-priori probabilities. Therefore, in the Naive Bayes Classifier context with attributes $V_1, \ldots, V_k$, we choose for a given input vector $(v_1, \ldots, v_k)$ the class $i \in \{0, 1\}$ with the higher estimated probability

$$P_i = \prod P(V_t = v_t | C = i)$$

We started a forward selection, beginning with an empty set, and added always that variable, which performed best on the whole training set. As a measure, we used the number of correct classified clients. When the progress was small or when there was no progress at all, we investigated, whether we could remove an older variable. Furthermore, the last column shows, whether the empirical distributions of training and test set values differ significantly for the chosen variable. The variables were added (and removed) in the following way

Step Nr.	Added	correct cases	distr. changed?
1	13	7570	yes
2	25	8463	no
3	24	8539	yes
4	11	8603	no
5	4	8613	no
6	31	8614	no
7	28	8686	yes
8	20	8715	yes
9	8	8757	yes
10	12	8806	no
11	16	8845	yes
12	29	8873	yes
13	7	8882	yes
14	-4	8874	no
15	3	8887	no
16	2	8884	yes
17	6	8891	yes

At this point, we did not achieve any better result by adding or removing variables. A backward selection, starting with all variables, yielded worse results. Of course, there can exist much better combinations, but due to the limited time, from now on we used only the set $\{13, 25, 24, 11, 31, 28, 20, 8, 12, 16, 29, 7, 3, 2, 6\}$ as possible inputs for our neural nets.

Additionally, we tried some trees (axis parallel) to see, which variables were chosen. Here, the variables $V = \{13, 28, 25, 24, 20, 23, 33\}$ were chosen. As the results were slightly worse than the Naive Bayes Classifier results, we kept the other set of inputs.

4 Neural Nets

Why do we use neural nets at all? We divided the training set randomly in two parts of size 6000, fitted a neural net and classified the other part. The whole procedure was repeated several times. With the Naive Bayes Classifier, the number of correct cases was between 4100 and 4300, for the neural nets this number was between 4300 and 4550.

So finally we used a Softmax approach to predict not only the class but also the probability that a client may be active.

After a few experiments, we used finally 6 neurons in the hidden layer and the first 6 variables of our list, ($\{13, 25, 24, 11, 31, 28\}$). In several runs with randomly selected 6000 clients in the training set, this seemed to be a reasonable choice, as there were no outliers for the number of correct classified clients for the remaining 6000 cases. Maybe even more hidden neurons but also lesser hidden neurons would have been possible, but due to the limited time we did not try many different values.

To reduce the possibility of over-fitting, we trained 14 neural nets with randomly generated initial weights, calculated for all 14 models the probabilities for the test set. We declared a customer as active when the mean of the probabilities was larger than 0.5. On the training set, 5409 people were declared as ”active”. For the test set, the number of the people, which were declared as ”active”, was only 4840 and we have a difference of 1160 to the 6000 which can be found in the test set, and therefore we cannot have more than 9680 correct classified cases. This is caused by the dilemma described earlier. The input distribution has probably changed and at those regions, which we have declared as ”active”, we found lesser clients. We could argue, as we don't know where the ”active” clients have moved to, to declare all people with the next smaller probabilities as ”active”. With a threshold of roughly 0.4 we have exactly 6000 ”active” clients. But this is a solution, where we assume, that the active people have moved to ”all directions”. We have observed, that some distributions have been shifted. Maybe the belonging probabilities have moved in the same way. But these are assumptions, which we cannot prove. In our case, we have to look mainly at the variables $\{13, 24, 28\}$.

`V13: most frequent values`

V13	#Train	#active	#a/#Tr	#Test
17	380	153	0.40	436
18	496	208	0.42	534
19	721	407	0.56	671
20	1348	996	0.74	1214
21	2361	1782	0.75	2128
22	4260	2037	0.48	4406

If $V23$ was the only used variable, we would declare the (transformed) values $19, 20, 21$ as ”active”. These are 4430 cases. The test set has only 4013 cases for these values. But we are not sure, that the missing ”active” clients can be found now in the set of ”$V13 = 22$-persons, which had the next highest probability before.

For $V24$ the ”active” tails are less frequent in the test set, but most changes in the distribution occurred for values with a smaller percentage of ”active” clients. But there is also a large descent from 337 (including 164 ”active” clients) to 143 cases.

`V24: changes near the most frequent value`

Value	#Train	#act/#Tr	#Test
37	229	0.432	207
38	491	0.411	657

39	1876	0.370	1945
40	256	0.477	437
41	218	0.477	165
42	337	0.487	143

Finally, we investigated $V28$ around the interesting values:

V28: changes near the most frequent value

Value	#Train	#act/#Tr	#Test
15	198	0.672	177
16	309	0.696	268
17	279	0.584	247
18	292	0.534	263
19	386	0.523	380
20	477	0.459	496
21	978	0.509	922
22	4756	0.451	5055
23	390	0.367	433
24	279	0.308	253
25	195	0.431	193
26	161	0.354	171

As before the number of people, who would be declared as ”non-active” has risen. As $V23$ was the most important variable for the training set, we don’t think, that is a good idea to use only variables whose distributions have not changed significantly. It would have been *much* easier to classify the test set cases with some information about time to recognize trends.

Additionally, we do not know, whether the variables were coded in a way, that neighboring values mean some relationship. Therefore we left our forecasts unchanged, although only 4840 people have been declared as ”active”.

5 Update of our model

Unfortunately, the time is not explicitly given as an input. We could have used the time as an additional input.

An alternative way is to use always the same number of cases, whereby older cases leave the database. When we store the weights of our nets, it is also possible to start training with the stored weights, assuming, that no huge change in the customer behavior has occurred. The number of cases depends on the speed of changes. The slower the changes, the larger the number of used cases is allowed to be. But in this contest, we have no idea, how many cases should be used for training.

Our experience is, that retraining with only a few of the recent examples often causes chaotic behavior. We prefer the already mentioned strategy with a quite broad window.

6 Summary

We used neural nets with the softmax approach to model the probability that a given client is ”active”. Assuming, that our model performs well, this strategy has the advantage, that we can express our belief, for example a value of 0.99 means ”quite sure active” but a value of 0.51 means ”we are not sure, but the client is more active than non-active”. Of course we must rely on the assumption, that the model fits well.

We used a Naive Bayes Classifier for a quick variable selection. As at least five of the first seven variables were also chosen by a tree approach, performed with the original inputs, the variable selection seems to be reasonable.

Most work and time was put into the data exploration. It seems that training and test sets are not generated by the same law. It would have been easier, if we knew the times when the observations have occurred. The heavily skewed distributions of most inputs made things not easier.

Nevertheless, we enjoyed the competition and would like to thank the organizers.