

KDD Cup 2009 @ Budapest: feature partitioning and boosting

Miklós Kurucz Dávid Siklósi István Bíró Péter Csizsek
Zsolt Fekete Róbert Iwatt Tamás Kiss Adrienn Szabó

{MKURUCZ, SDAVID, IBIRO, CSIZSEK, ZSFEKETE, RIWATT, KISSTOM, ASZABO}@ILAB.SZTAKI.HU

Computer and Automation Research Institute of the Hungarian Academy of Sciences

Editor: Gideon Dror, Marc Boullé, Isabelle Guyon, Vincent Lemaire, David Vogel

Abstract

We describe the method used in our final submission to KDD Cup 2009 as well as a selection of promising directions that are generally believed to work well but did not justify our expectations. Our final method consists of a combination of a LogitBoost and an ADTree classifier with a feature selection method that, as shaped by the experiments we have conducted, have turned out to be very different from those described in some well-cited surveys. Some methods that failed include distance, information and dependence measures for feature selection as well as combination of classifiers over a partitioned feature set; in addition LogitBoost outperformed most classifiers for most feature subsets of the KDD Cup 2009 data.

Keywords: Feature selection, classifier combination, LogitBoost, Alternating Decision Trees.

1. Introduction

The KDD Cup 2009 task targeted for the propensity of customers to switch provider (churn), buy new products or services (appetency), or buy upgrades or add-ons proposed to them to make the sale more profitable (up-selling). For 50,000 anonymous customers a small data set of 230 and a large of 15,000 features was provided; in this paper we describe our various successful and failed attempts mostly over the large data set.

Telephone customer behavior analysis appears less frequently in publications of the data mining community. Some exceptions include machine learning methods for churn prediction on real data (evolutionary algorithm, Au et al. (2003); classifier combination by linear regression, Wei and Chiu (2002); Naive Bayes, Nath and Behara (2003); graph stacking, Csalogány et al. (2007)). The area is explored in more depth in marketing research including small sample survey results of Kim and Yoon (2004) and rule extraction and behavior understanding over a small 21-feature data by Ultsch (2002). Of closest interest, Neslin et al. (2006) present the overview of a churn classification tournament very similar to the KDD Cup 2009 task but with emphasis also on managerial meaningfulness and model staying power.

The difference in the KDD Cup 2009 large data set compared to typical classification problems is the abundance of features. Due to their large number we had prepared feature selection and partitioning methods before the training label release. By the lessons we have

learned from Web Spam Challenges (Siklósi et al. (2008)) we had expected that feature partitioning and classifier combination would perform better than global classifiers.

Due to the large number of features it was also clear that feature selection methods are required. Our best performing methods have turned out to be very different from those described in some well-cited surveys as e.g. by Dash and Liu (1997): feature evaluation could only be used for a weak pre-selection while wrapper methods failed due to slow convergence and overfitting. Our final feature selection method is the union of the best features selected by LogitBoost of Friedman et al. (2000) over the feature partition that we have originally devised for classifier combination.

Our classifier implementation choice was mostly determined by the possibilities of the machine learning toolkit Weka of Witten and Frank (2005) that has wide variety in logistic regression, decision tree, neural nets mostly considered applicable for churn prediction described e.g. in Neslin et al. (2006). In addition we tested the SVM implementation of Chang and Lin (2001) considered most powerful for several classification tasks as well as Latent Dirichlet Allocation, a dimensionality reduction and generative modeling approach by Blei et al. (2003) that is believed to work well for skewed features. In our experiments the ultimate method turned out to be the combination of LogitBoost Friedman et al. (2000) and ADTrees Freund and Mason (1999).

Next we describe our method and experimental results in detail including some partial results that appeared promising but did not perform as expected. In Section 2.1 we describe the way we partitioned features by their global properties and in Section 2.2 our initial feature selection procedures. The selection of our classifier and the detailed AUC evaluation is found in Section 2.3.

2. Experiments and Results

We ran our tests on standalone multicore machines with more than 32GB RAM and a condor driven cluster of some older dual-core machines. We run in parallel different algorithms on different machines.

For method selection and parameter tuning the on-line feed-back on 10% of the **test set** was also used but our main procedure consisted of

- 80% for training individual classifiers;
- 10% **heldout set** for method combination;
- 10% **validation set** internal testing that typically performed 2-3% better than the on-line feedback but with more or less kept the relative order of the predictors.

We used online feedback from the 10% test set scarcely and thus we avoided overfitting for the 10%: except for a single task (appetency with the difference in the number of LogitBoost iterations) among all of our submissions, the relative order over the 10% and 100% test set was identical. Up to now however we are not able to explain why the relative order compared to other teams have changed; note that over 10% of the large test set our submission performed best with an AUC score 0.8457 while our final result is behind the winner by 0.0065.

We have managed to select a powerful small feature subset by a LogitBoost based method described in Section 2.2. While classifiers were plain Weka implementations, we

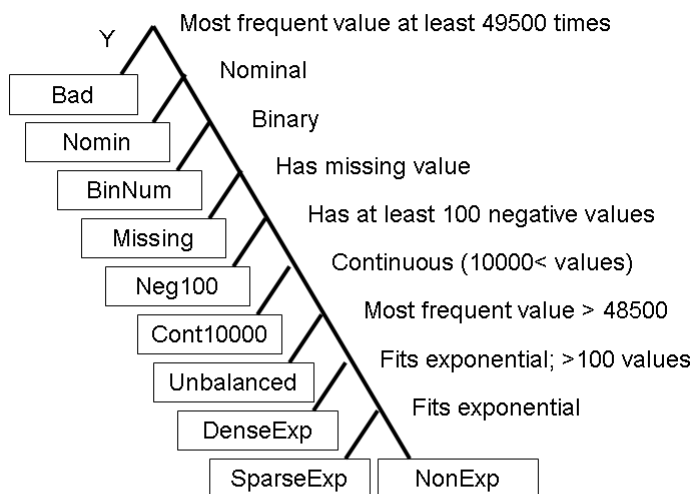


Figure 1: The feature partitioning tree obtained by investigating elementary properties. The “yes” branch is always on the left side.

have intensively used parameter search over the 10+10% heldout and validation set that we have set aside. Combination of classifiers has also improved eventually for the Slow track as described in Section 2.3.

2.1 Feature exploration and partitioning

In our first attempt before test set release we explored feature properties and partitioned the data as described below and in Fig. 1. Our motivation stems from our Web spam classifier of Siklósi et al. (2008) where for example content and link based features are best classified separately with the results combined by e.g. random forest.

Bad: the most frequent or missing value has frequency at least 49500 (10500).

Nomin: nominal with at least 500 non-missing values that is not Bad (269).

BinNum: numeric binary feature that is not Bad (1190).

Missing: numeric with missing values that is not BinNum or Bad (330).

Neg100: numeric with at least 100 negative values that is not Missing or Bad (85).

Cont10000: numeric with continuous range (at least 10,000 distinct values) that is not Missing or Neg100 (503).

Unbalanced: numeric with the most frequent value appearing at least 48500 times that is neither Bad, Missing nor Neg100 (540).

DenseExp: numeric with good fit to the exponential distribution that has more than 100 distinct values and neither Bad, Neg100, Cont10000, Missing or Unbalanced (530).

SparseExp: numeric with good fit to the exponential distribution that has at most 100 distinct values and neither Bad, Neg100, Cont10000, Missing or Unbalanced (445).

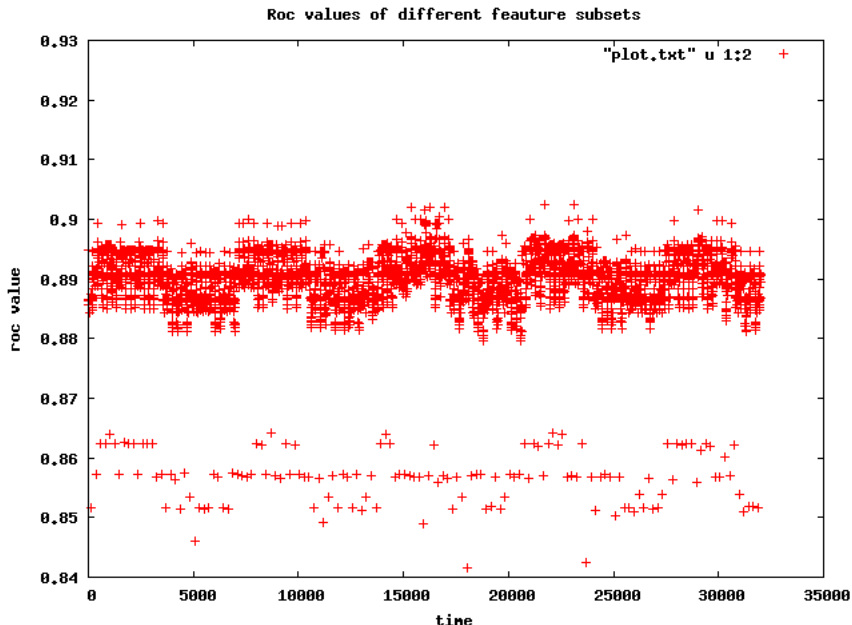


Figure 2: A sample 35,000 feature subset performance selected by our random walk wrapper method biased towards better AUC over the small data set.

NonExp: numeric that is not Bad, Neg100, Cont10000, Missing, Unbalanced, DenseExp and SparseExp (587).

2.2 Feature selection

Our first feature selection attempt relied on well known feature evaluation methods such as Information Gain, Gain Ratio, and Chi Squared Probe. These methods turned out to be useful only as a pre-selection of still a relative large number of features. Information Gain and Chi Squared Probe tends to overscore features with many unique values while Gain Ratio that normalizes scores proportional to the number of unique values tends to overscore features with few unique values. Due to the great number of features we decided to select only those that performed well in both Gain Ratio and Chi Squared Probe measures. The problems with this selection are the following:

- Non-predictive features were selected in a high number.
- The threshold to drop features was generally hard to decide and justify.
- These methods tended to select highly correlated features.

Our second attempt was the classifier-driven approach. One possibility is to start a random walk in the feature space, starting from a preselected feature set, and at each step choosing a feature to add or drop. The evaluation of every feature set can be made by a given classifier. Although this method ran efficiently in our parallel environment, it still

	churn		appetency		upselling	
	heldout	valid	heldout	valid	heldout	valid
Combination LogitBoost		0.7667		0.8537		0.9100
LogitBoost by partition	0.7557	0.7649	0.8668	0.8509	0.9122	0.9099
LogitBoost random	0.7540	0.7612			0.9064	0.9069
Combination log-odds LogitBoost		0.7583		0.8361		0.9026
feature evaluation LogitBoost	0.7335	0.7414	0.8033	0.7924	0.8935	0.8868
Linear SVM	0.6764	0.7026	0.8028	0.7987	0.8845	0.8760
BayesNet by partition	0.6903	0.7012	0.7809	0.7905	0.8095	0.8057
Best AUC selection w/ LogitBoost	0.6624	0.6744				
LDA with BayesNet	0.6008	0.6246	0.5995	0.6278	0.7598	0.7539
Churn predictor			0.5995			
Missing w/ LogitBoost	0.7232	0.7318	0.8394	0.8217	0.8855	0.8931
NonExp w/ AdaBoost	0.7188	0.7359	0.8551	0.8332	0.8835	0.8815
Nominal w/ LogitBoost	0.6657	0.6696	0.8385	0.7868	0.7623	0.7649
Cont10000 w/ LogitBoost	0.6465	0.6631	0.6564	0.6712	0.7419	0.7474
BinNum	0.6369	0.6187	0.7204	0.7233	0.8016	0.8126
DenseExp w/ LogitBoost	0.6294	0.6473	0.6398	0.6591	0.7251	0.7391
NonExp w/ Bayes	0.6230	0.6531	0.5870	0.6393	0.7330	0.7224

Table 1: The AUC value of different classification methods over the whole set (top) and certain feature subsets (bottom) for the three subtasks.

took a long time to find a generally good feature set. In addition in our experiments the method was prone to overfitting: the difference between the exceptionally best and overall good feature subsets diminished when we switched between our heldout and validation sets. A sample run over 170 features of the small data set selected by feature evaluation is shown in Fig. 2.

For this reason decided to use a much simpler method. For each partition of features from Section 2.1, after preselection by feature evaluation we run LogitBoost Friedman et al. (2000) with Decision Stump base classifier. This composite classifier chooses only a few (in our case 10–60) features to build a model. The reason why we ran it separately for each partition is because the running time grew superlinearly with the number of features. We created our feature set from the union of those selected over the partition.

In a simplified partitioned LogitBoost based feature selection method we used a random partition of about 250-300 features in each set. Since the number of sets in the partition was large, we had to iterate selection until only a smaller number of approximately 200 features remained.

We found that the two methods find almost the same features and the classifier results are basically the same. We observe that the features with great imbalance are generally non-predictive: they were only exceptionally selected into the final feature sets. The problems with these methods are the following:

	churn	appetency	upselling	score
Winner (University of Melbourne)	0.7570	0.8836	0.9048	0.8484
LogitBoost + ADTree by partition (final)	0.7567	0.8736	0.9065	0.8456
LogitBoost by partition	0.7496	0.8683	0.9042	0.8407
Combination LogitBoost	0.7409	0.8561	0.8894	0.8288

Table 2: The AUC value of selected final methods over the test set.

- Some discarded features could have been useful for other classifiers—this is a common problem for classifier driven feature selection.
- Non-predictive features can still be selected, although unlikely.
- Uneven distribution of predictive features in partitions may cause dropping some of them, although unlikely.

2.3 Classifier

After a preselection of a few hundred features LogitBoost Freund and Mason (1999) with decision stump as base classifier for the fast track and additionally ADTree Friedman et al. (2000) classifiers for the slow track were used. Both classifiers were used with bagging; cost matrices improved performance for appetency only. The actual values were tuned over our heldout set.

Classifiers over the feature partition did not combine as well as expected over our heldout and validation sets. For all feature subsets we have tested a variety of Weka classifiers, and libsvm with various kernels. Even applied Latent Dirichlet Allocation for dimensionality reduction as in Bíró et al. (2009). None of the methods have outperformed LogitBoost for any feature subset and they did not improve each other in combination.

First we give results based on our 10+10% heldout and validation sets. The overall results are given in the top and the individual subsets in the bottom of Table 1.

- Over our validation set, the combination of individual best feature subset classifiers performed best. Here LogitBoost outperformed other methods and the log-odds based combination proposed by Lynam et al. (2006) deteriorated performance.
- Among the individual feature subsets surprisingly those with missing values performed best. In our guesses these may include responses to questionnaire or call center operators with predictive value stronger than generated features based on service usage. These were followed by the “regular” numeric features with non-exponential distribution, the only exception in that here AdaBoost performed better than LogitBoost. Other classifiers performed much worse for all subsets.
- Global classifiers such as BayesNet of linear SVM performed much worse; LDA as dimensionality reduction performed surprisingly weak.
- Selection by combining best features over our partition slightly outperformed selection by random partition.
- The best feature evaluation method turned out AUC itself: churn classification based on the 95 best features is found in Table 1.

Next we turn to the KDD Cup test set (both 10% and the whole). As we see in Table 2, here classifiers over the best features selected over our partition performed best, thus giving our final submission of AUC 0.8456 that ranked sixth in the competition. The combination of LogitBoost and ADTree improved performance; best combination turned out to be the plain average. Bagging and, in case of appetency, cost sensitive classification have both significantly improved accuracy.

As a final unsuccessful trial we experimented with using training data from one task to improve prediction for the other. The rationale is that a user who churns will not buy upgrades and vice versa. We have also observed that positive labels were disjoint for the three tasks. We tested two methods but could not improve our results:

Classifier combination: We used the results of several final and partial classifiers across tasks in combination. Note that the AUC of one classifier for another task never reached even close to 0.6 and hence failure is no surprise.

Case weighting: If we classify appetency, those who churn are “more negative” than those who just do not buy new services. For a decision stump classifier we may introduce three classes and use a cost matrix with penalties higher for classifying churned customers positive for appetency than non-churned negatives. In this way decision stump acts as regression for the outcome 1 for appetency, 0 for no appetency, and a larger negative value for churn.

Conclusion

In our experiments we observe a clear gain of two classifiers, LogitBoost with decision stump and ADTrees. LogitBoost also performed well for feature selection. We used a feature partitioning method based on statistical properties. The combination of the classifiers over this partition performed close to best (in some cases even better on our heldout sets) and thus we believe that a partition relying also on the meaning of the features (e.g. traffic, sociodemographic or neighborhood based) may outperform the blind anonymous classifiers of the Cup participants. We also expect the call graph extracted from the call detail record can boost performance via the graph stacking framework as e.g. in Csalogány et al. (2007).

Acknowledgments

This work was supported by the EU FP7 project LiWA—Living Web Archives and by grant OTKA NK 72845.

References

- Wai-Ho Au, Keith C. C. Chan, and Xin Yao. A novel evolutionary data mining algorithm with applications to churn prediction. *IEEE Trans. Evolutionary Computation*, 7(6): 532–545, 2003.
- István Bíró, Dávid Siklósi, Jácint Szabó, and András A. Benczúr. Linked latent dirichlet allocation in web spam filtering. In *AIRWeb '09: Proceedings of the 5th international workshop on Adversarial information retrieval on the web*. ACM Press, 2009.

- D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(5):993–1022, 2003.
- C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines, 2001.
- Károly Csalogány, A.A. Benczúr, D. Siklósi, and L. Lukács. Semi-Supervised Learning: A Comparative Study for Web Spam and Telephone User Churn. In *Graph Labeling Workshop in conjunction with ECML/PKDD 2007*, 2007.
- M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.
- Y. Freund and L. Mason. The alternating decision tree learning algorithm. In *In Machine Learning: Proceedings of the Sixteenth International Conference*, 1999.
- J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of statistics*, pages 337–374, 2000.
- H.S. Kim and C.H. Yoon. Determinants of subscriber churn and customer loyalty in the Korean mobile telephony market. *Telecommunications Policy*, 28(9-10):751–765, 2004.
- T.R. Lynam, G.V. Cormack, and D.R. Cheriton. On-line spam filter fusion. *Proc. of the 29th international ACM SIGIR conference on Research and development in information retrieval*, pages 123–130, 2006.
- S.V. Nath and R.S. Behara. Customer churn analysis in the wireless industry: A data mining approach. *Proceedings of the 34th Annual Meeting of the Decision Sciences Institute*, 2003.
- S.A. Neslin, S. Gupta, W. Kamakura, J. Lu, and C.H. Mason. Defection detection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of Marketing Research*, 43(2):204–211, 2006.
- Dávid Siklósi, András A.Benczúr, István Bíró, Zsolt Fekete, Miklós Kurucz, Attila Pereszlényi, Simon Rácz, Adrienn Szabó, and Jácint Szabó. Web Spam Hunting @ Budapest. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2008.
- A. Ultsch. Emergent self-organising feature maps used for prediction and prevention of churn in mobile phone markets. *Journal of Targeting, Measurement and Analysis for Marketing*, 10(4):314–324, 2002.
- Chih-Ping Wei and I-Tang Chiu. Turning telecommunications call details to churn prediction: a data mining approach. *Expert Syst. Appl.*, 23(2):103–112, 2002.
- Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005. ISBN 0120884070.