Multi-View Risk Classification for Customer Due Diligence

Juan Manuel Gonzalez Huesca[1,3], **Simon van der Zon**[1],
Werner van Ipenburg[2], **Jan Veldsink**[2], **Mykola Pechenizkiy**[1]
[1]Department of Computer Science, TU Eindhoven, the Netherlands
[2]Rabobank Compliance, the Netherlands
[3]Data Science and Engineering group, ASML Netherlands B.V., the Netherlands

Abstract

In the last years, it has become critical for financial institutions to develop efficient solutions that leverage all available heterogeneous customer data. While financial crime is growing in scale and complexity, financial analysts require a new set of tools to protect and shield the financial system against illegal activities. We propose a multi-view semi-supervised risk classification approach that makes use of text and financial transactions data for Customer Due Diligence (CDD) reporting. We conducted a case study which illustrates that by using multiple learning sources (views) at the same time, higher performance is achieved compared to models learned on each source individually. Furthermore, our system allows to explain why and how the prediction was done thus providing new insight to domain experts.

Introduction

The Customer Due Diligence (CDD) report is a key requirement of the Know Your Customer (KYC) processes, as it provides a complete overview to assess customer risk based on different characteristics. These processes are the cornerstone for effective anti-money laundering and counter-terrorism financing programs, including other illegal activities.

There are numerous manual steps in the current processes and most of the CDD report is done manually by domain experts, leading to inconsistent results and low efficiency, while only a small set of all the available data is used. Another challenge is the complexity of manually extracting knowledge from big data and finding patterns, this is an extremely difficult if not impossible task for these domain experts.

Recently, financial institutions have strengthened the KYC (PricewaterhouseCoopers 2014) processes and due to their importance, we were motivated to find more robust artificial intelligence (AI) solutions that leverage the high amount of available data: unstructured (text) and structured (financial transactions).

We proposed an AI solution that helps automating a risk classification task which provides more insight to CDD experts. The nature of these data do not allow for training one model to leveraging the structural properties of the different sources. Therefore, a multi-view learning paradigm has been implemented which outperformed single-view learning scenarios in many fields (Xu, Tao, and Xu 2013). This implementation also considers a semi-supervised learning schema which only needs a relatively small initial labeled training set and that leverages a large amount of heterogeneous unlabeled data. Another major consideration was the need of an interpretable model. The goal was to augment the capabilities of domain experts with new knowledge by displaying the top features (per view) driving the classification.

Framework

The proposed framework implements co-training, one of the most popular multi-view learning algorithms (Li, Li, and Fu 2016). As we focused on improving the process for the CDD report, while including the domain experts in the knowledge discovery process, co-training was chosen because it is a model-based integration method and has a good trade-off between classification accuracy and model interpretability (Molnar 2018).

In a co-training schema, two classifiers are trained independently, after which the labels predicted by each of them (one per data view) help each other to retrain themselves and augment the labeled dataset (as shown in Figure 1). This allowed us to select a more appropriate classifier per data type (text or financial transactions).

This framework provides flexibility to augment the labeled dataset based on prediction certainty: we only consider the predicted instances with the highest probability for both classes to be included in the labeled dataset. By using this approach, as the number of training and predicting iterations increases, the performance is better and our solution is more robust.

We assume to have an instance space $O = O_1 + O_2$, where O_1 and O_2 represent the different views (text and financial transactions) of the same instance (customer). In this way, every training instance o has two components: o_1, o_2. One key assumption is that every view contains enough data to build an initial classification model.

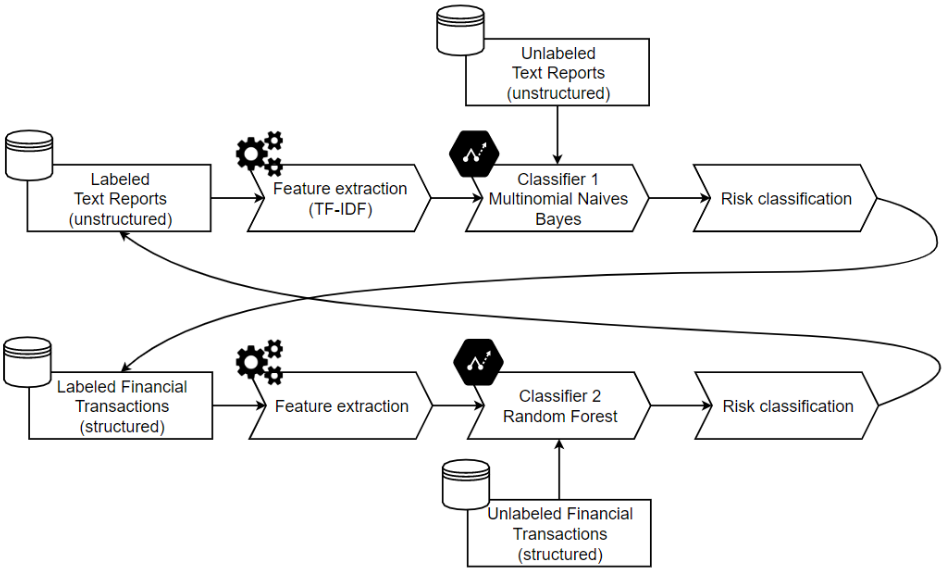

Figure 1: Framework for multi-view predictive analytics.

Text classification analysis (unstructured data)

Text represents most of the information available in the world; this may be the reason why text mining has become very important and why many companies and research groups are very interested in finding ways to leverage and extract value from it for many use cases (Miner et al. 2012). This provides us with strong reasons to focus on and leverage the potential of text in knowledge discovery. For financial institutions this applies as well, they want to start using the huge volume of mostly unstructured text data to gain insight and improve KYC processes.

Text classification analysis is divided in three steps: text preprocessing, feature extraction and text classification.

The first step, text preprocessing, is defined with seven phases: tokenization, formatting numbers, formatting emails and URLs, removing stopwords, removing punctuation, stemming and removing monosyllabic words. This step is important due to the nature of text documents, it is very difficult to process raw text files so some clean up is needed.

The second step is to extract features from raw documents. Based on its properties (Baeza-Yates and Ribeiro-Neto 2008), we used Term Frequency-Inverse Document Frequency (TF-IDF) (Manning, Raghavan, and Schütze 2008) to create a vector space representation for every document associated with a customer. Luhn (1958) suggested that both extremely common and extremely uncommon words were not useful for indexing.

The third and final step is the text classification. In this step, the chosen classification algorithm is Multinomial Naive Bayes (MNB) because it is a canonical text classification algorithm usually used as a baseline. This algorithm based on the Bayes theorem performs better than multivariate Bernoulli, proving an average of 27% reduction in error (McCallum and Nigam 1998). This supervised learning algorithm has the "naive" assumption that every pair of features are independent. In real-world applications this assumption usually does not hold, but empirical experiments have shown a competitive performance with state-of-the-art supervised algorithms, leading in some cases to a dependence cancelation among the different features. Furthermore, Naive Bayes is inherently interpretable, contributing to one of our main goals.

Financial transactions classification analysis (structured data)

One data type that is ubiquitous in financial institutions are the transactions. Every customer has at least one account where they have double-entry transactions defined as "debit" or "credit". Extracting features from financial transactions is a challenging task. We used several properties of the transactions (time stamp, transaction type, amount of money, currency, balance, etc.), which were provided by the CDD experts, obtained through literature review, and statistics, to engineer features from this time series dataset.

The chose algorithm for this classification analysis is Random Forest (Breiman 2001) because of its performance and because there exist very efficient explanation methods for tree ensembles, thus having the ability to explain predictions. Also, ensemble algorithms have shown better performance than a single model, that also motivates us to use Random Forest.

Co-training

In the financial services industry there is a high amount of data from several heterogeneous sources. While this may be seen as a drawback, empirical evidence shows that by exploring several views of the same instance, an algorithm can be more effective and it can generalize better than single-view classifiers (Xu, Tao, and Xu 2013).

There are different methodologies for data integration (Ritchie et al. 2015): a) early integration, where there is a simple concatenation of the several views in a single feature space; b) intermediate integration, where the data is transformed in a common feature space before combining them; and c) late integration, where we analyse each view separately and the results are combined. Co-training falls in the latter, where we have one classifier per view after which the predictions are combined.

The co-training algorithm (Blum and Mitchell 1998) allows us to exploit multiple views of the same instance, learn from a small labeled dataset and leveraging unlabeled data, all of this while providing a good classification performance with prediction explanations.

```
Data: $L_1, L_2, U_1, U_2$
Result: Trained classifiers: $h_1, h_2$
begin
    Given:
    • a set $L_1$ of labeled training examples for $h_1$
    • a set $L_2$ of labeled training examples for $h_2$
    • a set $U_1$ of unlabeled examples for $h_1$
    • a set $U_2$ of unlabeled examples for $h_2$
    while $k \neq K$ do
        Use $L_1$ to train a classifier $h_1$
        Allow $h_1$ to label 1 positive and 1 negative examples from $U_1$
        Add these self-labeled examples to $L_2$
        Remove self-labeled examples from $U_1$ and $U_2$
        Use $L_2$ to train a classifier $h_2$
        Allow $h_2$ to label 1 positive and 1 negative examples from $U_2$
        Add these self-labeled examples to $L_1$
        Remove self-labeled examples from $U_1$ and $U_2$
        $k$ += 1
    end
end
```

Algorithm 1: Proposed Co-training Algorithm.

Based on the original co-training algorithm we propose two modifications (as shown in Algorithm 1). The first is to change the order of the training and predicting operations. We propose to do the training and prediction steps alternately between the two classifiers, contrary to the original algorithm where they are simultaneous. Our second modification is to have two labeled (L_1 and L_2) and unlabeled (U_1 and U_2) datasets, and treat them separately (one per view), such that labeled instances from $h1$ can go only to the training instances for $h2$, something that is not happening in the original algorithm.

The output of five experiment configurations is recorded in Table 1, where the combined classifier is done using a probability sum-rule because it is less sensitive to noise than other rules (Kittler et al. 1998) and it has a high recognition rate (Tulyakov et al. 2008): Single classifier (two separate classifiers with simple score combination, without iterations and without interaction between them), self-training (each classifier is being retrained in isolation using its predicted samples), co-training original (original co-training algorithm), co-training proposal 1 (described in Algorithm 1) and co-training proposal 2 (only when both classifiers predict the same label). The next section discusses the impact of each modification by look at the change in performance.

Experimental evaluation and results

The dataset contains 693 instances, each one of them consist of two views: text and financial transactions features. Building the dataset was a challenging task: we had around 35 thousands files in Dutch language, in different formats (xls, xlsx, doc, docx, pdf, rtf, etc.) from several sources, so we designed and implemented an information retrieval system to obtain as much of the data for each customer with high precision. Due to customer data confidentiality it is not possible to show sample vectors of features.

When using a co-training schema, the first assumption is that we have a small initial labeled dataset. In the original co-training implementation, 12 labeled instances initial instances where used with 30 iterations. In our case, as our dataset is bigger, we experimented with 12, 24 and 50 initial labeled instances. Another parameter that we vary is the number of iterations: 10, 20, 30, 40, 50, 60, 70, 80 and 100. We wanted to analyze the trade-off between the number of initially labeled instances, number of iterations and algorithm performance. The average performance values of 10 runs are recorded in the results.

The dataset has a class imbalance of low/high risk ratio 1:2. The metrics to evaluate performance were accuracy (ACC) and the area under the Receiver Operating Characteristic (ROC) known as AUC.

We showed empirical evidence that our co-training proposal outperformed the other experiments, and more importantly, the insight provided by both text and financial transaction features was relevant and useful for domain experts. The latter was confirmed by a CDD expert through an interview and two case studies, where we showed new patterns that can be used in future CDD reporting cases.

For most cases, as the number of initially labeled instances increases, the classification performance increases, but we decided to use 24 initially labeled instances and 30 iterations as this experiment showed more steady results and higher insight value to domain experts. We can see that our co-training proposals perform best in the set of experiments shown in Table 1. Also, the co-training proposal 1 had the standard deviation of the accuracy and AUC scores.

Another useful observation was the correlation between some text and financial transaction features. This is important to domain experts because sometimes there is no text available to start a CDD report. For those cases where the text data is missing, with this moderate correlation (>0.3 Pearson's correlation coefficient) we may be able indicate some initial text inquiries based on just the financial transactions' classification analysis, however further research needs to be done to confirm the consistency of these results.

	Text classifier		**Transactions classifier**		**Combined classifier**	
Configuration	**ACC**	**AUC**	**ACC**	**AUC**	**ACC**	**AUC**
Single classifier	0.8088	0.7816	0.5104	0.4966	0.8093	0.7823
Self training	0.6684	0.6104	0.6440	0.6134	0.6606	0.6017
Co-training original	0.6482	0.6007	0.6254	0.6147	0.6456	0.5986
Co-training proposal 1	**0.8143**	**0.8172**	0.6563	**0.6439**	**0.8101**	**0.8129**
Co-training proposal 2	0.7412	0.7195	**0.6664**	0.6084	0.7445	0.7257

Table 1: Accuracy (ACC) and AUC scores for experiments with 24 initial training instances and 30 iterations.

A senior CDD expert validated the results and gave a positive feedback: ”the tool provided new relevant insight for target cases while having an accurate prediction”. The domain experts confirmed that this tool has the potential to make their day to day work more efficient and effective.

Additionally, we carried out an experiment relying on early data integration with just one classifier (Random Forest). It was surprising that a mere concatenation of text and financial transactions features achieved very good performance. However, the drawback is in the loss of interpretability, the model did not provide insights for the domain experts. With this experiment we were able to see the trade-off between accuracy and interpretability, where the latter is of crucial importance for our application.

Conclusions

The analysis showed empirical evidence of the benefits of co-training for a risk classification in the context of CDD reporting. We showed that a multi-view learning scenario (text and financial transactions data) with a co-training schema outperforms a single-view learning scenario. We propose two modifications to the original co-training scheme, and showed that each modification improves the performance (accuracy and AUC scores) even further. Furthermore, the inherently interpretable nature of the models for each view allowed domain experts to extract valuable explanations.

This framework allowed us to augment the capabilities of domain experts and make them part of the knowledge discovery loop by adding different sources to the machine learning loop. Our method managed to provide new insights, which can be used to speed up the process of CDD reporting, allowing the domain expert to focus on high risk customers more effectively.

Most importantly overall, it showed how a financial institution can leverage a small set of labeled heterogeneous customer data in a knowledge discovery process to assess customer risk and combat financial crimes. This will definitely improve the KYC processes and will help protecting the financial system from illegal activities.

References

Baeza-Yates, R., and Ribeiro-Neto, B. 2008. *Modern Information Retrieval: The Concepts and Technology Behind Search*. USA: Addison-Wesley Publishing Company, 2nd edition.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT’ 98, 92–100. New York, NY, USA: ACM.

Breiman, L. 2001. Random forests. *Machine Learning* 45(1):5–32.

Kittler, J.; Hatef, M.; Duin, R. P. W.; and Matas, J. 1998. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(3):226–239.

Li, S.; Li, Y.; and Fu, Y. 2016. Multi-view time series classification: A discriminative bilinear projection approach. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM ’16, 989–998. New York, NY, USA: ACM.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

McCallum, A., and Nigam, K. 1998. A comparison of event models for naive Bayes text classification. In *Learning for Text Categorization: Papers from the 1998 AAAI Workshop*, 41–48.

Miner, G.; Elder, J.; Fast, A.; Hill, T.; Nisbet, R.; and Delen, D. 2012. *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Elsevier Science.

Molnar, C. 2018. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. USA: Bookdown.

PricewaterhouseCoopers. 2014. Know your customer: Quick reference guide. https://www.pwc.com/gx/en/financial-services/publications/assets/pwc-anti-money-laundering-know-your-customer-quick-reference-guide.pdf, Last accessed on 2018-05-28.

Ritchie, M. D.; Holzinger, E. R.; Li, R.; Pendergrass, S. A.; and Kim, D. 2015. Methods of integrating data to uncover genotype–phenotype interactions. *Nature Reviews Genetics* 16:85–97.

Tulyakov, S.; Jaeger, S.; Govindaraju, V.; and Doermann, D. S. 2008. Review of classifier combination methods. In *Machine Learning in Document Analysis and Recognition*.

Xu, C.; Tao, D.; and Xu, C. 2013. A survey on multi-view learning. *CoRR* abs/1304.5634.