Data science for tax administration

Pijnenburg, M.G.F.

3

Extending Logistic Regression Models with Factorization Machines

In this section we address the following research question:

> **Research Question** *Can we improve on current audit selection models by incorporating categorical variables with many values (like 'industry sector') that are valued by experts doing manual audit selection?*

Interest in including categorical variables with many levels was raised by a practical problem at the NTCA: a much used risk model did not contain these variables because they were rejected in a feature selection step. Nevertheless, experienced auditors used these variables successfully in the manual selection of audits. The idea originated that complementary techniques must be used to exploit the information hidden in these variables.

The research has value outside the realm of taxes as many data scientists encounter categorical features with many levels and are looking for the right way to incorporate them in risk models or other applications, as is evident from the questions on the subject on *www.stackoverflow.com*.

We focus on logistic regression models, a technique frequently used for risk modeling within tax administrations. Including categorical variables with many levels in a logistic regression model easily leads to a sparse design matrix. This can result in a big, ill-conditioned optimization problem causing overfitting, extreme coefficient values and long run times. Inspired by recent developments in matrix factorization, we propose four new strategies to overcome this problem. Each strategy uses a Factorization Machine that transforms the categorical variables with many levels into a few numeric variables that are subsequently used in the logistic regression model. The

application of Factorization Machines allows to include interactions between the categorical variables, often substantially increasing model accuracy. The four strategies have been tested on a data set of the NTCA and three public data sets, demonstrating superiority of the approach over other methods of handling categorical variables with many levels. The chapter is based on the following article:

- M. Pijnenburg and W. Kowalczyk. Extending logistic regression models with factorization machines. In *International Symposium on Methodologies for Intelligent Systems*, pages 323–332. Springer, 2017

3.1 Introduction

Logistic regression is a well-known classification algorithm that is frequently used by businesses and governments to model risks. However, logistic regression will run into problems when one tries to include categorical variables with many levels. The standard approach will transform each level of each categorical variable into a binary ('dummy') variable (see, e.g., [42]), resulting in a large, sparse design matrix. The sparsity usually leads to an ill-conditioned optimization problem, resulting in overfitting, extremely large values of model coefficients and long run times or even lack of convergence. The size and sparsity of the design matrix will increase even more if interactions are included between categorical variables with many levels or interactions between these categorical variables and some numeric variables. Finally, a large design matrix leads to a model with many coefficients, making it difficult to interpret.

Existing approaches for incorporating multi-level categorical variables into logistic regression reduce the problem of sparsity at the price of losing some information from data and consequently leading to models of inferior quality, see section 3.2.1 for an overview. This became clear to the authors when working on a risk model for selecting risky VAT tax returns for the Netherlands Tax and Customs Administration (NTCA), see also Section 2.5. Although the risk model performed pretty well, one experienced auditor was able to outperform the model by manually extracting information from categorical variables with many levels such as industry sector code and zip code. This information was clearly not picked up by the model, where standard approaches had been followed to include these categorical variables. The real-life example of the NTCA will be referred to in this chapter several times.

Logistic regression is traditionally used at the NTCA since it is a standard tool in the industry and the role of various features can be easily interpreted. Moreover, it performs well and its output can be interpreted as probabilities. This latter fact is important since it allows a decoupling of two key components of a tax return risk model: the probability of an erroneous tax return and the size of the financial loss connected to the error in the tax return, see [11].

In this chapter we propose four strategies for including categorical variables with many levels into a logistic regression model, by making use of Factorization Machines, [97]. Factorization Machines transform the categorical variables into vectors of numerical ones, taking interactions of the categorical variables into account. Factorization Machines can be viewed as an extension of matrix factorization methods, [58], that in turn have been developed in the context of recommendation systems, stimulated by the Netflix Challenge.

The chapter is organized as follows. Section 3.2 reviews the related literature. Section 3.3 introduces the four strategies of combining Factorization Machines and

Logistic Regression. Section 3.4 describes the experiments on four data sets. Section 3.5 contains conclusions and a discussion of the results.

3.2 Related Research

3.2.1 Existing Methods for Many Levels

A number of approaches have been suggested to deal with categorical variables with many levels. Many approaches focus on grouping levels of categorical variables into a smaller number of levels. This grouping can be done in a supervised manner (i.e. involving the target) or in an unsupervised way.

A well-known supervised way of grouping levels comes from the decision tree algorithm CART [21]. Here, levels are ordered by the percentage of cases in the target class. Subsequently, all levels with the percentage above a certain threshold value are grouped into one new level, and the remaining levels into another one. Breiman et al. [21] proved that this approach will find the optimal partitioning of a train set under the conditions that the target is binary, the new categorical variable has two levels, and a convex criterion (like Gini) is used to measure the quality of the partition. Several extensions of this approach have been developed, e.g. [24, 26], but they either lack the guarantee of finding the optimal partitioning, or have a substantially higher order of complexity.

Other, frequently employed, supervised ways of grouping levels include search methods, like forward or stepwise search [16]. Typically one starts with each level forming a group of its own. Then groups are merged one at a time based on various criteria, leading to fewer groups. This approach is used, for instance, in the CHAID algorithm [57].

A simple, but often effective, unsupervised way of grouping levels has been proposed by Hosmer and Lemeshow [53]. They suggest to group levels that occur infrequently. This approach gave the best results on our data sets from all supervised and unsupervised groupings tried. Another frequently used unsupervised approach is to let experts group the levels.

Other methods than grouping levels have been put forward. For example, in a data pre-processing step the categorical variable with many levels can be transformed into a numeric variable with help of an Empirical Bayes criterion, [77]. Another approach, see [10], is to find additional numeric predictors. For instance, if the categorical variable is “city”, the number of inhabitants of the city could be added to the data set as a predictor.

Although all these approaches for dealing with categorical variables with many levels have their merits, none of them was able to improve the current risk model at

the NTCA. So other approaches need to be considered.

3.2.2 Factorization Machines

Factorization Machines are introduced in a seminal paper of Rendle [97]. This class of models is often employed for recommendation systems, where categorical variables with many levels occur frequently. The model equation of a Factorization Machine is given by:

$$\hat{y} = w_0 + \sum_{j=1}^{s} w_j x_j + \sum_{j=1}^{s} \sum_{k=j+1}^{s} \langle \mathbf{v}_j, \mathbf{v}_k \rangle x_j x_k \tag{3.1}$$

where $\hat{y}$ is the predicted value for an observation with variables $x_1, \ldots, x_s$. The w's are numeric coefficients to be fitted and the $\mathbf{v}$'s are vectors to be fitted (one for each variable). All vectors $\mathbf{v}$ have the same (usually small) length r, which is an input parameter. These vectors can be interpreted as low dimensional numeric representations of levels.

The interesting part of equation (3.1) are the interaction terms. Instead of assigning a new coefficient w_{jk} to each interaction term, a Factorization Machine models the interaction coefficients as an inner product between the vectors $\mathbf{v}_j$ and $\mathbf{v}_k$. The introduction of such a vector for each variable reduces the number of interactions from $O(s^2)$ to $O(rs)$, so from quadratic to linear in the number of variables s. Typically, variables x_j in a Factorization Machine are binary variables resulting from transforming a categorical variable with many levels in dummy variables. In this case the number of coefficients is thus not quadratic in the number of levels, but linear. Note that when s is small (e.g., $s \leq 2r + 1$) there is no reduction in the number of coefficients.

The loss function that is used to find the optimal values of parameters in (3.1) usually involves a regularization term which controls the L^2 norm of model parameters. To find an optimum of the loss function several techniques can be used, among them Markov Chain Monte Carlo, Alternating Least Squares and Stochastic Gradient Descent [98].

3.3 Combining Logistic Regression with Factorization Machines

In this section we propose four new strategies for extending Logistic Regression with Factorization Machines. The key idea of these new approaches is the usage of Factorization Machines for squeezing relevant information from many-level categorical variables and their interactions into numeric variables and incorporating these latter in a logistic regression model. In this way, potential problems with large and sparse

design matrix are handled by Factorization Machines, while Logistic Regression takes care of combining "non-sparse" variables in a standard way.

Notation. We model a binary target y with help of p numeric variables $x_1, \ldots, x_p$, and q categorical variables $d_1, \ldots, d_q$ with $l_1, \ldots, l_q$ levels.

We will compare the performance of our four strategies with two benchmarks: (1) a logistic regression model without categorical variables, and (2) a logistic regression model where infrequent levels have been grouped as suggested by [53], see section 3.2.1. We start by introducing these latter methods in more detail.

3.3.1 Plain Logistic Regression (PLM)

The PLM model consists of a standard logistic regression model of the numeric variables $x_1, \ldots, x_p$. The model equation is:

$$\hat{y} = \frac{1}{1+e^{-z}}, \text{ where } z = \alpha_0 + \alpha_1 x_1 + \ldots + \alpha_p x_p. \tag{3.2}$$

The coefficients α_i are estimated by finding the unique maximum of the log-likelihood function over the train set.

3.3.2 Logistic Regression with Grouping (LRG)

This model groups infrequent levels of the categorical variables with many levels $d_1, \ldots, d_q$ into a default level for each d_j. The actual threshold for calling a level 'infrequent' depends on the size of the data set and can be found in Table 3.1. The grouping of infrequent levels leads to a new set of categorical variables $\tilde{d}_1, \ldots, \tilde{d}_q$ with less levels. Next, a standard approach is followed to replace categorical variables by dummy variables, i.e. each $\tilde{d}_j$ is transformed into $l_j - 1$ binary variables. Subsequently, PLM is applied on these binary variables and the numeric predictors.

In mathematical terms, the LRG model is given by:

$$\begin{aligned}\hat{y} = \frac{1}{1+e^{-z}}, \quad \text{where } z = {} & \alpha_0 + \alpha_1 x_1 + \ldots \alpha_p x_p + \\ & \alpha_{11} b_{11} + \ldots + \alpha_{1 l_1 - 1} b_{1 l_1 - 1} + \ldots + \alpha_{q l_q - 1} b_{q l_q - 1},\end{aligned} \tag{3.3}$$

where b's are binary variables. Note that in order to limit notation, we denote the coefficients of the logistic regression in all model equations ((3.2), (3.3), (3.4), (3.5), (3.6), and (3.7)) with $\alpha_0, \alpha_1, \ldots$, despite the fact that the values of these coefficients differ.

3.3.3 LRFM1

The model LRFM1 (Logistic Regression with Factorization Machines 1) is the first model showing our new approach. The categorical variables with many levels $d_1, \ldots, d_q$ are first put into a Factorization Machine f_0 whose coefficients are estimated from a train set with the target variable y. The output of f_0 — denoted by g_0 — is then added to the model equation of the logistic regression. Therefore,

$$\begin{aligned} g_0 = f_0(d_1, \ldots, d_q) \quad \text{and} \\ \hat{y} \quad = \tfrac{1}{1+e^{-z}}, \text{ where } z = \alpha_0 + \alpha_1 x_1 + \ldots \alpha_p x_p + \alpha_{g_0} g_0. \end{aligned} \tag{3.4}$$

3.3.4 LRFM2

Although LRFM1 is able to model interactions between categorical variables with many levels, it does not model interactions between the variables $d_1, \ldots, d_q$ and one or more numeric variables x_j. For this reason we allow the model equation (3.4) to be extended with additional variables $g_1, \ldots g_t$, where $t \leq p$. Each g_j is a prediction from a Factorization Machine f_j that takes as input the categorical variables $d_1, \ldots, d_q$ and a variable $\bar{x}_j$. The variable $\bar{x}_j$ is a discretized version of x_j, obtained by an equal frequency binning with 5 bins.

The coefficients of the Factorization Machine f_j are learned from a train set that contains the target y. Only variables g_j that significantly improve the results on the train set (compared to the model with only g_0, significance level $\alpha = 0.05$) will enter the model equation. Therefore, the model equation for LRFM2 is:

$$\begin{aligned} g_j = f_j(\bar{x}_j, d_1, \ldots, d_q) \text{ and } \hat{y} = \frac{1}{1+e^{-z}}, \text{ where} \\ z = \alpha_0 + \alpha_1 x_1 + \ldots \alpha_p x_p + \alpha_{g_0} g_0 + \alpha_{g_1} g_1 + \ldots + \alpha_{g_t} g_t. \end{aligned} \tag{3.5}$$

3.3.5 LRFM3

Instead of learning the coefficients of a Factorization Machine f on a train set with known binary target y, we can do an intermediate step. We first fit a logistic regression model with the numeric variables $x_1, \ldots, x_p$ on the train set (so without $d_1, \ldots, d_q$), and then compute the deviance residuals r_i (see [53]):

$$r_i = \pm \sqrt{2 \left[y_i \log \frac{y_i}{\hat{y}_i} + (1 - y_i) \log \frac{1 - y_i}{1 - \hat{y}_i} \right]},$$

where $\hat{y}_i$ denotes the predicted probability that $y_i = 1$ and the sign is $+$ iff $y_i = 1$. The residual vector $\mathbf{r}$ can then be used to train the coefficients of the Factorization Machine instead of the original target $\mathbf{y}$. This will give a Factorization Machine $\tilde{f}$.

Note that the Factorization Machine is now performing a regression task, instead of classification. LRFM3 is described by the equations:

$$\begin{aligned} h_0 &= \tilde{f}(d_1, \ldots, d_q) \text{ and} \\ \hat{y} &= \tfrac{1}{1+e^{-z}}, \text{ where } z = \alpha_0 + \alpha_1 x_1 + \ldots \alpha_p x_p + \alpha_{h_0} h_0. \end{aligned} \tag{3.6}$$

3.3.6 LRFM4

Similarly as LRFM1 was extended to LRFM2, we can extend LRFM3 to LRFM4. More specifically, we form additional variables h_j by including a discretized numeric variable $\bar{x}_j$ in the Factorization Machine that is trained on residuals. Only variables h_j that significantly ($\alpha = 0.05$) improve the result on the train set will enter the model equation. This provides our last strategy:

$$\begin{aligned} h_j &= \tilde{f}(\bar{x}_j, d_1, \ldots, d_q) \text{ and } \hat{y} = \tfrac{1}{1+e^{-z}}, \text{ where} \\ z &= \alpha_0 + \alpha_1 x_1 + \ldots \alpha_p x_p + \alpha_{h_0} h_0 + \alpha_{h_1} h_1 + \ldots + \alpha_{h_r} h_r. \end{aligned} \tag{3.7}$$

3.4 Experiments

Interest in including categorical variables with many levels was raised by a practical problem at the NTCA. Most research has been focused on this data set. However, in order for the research to be reproducible, we also applied our approach to three public data sets in the UCI Repository [65] that contain categorical variables with many values. See Table 3.1 for some key characteristics of the four data sets. We considered a variable to have 'many levels' if the number of levels exceeded 30. This number corresponds roughly with the situation where the design matrix becomes sparse in our four data sets. Below we will describe the data sets, the exact parameters of the experiments, and the results.

3.4.1 Data Sets

3.4.1.1 Tax Administration

This data set consists of approximately 80,000 audited VAT tax returns. A small part of these tax returns (17.5%) were found to contain one or more erroneous statements when audited. The data set has 33 numeric variables that are the result of a stepwise feature selection process that started with over 500 variables.

	Data Set			
	tax	kdd98	retail	census
# observations	86,235	95,412	532,621	199,523
# numeric vars	33	18	3	23
% target = 1	17.5%	5.0%	29.1%	6.2%
% target = 0	82.5%	95.0%	70.9%	93.8%
threshold infreq. level (LRG meth.)	100	100	1000	100
cat. features (# levels)	zipcode (1,027) industry sector (3,747)	DMA (207) RFA 11 (101) RFA 14 (95) RFA 23 (87) OSOURCE (869) ZIP (19,938)	InvoiceNo (25,900) Description (4148) CustomerID (4,373) Cntr (38)	ind. code (52) occ. code (47) prev. state (51) hh. stat (38) cntr fthr (43) cntr mthr (43) cntr birth (43)

Table 3.1: Summary of data sets used in the logistic regression / Factorization Machine experiments

3.4.1.2 KDD 98 cup

This data set comes with a binary target and a numeric target. We only use the former as we focus on classification. Additionally, we only used the 'learning' data set and not the 'validation' data set. The original data set contains 480 variables. We selected 22 variables to get a data set similar to the tax data. The variable selection has been done by keeping the variables reported in [44] (page 147, Table 6) and adding the two variables with many levels: OSOURCE and ZIP. Missing values have been replaced by 0, except for WEALTH where the median has been inserted.

3.4.1.3 Online Retail

The following data processing steps have been performed: (1) canceled transactions have been removed, (2) InvoiceDate has been split in a date part and a time part, (3) StockCode has been removed since its values can be mapped almost one-to-one to the values of Description. The data set does not contain a target. We created a binary target by defining the target to be 1 if the variable Quantity is larger or equal to 10, and 0 otherwise. After this, Quantity has been removed from the data set.

3.4.1.4 Census

The following data processing steps have been taken: (1) Weight has been discarded as is advised in the data description, (2) variables that are aggregates of other variables have been removed. For instance, Major Industry Code aggregates the levels of Industry Code. For this reason the variables: Major Industry Code, Major Occupation Code, Previous Region, Household Summary, Migration Code Reg, Migration Sunbelt, and Live1year House have been removed. Similarly, (3) Veteran Questionnaire is removed since most relevant information is in Veteran Benefits. Finally, (4) Migration Code MSA has been removed since it is highly collinear with Previous State.

3.4.2 Model Quality Measures

Various performance measures can be used to assess the results of a classification algorithm (e.g. accuracy, precision, area under the curve, recall). At the NTCA audit capacity is limited and rather fixed. For this reason, one is more interested in using the available capacity to the best (so avoiding false positives), than trying to find all taxpayers that make an error (avoiding false negatives). *Precision* (i.e. the true positives divided by the sum of true positives and false positives) is therefore a natural measure, and more useful than accuracy (the sum of true positives and true negatives divided by all cases). We applied precision at a '10% cut-off level', measured on a test set (i.e., we select the 10% highest scoring observations of a test set and then compute the precision), in similar fashion as is done at the NTCA.

For reference, we have also provided the frequently used Area Under the Curve (AUC) characteristic. For confidentiality reasons the precision and AUC could not be reported for the tax data set (although tested in practise). Instead we have provided the *increase in precision*, measured using the Plain Logistic Regression as a baseline. For instance, if the precision of Plain Logistic Regression is 60%, and the precision with the new technique is 66%, then the *increase in precision* is $100\% \cdot (66 - 60)/60 = 10\%$.

In our experiments we used 5-fold cross validation to get reliable estimates of all quality measures listed above.

3.4.3 Settings Factorization Machines

In our experiments Factorization Machines were constructed with libFM software [98] with the following settings: number of iterations: 25, lengths of the parameter vectors $\mathbf{v}$: 16 (see equation (3.1)), and the optimization technique is set to 'MCMC'. The standard deviation of the normal distribution that is used for initializing the parameter vectors $\mathbf{v}$ in MCMC is set to 0.1. When building the Factorization Machines in

	Precision (%)			Precision (% increase w.r.t. PLR)				AUC		
	kdd98	retail	cens.	tax	kdd98	retail	cens.	kdd98	retail	cens.
PLR	6.05	69.3	42.7	0.0%	0.0%	0.0%	0.0%	.5348	.7845	.9358
LRG	7.88	80.7	44.7	2.7%	30.3%	16.4%	4.6%	.5748	.8442	.9439
LRFM1	7.92	99.6	44.5	3.9%	31.0%	43.7%	4.2%	.5775	.9634	.9426
LRFM2	7.92	99.8	44.5	5.1%	31.0%	44.1%	4.2%	.5775	.9689	.9426
LRFM3	6.93	99.5	44.3	6.1%	14.6%	43.6%	3.7%	.5502	.9710	.9410
LRFM4	6.79	99.5	44.3	9.6%	12.3%	43.6%	3.7%	.5553	.9713	.9410

Table 3.2: Performance (measured in precision, increase of precision with relation to Plain Logistic Regression, and Area Under Curve) for each strategy on all data sets using five-fold cross-validation. Absolute values for the tax data set have been deliberately omitted for confidentiality reasons.

approaches LRFM1 and LRFM2 we set the 'task' to 'classification', and in the remaining two cases to 'regression'.

3.4.4 Results

The results of applying two benchmark methods Plain Logistic Regression (PLR) and Logistic Regression with Grouping (LRG), as well as our four strategies are summarized in Table 3.2. The columns *precision* and *AUC* are not filled for the tax administration data set, because of confidentiality reasons.

3.5 Conclusion and Discussion

Looking at Table 3.2, some conclusions can be drawn. First, our proposed strategies give better results than plain logistic regression for all data sets. Second, when comparing with LRG (the strategy that gave the best results from the methods of section 3.2.1), we see a subtler picture. Our methods outperform LRG clearly on the tax data and the retail data. For the tax data we see that taking interactions of the categorical variables with numeric variables into account, while training on residuals (i.e. LRFM4), can substantially improve the result. When looking at the data set kdd98, the approaches that train directly on the target y (LRFM1 and LRFM2) are able to give slightly better results compared to LRG. However, LRG gives a slightly better result for the census data set. The latter might be caused by the relatively small number of levels of the categorical variables (maximum 52). Finally, our four strategies LRFM1, LRFM2, LRFM3, LRFM4 lead to different results on different data sets, without one

strategy being the best for all data sets. The only exception is that in some cases the result of LRFM2 equals LRFM1 or the result of LRFM4 equals LRFM3. This is the case when no interaction term exceeded the significance level for entering the modeling equation. Therefore, we suggest to explore all four strategies in a practical problem setting.

The results of this chapter show that Factorization Machines can be successfully combined with Logistic Regression to overcome problems with categorical variables with many levels. This will lead to better performing (risk) models. We think that our methods can be adjusted without much effort to allow inclusion of categorical variables with many levels in other classification algorithms that suffer from a sparse, ill-conditioned model matrix, like other Generalized Linear Models or Support Vector Machines. Also a generalization to a multinomial logistic regression is straightforward. Note that some well-known classification algorithms have problems with categorical variables with many levels. For example, the standard implementation in R of RandomForest [64] accepts only categorical variables with at most 53 levels.

Further research can address the issue of explainability of a Factorization Machine. Although our strategies lead to relatively simple logistic regression models, the introduction of the Factorization Machines worsens the explainability for that part of the model. We think that this problem can be solved by applying various dimensionality reduction and visualization techniques to the matrix $\mathbf{V}$ that consists of vectors $\mathbf{v}$ that represent levels of categorical variables, [107]. One could experiment as well with using an L^1 norm as a regularization term in the Factorization Machine.

Finally, we mention that some categorical variables with many values, like zip codes, may have a relation with ethnicity. For example people with the same ethnic background might be clustered in certain zip codes. Since the NTCA wants to avoid ethnic profiling, it wants to investigate this issue prior to including zip codes in a risk model.