FEATURE QUANTIZATION FOR PARSIMONIOUS AND MEANINGFUL PREDICTIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

For regulatory and interpretability reasons, the logistic regression is still widely used by financial institutions to learn the refunding probability of a loan given the applicant's characteristics from historical data. Although logistic regression handles naturally both continuous and categorical data, a preprocessing step to quantize them is usually performed for improving simultaneously prediction accuracy and user interpretability: continuous features are discretized by assigning factor levels to intervals; some levels of categorical features (with numerous levels) are grouped. However, a better predictive accuracy can be reached by embedding this quantization estimation step directly into the predictive estimation step itself. A related information criterion has then to be optimized on a huge and untractable discontinuous quantization set, requiring to introduce a specific two-step optimization strategy: first, the optimization problem is relaxed in order to deal with smooth functions; second, a particular neural network is involved through a stochastic gradient algorithm to optimize the resulting criterion, giving access to good candidates for the initial optimization problem. The good performances of this approach are illustrated on simulated and real data from Crédit Agricole Consumer Finance (a major European historic player in the consumer credit market).

1 MOTIVATION

As stated by Hosmer Jr et al. (2013), in many applicative contexts (credit scoring, biostatistics, *etc.*), logistic regression is widely used for its simplicity, decent performance and interpretability in predicting a binary outcome given predictors of different types (categorical, continuous). However, to achieve even higher interpretability, continuous predictors are sometimes discretized so as to produce a "scorecard", *i.e.* a table assigning "points" to an applicant in credit sco (or a patient, *etc.*) depending on its predictors being in a given interval. Moreover, logistic regression makes the restrictive assumption of the linearity of the log-odd ratio of the target variable given a continuous feature, which might lead to a (possibly large) modeling bias. Another important argument for discretization is that it overcomes this limit by approximating the true relationship with a step function (where the number of steps and their sizes are determined by the discretization intervals). However, Yang & Webb (2009) show in the finite sample setting that the resulting increase of the number of parameters in predictive models can lead to an increase in variance (overfitting). Likewise when dealing with categorical features which take numerous levels, their respective regression coefficient suffer from high variance. A straightforward solution formalized by Maj-Kańska et al. (2015) is to merge their factor levels which leads to less coefficients and therefore less variance. Designating both discretization (for continuous features) and level groupong (for categorical ones) by the generic term of quantization, an important question is thus to estimate this latter

It can be claimed that discretization and grouping are a special case of representation learning for logistic regression where the input space $\mathbb{R}^d$ is partitioned in rectangles along the axes. Discretization and grouping can also be seen as a variable selection tool, as discretizing a continuous feature into a single interval or grouping factor levels to a single level amounts to discard the feature from the logistic regression model. In the next section, we formalize discretization and grouping then introduce a model selection criterion to select the best representation of the data. In section 3, we introduce modeling assumptions to facilitate the search of the best model w.r.t. the aforementioned criterion. In section 3.2, we introduce a particular neural network architecture that will allow to efficiently estimate the model introduced in the preceding section. Section **??** discusses specifically

perspectives on how to add to the proposed algorithm the ability to introduce well-chosen interactions. The last section is dedicated to numerical experiments on both simulated and real data from the field of *Credit Scoring*.

2 QUANTIZATION AS A COMBINATORIAL CHALLENGE

2.1 QUANTIZATION DEFINITION

Quantization procedure consists in turning a d dimensional raw vector

$$\boldsymbol{x} = (x_1, \ldots, x_d) \in \boldsymbol{\mathcal{X}} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$$

into a d dimensional categorical vector (each $\mathcal{X}_j^{f_j}$ have below m_j levels)

$$\boldsymbol{x}^{\boldsymbol{f}} = (x_1^{f_1}, \ldots, x_d^{f_d}) \in \boldsymbol{\mathcal{X}}^{\boldsymbol{f}} = \mathcal{X}_1^{f_1} \times \ldots \times \mathcal{X}_d^{f_d},$$

through the component wise mapping $\boldsymbol{f}(\boldsymbol{x}) = (f_j(x_j))_1^d$, with the shortcuts $\boldsymbol{x}^{\boldsymbol{f}} = \boldsymbol{f}(\boldsymbol{x})$ and $x_j^{f_j} = f_j(x^j)$. Here the last term is an integer which designates the level of the corresponding categorical feature. Moreover, above $j = 1, \ldots, d$ and it will be hereafter the implicit range of the index j. However, raw data gather possibly both continuous and categorical features, leading to a quite different quantization definition. The resulting whole space where $\boldsymbol{f}$ lies is denoted by $\mathcal{F}$.

Raw continuous features If x_j is a continuous component of $\boldsymbol{x}$, quantization f_j consists to perform a discretization of x_j, namely f_j is selected from the family of step functions of the form

$$f_j(\cdot) = \sum_{h=1}^{m_j} \mathbb{1}_{(c_{j,h-1}, c_{j,h})}(\cdot)$$

where m_j ranges over $\mathbb{N}$, $c_{j,1}, \ldots, c_{j,m-1}$ are increasing numbers called cutpoints, $c_{j,0} = -\infty$ and $c_{j,m} = \infty$.

Raw categorical features If x_j is a categorical component of $\boldsymbol{x}$, quantization f_j consists in grouping levels of x_j, namely f_j has the form

$$f_j(\cdot) = \sum_{h=1}^{m_j} h \mathbb{1}_{\mathbb{O}_h}(\cdot)$$

where $(\mathbb{O}_h)_1^{m_j}$ is a partition of the set of levels of x_j.

2.2 QUANTIZATION EMBEDDED IN A PREDICTIVE PROCESS

Quantization is a widespread preprocessing step to perform a learning task consisting to predict, say, a binary variable $y \in \{0, 1\}$, from a quantized predictor set $\boldsymbol{x}^{\boldsymbol{f}} \in \boldsymbol{\mathcal{X}}^{\boldsymbol{f}}$, trough, say, a parametric conditional distribution $p_{\boldsymbol{\theta}}(y|\boldsymbol{x}^{\boldsymbol{f}})$. Interest of considering quantized data instead of raw data is twofold. First, at the estimation level from a data set $(\mathbf{x}, \mathbf{y})$, with $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\boldsymbol{n}})$ and $\mathbf{y} = (y_1, \ldots, y_n)$, levels m_j act as a tuning parameter for controlling parsimony and thus the bias/variance trade-off of the estimate of the parameter $\boldsymbol{\theta}$ (or of its predictive accuracy). This claim becomes clearer with the example of logistic regression on which we focus on since it is a widespread method for many practitioners still nowadays, as in the credit scoring real application that will illustrate this work in its numerical part. Logistic regression is classically expressed by

$$\ln\left(\frac{p_{\boldsymbol{\theta}}(1|\boldsymbol{x}^{\boldsymbol{f}})}{1 - p_{\boldsymbol{\theta}}(1|\boldsymbol{x}^{\boldsymbol{f}})}\right) = \theta_0 + \sum_{j=1}^{d} \theta_j^{x_j^{f_j}},$$

where $\boldsymbol{\theta} = (\theta_0, \theta_1^1, \ldots, \theta_1^{m_1}, \ldots, \theta_d^1, \ldots, \theta_d^{m_d}) \in \boldsymbol{\Theta}^{\boldsymbol{f}}$ and where $\boldsymbol{\Theta}^{\boldsymbol{f}}$ corresponds to the space $\mathbb{R}^{\sum_j m_j + 1}$ restricted as follows to guarantee the identifiability of the model: for all features j, m_j is set as the "reference" value and consequently for all j, $\theta_j^{m_j} = 0$. Second, at the practitioner level,

the previous tuning of m_js, especially when it is quite low, allows to have an easier interpretation of the most important predictor values involved in the predictive process.

However, the current practice of quantization is to fix it previously to any predictive task, thus ignoring its consequence on the final predictive ability. Traditional quantization estimation consists to optimize heuristic criteria, all unrelated at least explicitly to prediction, as for instance to limit feature correlations, *etc.* Since also quantization involves natively a huge space $\mathcal{F}$ (it typically involves combinatorics for categorical raw features for instance), it leads to a practical intractable optimization problem of the retained criterion, and thus a set of heuristic algorithms has been also designed for this task. Ramírez-Gallego et al. (2016) gives a review of approximatively 200 quantization strategies, merging both criteria and related algorithms.

Alternatively, it makes sense to focus quantization on the predictive task in order to expect an improvement of this latter. Denoting by $\hat{\boldsymbol{\theta}}^{\boldsymbol{f}}$ the maximum likelihood of $\boldsymbol{\theta}$ from the sample $(\mathbf{x}, \mathbf{y})$ under the standard assumption of mutual independence between the y_is conditionally to the $\boldsymbol{x}_i$s, and denoting by

$$\ell(\boldsymbol{\theta}; \mathbf{y}|\mathbf{x}^{\boldsymbol{f}}) = \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i|\boldsymbol{x}_i^{\boldsymbol{f}})$$

the related log-likelihood, some standard information criteria can be used to select $\boldsymbol{f}$, as the BIC criterion (Schwarz (1978)) defined by

$$\text{BIC}^{\boldsymbol{f}} = \ell(\hat{\boldsymbol{\theta}}^{\boldsymbol{f}}; \mathbf{y}|\mathbf{x}^{\boldsymbol{f}}) - 0.5\nu^{\boldsymbol{f}} \ln n, \tag{1}$$

with $\nu^{\boldsymbol{f}}$ the number of continuous parameters to estimated in the parameter space $\boldsymbol{\Theta}^{\boldsymbol{f}}$, and this criterion to be maximized as follows:

$$\hat{\boldsymbol{f}} = \arg\max_{\boldsymbol{f} \in \mathcal{F}} \text{BIC}^{\boldsymbol{f}}. \tag{2}$$

Many other information criteria as AIC (Akaike (1973)) are also available.

Note at this step that an exhaustive browse of $\mathcal{F}$ is an untractable task due to its highly combinatorial nature. Even the continuous part of $\mathcal{F}$ is combinatorial, this property being hidden behind its apparent continuous nature. Indeed, the continuous part of $\mathcal{F}$ is not uniquely defined since all cutpoints between two successive raw data points of a given feature produce the same resulting discretized data points. Thus, conditionally to the data set $\mathbf{x}$, identifiability is obtained by imposing the continuous part of the set $\mathcal{F}$ to fix arbitrary cutpoint values between successive data points, feature by feature. As a consequence, the continuous part of the set $\mathcal{F}$ becomes a discrete set of values, as just mentioned.

Consequently, optimizing (1) requires a new specific strategy that we describe in the next section.

3 THE PROPOSED NEURAL NETWORK BASED QUANTIZATION

3.1 A RELAXATION OF THE OPTIMIZATION PROBLEM

In this section, we propose to relax the constraints on f_j which will simplify the search for $\hat{\boldsymbol{f}}$. The deterministic discretization and grouping scheme defined in the previous section yielded equation **??** which is clearly, for continuous features, a gate function and for categorical features, a dirac function, which derivatives are 0 almost everywhere. Consequently, finding the optimal discretization cannot be directly achieved using gradient descent, and a greedy search is needed, which is intractable according to the previous section.

A classical approach is to approximate the optimization problem by replacing the gate and dirac functions by smooth functions as it is done in the neural networks literature (Hahnloser & Seung (2001)). By making such a relaxation, f_j is now replaced by the fuzzy quantization $\tilde{\boldsymbol{f}}_j = (\tilde{f}_{j1}, \ldots, \tilde{f}_{jm_j})$ with $\tilde{\boldsymbol{f}}_j(\cdot) \in \mathcal{S}^{m_j}$ where $\mathcal{S}^{m_j}$ is the simplex m_j-dimensional simplex, *i.e.* $\sum_{h=1}^{m_j} \tilde{f}_{jh} = 1$. A parametric model for $\tilde{\boldsymbol{f}}_j$ is defined as follows:

For continuous features $\tilde{f}_{jh}(x_j; \boldsymbol{\alpha}_j) \propto \exp(\alpha_{0,j}^h + \alpha_{1,j}^h x_j)$, with $\boldsymbol{\alpha}_j^h = (\alpha_{0,j}^h, \alpha_{1,j}^h) \in \mathbb{R}^2$, $\boldsymbol{\alpha}_j = (\boldsymbol{\alpha}_j^1, \ldots, \boldsymbol{\alpha}_j^{m_j})$, and setting $\boldsymbol{\alpha}_j^{m_j} = (0, 0)$ for identifiabilty reasons.

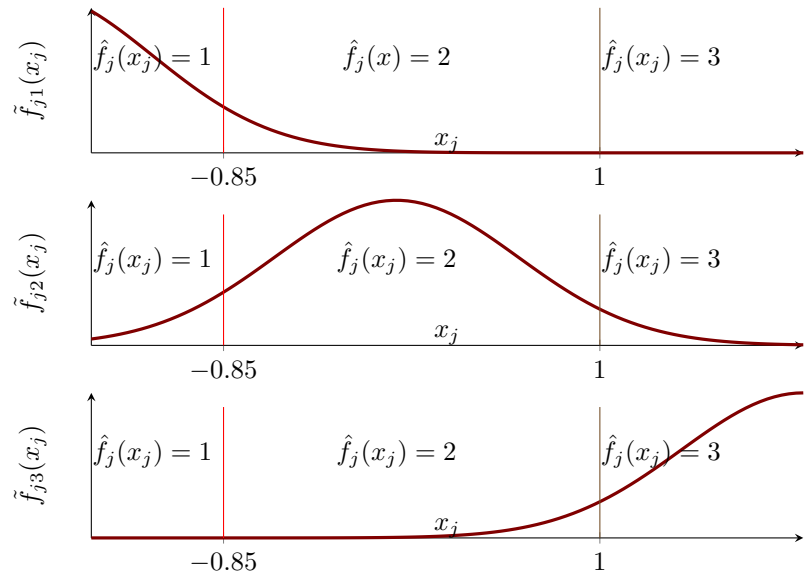

Figure 1: Quantization resulting from the hard thresholding of $\tilde{\boldsymbol{f}}_j$.

For categorical features $\tilde{f}_{jh}(x_j;\boldsymbol{\alpha}_j) \propto \exp(\alpha^h_{x_j,j})$, with $\boldsymbol{\alpha}^h_j = (\alpha^h_{1,j},\ldots,\alpha^h_{o_j,j}) \in \mathbb{R}^{m_j}$,$\boldsymbol{\alpha}_j = (\boldsymbol{\alpha}^1_j,\ldots,\boldsymbol{\alpha}^{m_j}_j)$, and setting $\forall\, h \in \{1,\ldots,o_i\}$ $\boldsymbol{\alpha}^{m_j}_{h,j} = 0$ for identifiabilty reasons.

Finally, we denote by $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_d)$ the parameters associated to $\tilde{\boldsymbol{f}}$, latter denoted by $\tilde{\boldsymbol{f}}(\cdot;\boldsymbol{\alpha})$.

The logistic regression on the representation on $\tilde{\boldsymbol{f}}$ is now

$$\ln\left(\frac{p_{\boldsymbol{\theta}}(1|\tilde{\boldsymbol{f}}(\boldsymbol{x}))}{1-p_{\boldsymbol{\theta}}(1|\tilde{\boldsymbol{f}}(\boldsymbol{x}))}\right) = \theta_0 + \sum_{j=1}^{d}\sum_{h=1}^{m_j}\theta^h_j\,\tilde{f}_{jh}(x_j),$$

where $\boldsymbol{\theta} = (\theta_0,\theta^1_1,\ldots,\theta^{m_1}_1,\ldots,\theta^1_d,\ldots,\theta^{m_d}_d) \in \boldsymbol{\Theta}_{\boldsymbol{f}}$. It is important here to notice that the presented relaxation include the hard partitioning presented in Section as a particular case, *i*.e. where $\tilde{\boldsymbol{f}}_j(\cdot)$ is on the border of the simplex. The predictive likelihood associated to this quantization model is thus

$$\ell(\boldsymbol{\theta},\boldsymbol{\alpha};\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{n}\ln p_{\boldsymbol{\theta}}(y_i|\tilde{\boldsymbol{f}}(\boldsymbol{x}_i;\boldsymbol{\alpha})).$$

The optimization process of is discussed in next section $\ell(\boldsymbol{\theta},\boldsymbol{\alpha};\mathbf{y}|\mathbf{x})$. It is also discussed how this optimization process can be used to generate good candidates for the optimization of the initial criterion, *e*.g. the optimization of BIC$^{\tilde{\boldsymbol{f}}}$ defined in Equation equation 1.

3.2 A NEURAL NETWORK-BASED ESTIMATION STRATEGY

To estimate parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$, a particular neural network architecture can be used. The most obvious part is the output layer that must produce $p_{\boldsymbol{\theta}}(y|\tilde{\boldsymbol{f}}(\boldsymbol{x}_i;\boldsymbol{\alpha}))$ which is equivalent to a densely connected layer with a sigmoid activation.

For a continuous feature x_j, the combined use of m_j neurons with a bias term $((\alpha^h_{0,j},\alpha^h_{1,j})^{m_j}_1)$ and a softmax layer obviously yields $\tilde{\boldsymbol{f}}_j$. Similarly, an input categorical feature x_j with o_j levels is equivalent to o_j binary input neurons (presence or absence of the factor level). These o_j neurons are densely connected to m_j neurons without a bias term and a softmax layer. All in all, it can be remarked that the proposed model is straightforward to optimize with very simple neural networks, as is shown on figure 2.

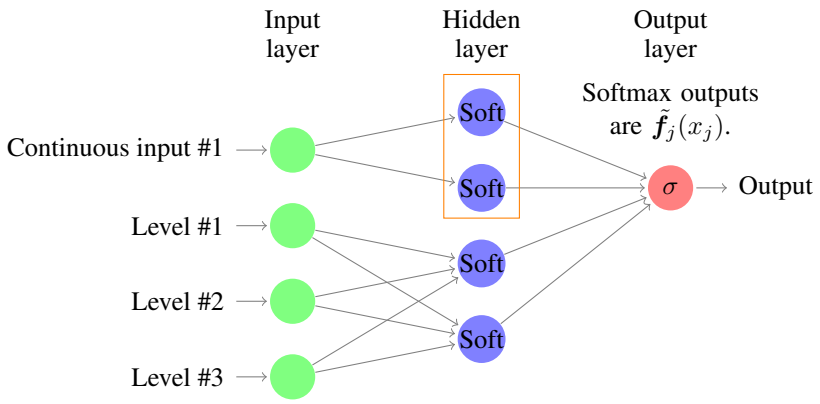

Figure 2: Proposed shallow architecture to maximize $\ell(\boldsymbol{\theta}, \boldsymbol{\alpha}; \mathbf{y}|\mathbf{x})$.

Another meaningful approach would be to adapt stochastic binary units introduced by Bengio (2013) to the multinomial case. In short, these units would draw at each training epoch (t) representations e_{ij} according to a multinomial distribution of parameters $(\tilde{f}_{j1}(x_j; \boldsymbol{\alpha}_j^{(t)}), \ldots, \tilde{f}_{jm_j}(x_j; \boldsymbol{\alpha}_j^{(t)})$.

On a side note, in section 3, we fixed the number of intervals or factor levels to $\boldsymbol{m} = (m_j)_1^d$ and as seen in section 3.1, looping over all candidates $\boldsymbol{m}$ would be intractable. But by setting $\hat{f}_j(x_j) = \arg\max_{h \in \{1,\ldots,m_j\}} \tilde{f}_{jh}(x_j)$, we might drop a lot of unseen factor levels, *e.g.* if $\tilde{f}_{jh}(x_j) << 1$ for all training observations $x_{i,j}$, level h "disappears". In practice, we recommend to start with a user-chosen maximum number of levels $\boldsymbol{m}_{\max}$ and we will see in the experiments of section 4 that the proposed approach is able to explore smaller values of $\boldsymbol{m}$ and even choose a value $\hat{\boldsymbol{m}}$ drastically smaller than $\boldsymbol{m}_{\max}$.

Relying on stochastic gradient descent, the likelihood $\ell(\boldsymbol{\theta}, \boldsymbol{\alpha}; \mathbf{y}|\mathbf{x})$ is minimized such that $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\alpha}}) = \arg\min_{\boldsymbol{\theta}, \boldsymbol{\alpha}} \ell(\boldsymbol{\theta}, \boldsymbol{\alpha}; \mathbf{y}|\mathbf{x})$. If it exists a true underlying quantization the obtained results should be close from the optimization of the rough likelihood $\ell(\boldsymbol{\theta}; \mathbf{y}|\mathbf{x}^{\boldsymbol{f}})$. In the misspecified model case, there is no such guarantee. Therefore, we evaluate $f_j^{(t)}(x_j) = \arg\max_{1 \leq h \leq m_j} \tilde{f}_{jh}(x_j; \boldsymbol{\alpha}_j^{(t)})$ at each training epoch (t), $\boldsymbol{\theta}^{(t)} = \arg\min_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{y}|\mathbf{x}^{\boldsymbol{f}^{(t)}})$ and the resulting $\text{BIC}^{\boldsymbol{f}^{(t)}}$. The discretization retained at the end being $\arg\min_{t \in \{1,\ldots,T\}} \text{BIC}^{\boldsymbol{f}^{(t)}}$.

4 Numerical Experiments

This section is divided into two complementary parts to assess the validity of our proposal, that we call hereafter *glmdisc*. First, simulated data are used to evaluate its ability to recover the true data generating mechanism, including its ability to be robust to non-predictive features. Second, the predictive quality of the new learned representation approach is illustrated on several *Credit Scoring* data sets provided by Crédit Agricole Consumer Finance, a major European company in the consumer credit market. The Python Notebooks of all experiments, excluding the confidential real data, can be found on the first author's website.

4.1 Simulated data: empirical consistency and robustness

Focus is here given on discretization of continuous features, although similar experiments could be conducted on categorical ones. Two continuous features x_1 and x_2 are sampled from the uniform distribution $\mathcal{U}([0,1])$ and discretized using $f(x) = \mathbb{1}_{]-\infty;1/3]}(x) + 2 \times \mathbb{1}_{]1/3;2/3]}(x) + 3 \times \mathbb{1}_{]2/3,\infty[}(x)$. The target feature y is then sampled following a logistic regression $p_{\boldsymbol{\theta}}(\cdot|x^{\boldsymbol{f}})$ where $\boldsymbol{\theta} = (0, -2, 2, 0, -2, 2, 0)$.

Sample size	(a)	(b)	(c)
$n = 1,000$	$[0.642, 0.752]$	$[2.601, 2.944]$	$[1.296, 1.571]$
$n = 10,000$	$[0.656, 0.681]$	$[2.694, 3.056]$	$[1.412, 1.723]$

Table 1: (a) Confidence interval (95%) of $\hat{c}_2$ around $c_2 = 2/3$. (b) Confidence interval (95%) of $\hat{m}$ around $m_1 = m_2 = 3$. (c) Confidence interval (95%) of $\hat{m}_3$ around $m_3 = 1$.

Portfolio	Additive linear logistic regression	Current performance	*ad hoc* methods	*glmdisc*
Automobile loans	59	57	55	56
Renovation loans	52	51	50	54
Revolving loans	62	58	59	59

Table 2: Gini indices of our proposed representation learning algorithm *glmdisc* and two baselines: a "naïve" logistic regression and the current scorecard (manual / expert representation) obtained on several portfolios of CACF.

First, the starting maximum number of intervals per discretized continuous feature is set to its true value $m = 3$ and the quality of the cutoff estimator $\hat{c}_2$ of $c_2 = 2/3$ resulting from the *glmdisc* algorithm can be assessed with a varying sample size from Table 1, Column (a).

Second, the starting maximum number of intervals per discretized continuous feature is set to $m_{\text{max}} = 10$ and Table 1, Column (b), shows the 95% confidence interval of the estimated number of intervals $\hat{m}$ of $m = 3$ resulting from the *glmdisc* algorithm with a varying sample size. The slight underestimation is a classical con of the BIC criterion on small samples.

Last, it can be remarked that if a non-predictive feature gets discretized or grouped into a single value, it is *de facto* excluded from the model. Feature selection is thus a possible positive side effect. Therefore, we add a third feature x_3 also drawn from $\mathcal{U}([0; 1])$ but uncorrelated to y and display on Table 1, Column (c), the 95% confidence interval number of intervals $\hat{m}_3$ chosen for x_3 (note that $m_3 = 1$) with a varying sample size.

4.2 *Credit Scoring* DATA

Discretization, grouping and interaction screening are preprocessing steps relatively "manually" performed in the field of *Credit Scoring*, using χ^2 tests for each feature or so-called Weights of Evidence (Zeng (2014)). This back and forth takes a lot of time and effort and provides no particular statistical guarantee.

The performance metric usually monitored by *Credit Scoring* practitioners is the Gini coefficient, which is directly related to the Area Under the ROC Curve (Gini $= 2\times$AUC-1). Table 2 shows Gini coefficients of several portfolios for which there are $n = 50,000$, $n = 30,000$ and $n = 100,000$ clients respectively and $d = 18$, $d = 12$ and $d = 12$ features respectively. Approximately half of these features were categorical, with a number of factor levels ranging from 2 to 100.

We compare the rather manual, in-house approach that yields the current performance to a naïve linear logistic regression, a logistic regression on continuous discretized data using the now standard MDLP algorithm from Fayyad & Irani (1993) and categorical grouped data using χ^2 tests and *glmdisc*. Beside the classification performance, interpretability is maintained and unsurprisingly, the learned representation comes often close to the "manual" approach: for example, the complicated in-house coding of job types is roughly grouped by *glmdisc* into *e.g.* "worker", "technician", *etc.*

The usefulness of discretization, grouping and interactions is clear on *Credit Scoring* data and although *glmdisc* does not always perform significantly better than the manual approach, it allows practitioners to focus on other tasks by saving a lot of time, as was already stressed out. As a rule of thumb, a month is generally allocated to data pre-processing for a single data scientist working on a single scorecard. On Google Collaboratory, and relying on Keras (Chollet et al. (2015)) and Tensorflow (Abadi et al. (2015)) as a backend, it took less than an hour to perform discretization and grouping for all datasets.

5 CONCLUDING REMARKS

The essentially industrial problem of discretization, grouping of factor levels and introduction of interactions in a supervised multivariate classification setting was formalized as a highly combinatorial representation learning problem and a new approach, named *glmdisc*, has been proposed.

This algorithm relies on the use of polytomous logistic links between each discretized or grouped feature and their respective continuous or categorical version. These proposals can be replaced by any other univariate multiclass predictive model, which makes it flexible and adaptable to other problems. Prediction of the target feature, given engineered features, was required to stem from a logistic regression, although here as well, it can be swapped with any other supervised classification model. An estimation strategy putting neural networks' good computational properties to use was introduced while maintaining the interpretability necessary to some fields of application.

The experiments showed that, as was sensed empirically by statisticians in the field of *Credit Scoring*, discretization and grouping can indeed provide better models than standard logistic regression. This novel approach allows practitioners to have a fully automized and statistically well-grounded tool that achieves better performance than *ad hoc* industrial practices at the price of decent computing time but much less of the practitioner's valuable time.

As described in the introduction, logistic regression is additive in its inputs which does not allow to take into account conditional dependency, as stated by Berry et al. (2010). This problem is often dealt with by sparsely introducing "interactions", *i.e.* products of two features. This leads again to a model selection challenge on a highly combinatorial discrete space that could be solved with a similar approach. In a broader context with no restriction on the predictive model, Tsang et al. (2018) already made use of neural networks to estimate the presence or absence of statistical interactions. The parsimonious addition of pairwise interactions among quantized features, that might influence the quantization process introduced in this work, is a future area of research.

ACKNOWLEDGMENTS

The authors are thankful to Crédit Agricole Consumer Finance for providing data and funding through a CIFRE PhD, made possible by the Association Nationale Recherche Technology (ANRT). Many thanks also go to Pascal Germain for his useful insights on the architecture and optimization procedures of neural networks.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory, 1973*, pp. 267–281. Akademiai Kiado, 1973.

Y. Bengio. Estimating or Propagating Gradients Through Stochastic Neurons. *ArXiv e-prints*, May 2013.

William D Berry, Jacqueline HR DeMeritt, and Justin Esarey. Testing for interaction in binary logit and probit models: Is a product term essential? *American Journal of Political Science*, 54(1): 248–266, 2010.

François Chollet et al. Keras. `https://keras.io`, 2015.

Usama Fayyad and Keki Irani. Multi-interval discretization of continuous-valued attributes for classification learning. 1993.

Richard HR Hahnloser and H Sebastian Seung. Permitted and forbidden sets in symmetric threshold-linear networks. In *Advances in Neural Information Processing Systems*, pp. 217–223, 2001.

David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

Aleksandra Maj-Kańska, Piotr Pokarowski, Agnieszka Prochenka, et al. Delete or merge regressors for linear model selection. *Electronic Journal of Statistics*, 9(2):1749–1778, 2015.

Sergio Ramírez-Gallego, Salvador García, Héctor Mouriño-Talín, David Martínez-Rego, Verónica Bolón-Canedo, Amparo Alonso-Betanzos, José Manuel Benítez, and Francisco Herrera. Data discretization: taxonomy and big data challenge. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(1):5–21, 2016.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978. ISSN 00905364. URL http://www.jstor.org/stable/2958889.

Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ByOfBggRZ.

Ying Yang and Geoffrey I Webb. Discretization for naive-bayes learning: managing discretization bias and variance. *Machine learning*, 74(1):39–74, 2009.

Guoping Zeng. A necessary condition for a good binning algorithm in credit scoring. *Applied Mathematical Sciences*, 8(65):3229–3242, 2014.