# Feature quantization for parsimonious and interpretable predictive models

Adrien Ehrhardt, Christophe Biernacki, Vincent Vandewalle, Philippe Heinrich

# Feature quantization for parsimonious and interpretable predictive models

Adrien Ehrhardt [1 2] Christophe Biernacki [2] Vincent Vandewalle [2 3] Philippe Heinrich [4]

## Abstract

For regulatory and interpretability reasons, logistic regression is still widely used. To improve prediction accuracy and interpretability, a preprocessing step quantizing both continuous and categorical data is usually performed: continuous features are discretized and, if numerous, levels of categorical features are grouped. An even better predictive accuracy can be reached by embedding this quantization estimation step directly into the predictive estimation step itself. But doing so, the predictive loss has to be optimized on a huge set. To overcome this difficulty, we introduce a specific two-step optimization strategy: first, the optimization problem is relaxed by approximating discontinuous quantization functions by smooth functions; second, the resulting relaxed optimization problem is solved *via* a particular neural network. The good performances of this approach, which we call *glmdisc*, are illustrated on simulated and real data from the UCI library and Crédit Agricole Consumer Finance (a major European historic player in the consumer credit market).

## 1. Motivation

As stated by Hosmer et al. (2013), in many application 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 higher interpretability, continuous predictors are sometimes discretized so as to produce a "scorecard", *i.e.* a table assigning a grade to an applicant in credit scoring (or a patient in biostatistics, *etc.*) depending on its predictors being in a given interval. Discretization is also an opportunity for reducing the (possibly large) modeling bias which can appear in logistic regression as a result of the linearity assumption on the continuous predictors in the model. Indeed, this restriction can be overcome by approximating the true predictive mapping with a step function where the tuning of the steps and of their sizes allows more flexibility. However, the resulting increase of the number of parameters can lead to an increase of their variance (overfitting) as shown by Yang & Webb (2009). Thus, a precise tuning of the discretization procedure is required. Likewise when dealing with categorical features which take numerous levels, their respective regression coefficients suffer from this high variance phenomenon. 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.

From now on, the generic term quantization will stand for both discretization of continuous features and level grouping of categorical ones. Its aim is to improve the prediction accuracy but it suffers from yielding a highly combinatorial optimization problem whatever the predictive criterion used to select the best quantization. The present work proposes a strategy to overcome these combinatorial issues by invoking a relaxed alternative of the initial quantization problem leading to a simpler estimation problem since it can be easily optimized by a specific neural network. This relaxed version serves as a plausible quantization provider related to the initial criterion after a classical thresholding (*maximum a posteriori*) procedure.

The outline of this work is the following. In the next section, we formalize both continuous and categorical quantization. Selecting the best quantization in a predictive setting is reformulated as a model selection problem on a huge discrete space. In Section 3, a particular neural network architecture is used to optimize a relaxed version of this criterion and propose good quantization candidates. Section 4 is dedicated to numerical experiments on both simulated and real data from the field of Credit Scoring, highlightening the good results offered by the use of this new method without any human intervention. A final section concludes the work by stating also new challenges.

[1] Crédit Agricole Consumer Finance, Roubaix, France [2] Inria, Villeneuve d'Ascq, France [3] EA2694 Santé publique: épidémiologie et qualité des soins, Univ. Lille, Lille, France [4] UMR 8524 Laboratoire Paul Painlevé, Univ. Lille, Lille, France. Correspondence to: Adrien Ehrhardt <adrien.ehrhardt@inria.fr>.

## 2. Quantization as a combinatorial challenge

### 2.1. Quantization: definition

The quantization procedure consists in turning a $d$-dimensional raw vector of continuous and/or categorical features $\boldsymbol{x} = (x_1, \ldots, x_d)$ into a $d$-dimensional categorical vector via a component wise mapping $\boldsymbol{q} = (\boldsymbol{q}_j)_1^d$:

$$\boldsymbol{q}(\boldsymbol{x}) = (\boldsymbol{q}_1(x_1), \ldots, \boldsymbol{q}_d(x_d)),$$

where each of the $\boldsymbol{q}_j$'s is a vector of $m_j$ dummies:

$$q_{j,h}(\cdot) = 1 \text{ if } x_j \in C_{j,h}, 0 \text{ otherwise}, 1 \leq h \leq m_j, \quad (1)$$

where $m_j$ is an integer and the sets $C_{j,h}$ are defined with respect to each feature type as we describe just below.

#### 2.1.1. Raw continuous features case

If $x_j$ is a continuous component of $\boldsymbol{x}$, quantization $\boldsymbol{q}_j$ has to perform a discretization of $x_j$ and the $C_{j,h}$s, $1 \leq h \leq m_j$, are contiguous intervals

$$C_{j,h} = (c_{j,h-1}, c_{j,h}] \quad (2)$$

where $c_{j,1}, \ldots, c_{j,m_j-1}$ are increasing numbers called cutpoints, $c_{j,0} = -\infty$ and $c_{j,m_j} = +\infty$.

For example, the quantization of the unit segment in thirds would be defined as $m_j = 3$, $c_{j,1} = 1/3$, $c_{j,2} = 2/3$ and subsequently $\boldsymbol{q}_j(0.1) = (1, 0, 0)$.

#### 2.1.2. Raw categorical features case

If $x_j$ is a categorical component of $\boldsymbol{x}$, quantization $\boldsymbol{q}_j$ consists in grouping levels of $x_j$ and the $C_{j,h}$s form a partition of the set, say $\{1, \ldots, l_j\}$, of levels of $x_j$:

$$\bigsqcup_{h=1}^{m_j} C_{j,h} = \{1, \ldots, l_j\}.$$

For example, the grouping of levels encoded as "1" and "2" would yield $C_{j,1} = \{1, 2\}$ such that $\boldsymbol{q}_j(1) = \boldsymbol{q}_j(2) = (1, 0, \ldots, 0)$.

#### 2.1.3. Notations for the quantization family

In both continuous and categorical cases, keep in mind that $m_j$ is the dimension of $\boldsymbol{q}_j$. For notational convenience, the (global) order of the quantization $\boldsymbol{q}$ is set as

$$|\boldsymbol{q}| = \sum_{j=1}^{d} m_j.$$

The space where quantizations $\boldsymbol{q}$ live (resp. $\boldsymbol{q}_j$) will be denoted by $\mathcal{Q}_{\boldsymbol{m}}$ in the sequel (resp. $\mathcal{Q}_{j,m_j}$), when the number of levels $\boldsymbol{m} = (m_j)_1^d$ is fixed. Since it is not known, the full model space is $\mathcal{Q} = \cup_{\boldsymbol{m} \in \mathbb{N}_\star^d} \mathcal{Q}_{\boldsymbol{m}}$.

#### 2.1.4. Literature review

The current practice of quantization is prior to any predictive task, thus ignoring its consequences on the final predictive ability. It consists in optimizing a heuristic criterion, often either totally unrelated (unsupervised methods) or partially related (supervised methods) to the predictive task, and mostly univariate (each feature is quantized irrespective of other features' values). The cardinality of the quantization space $\mathcal{Q}$ can be calculated explicitely w.r.t. $d$, $(m_j)_1^d$ and, for categorical features, $l_j$. It is huge (see a more precise illustration of this combinatorial challenge in Section 2.2.2), so that a greedy approach is intractable and such heuristics are needed. Many algorithms have thus been designed and a review of approximatively 200 discretization strategies, gathering both criteria and related algorithms, can be found in (Ramírez-Gallego et al., 2016). For factor levels grouping, we found no such taxonomy, but some discretization methods, *e.g.* $\chi^2$ independence test-based methods can be naturally extended to this type of quantization, which is for example what the CHAID algorithm, proposed by Kass (1980) and applied to each categorical feature, relies on.

### 2.2. Quantization embedded in a predictive process

#### 2.2.1. Logistic regression on quantized data

Quantization is a widespread preprocessing step to perform a learning task consisting in predicting, say, a binary variable $y \in \{0, 1\}$, from a quantized predictor $\boldsymbol{q}(\boldsymbol{x})$, through, say, a parametric conditional distribution $p_{\boldsymbol{\theta}}(y|\boldsymbol{q}(\boldsymbol{x}))$ like logistic regression. Considering quantized data instead of raw data has a double benefit. First, the quantization order $|\boldsymbol{q}|$ acts as a tuning parameter for controlling the model's flexibility and thus the bias/variance trade-off of the estimate of the parameter $\boldsymbol{\theta}$ (or of its predictive accuracy) for a given dataset. This claim becomes clearer with the example of logistic regression we focus on, as a still very popular model for many practitioners. It is classically described by

$$\ln\left(\frac{p_{\boldsymbol{\theta}}(1|\boldsymbol{q}(\boldsymbol{x}))}{1 - p_{\boldsymbol{\theta}}(1|\boldsymbol{q}(\boldsymbol{x}))}\right) = \theta_0 + \sum_{j=1}^{d} \boldsymbol{\theta}'_j \cdot \boldsymbol{q}_j(x_j), \quad (3)$$

where $\boldsymbol{\theta} = (\theta_0, (\boldsymbol{\theta}_j)_1^d) \in \mathbb{R}^{|\boldsymbol{q}|+1}$ and $\boldsymbol{\theta}_j = (\theta_j^1, \ldots, \theta_j^{m_j})$ with $\theta_j^{m_j} = 0$, $j = 1 \ldots d$, for identifiability reasons. Second, at the practitioner level, the previous tuning of $|\boldsymbol{q}|$ through each feature's quantization order $m_j$, especially when it is quite low, allows an easier interpretation of the most important predictor values involved in the predictive process. Denoting the dataset by $(\mathbf{x}, \mathbf{y})$, with $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, $\mathbf{y} = (y_1, \ldots, y_n)$ and $n$ the sample size, the log-likelihood

$$\ell_{\boldsymbol{q}}(\boldsymbol{\theta}; (\mathbf{x}, \mathbf{y})) = \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i|\boldsymbol{q}(\boldsymbol{x}_i)) \quad (4)$$

provides a maximum likelihood estimator $\hat{\boldsymbol{\theta}}_{\boldsymbol{q}}$ of $\boldsymbol{\theta}$ for a given quantization $\boldsymbol{q}$. For the rest of the paper, the approach is exemplified with logistic regression as $p_{\boldsymbol{\theta}}$ but it can be applied to any other predictive model, as will be recalled in the concluding Section (5).

### 2.2.2. Quantization as a model selection problem

As dicussed in the previous section, and emphasized in the literature review, quantization is often a preprocessing step; however, quantization can be embedded directly in the predictive model. Continuing our logistic example, a standard information criterion such as the BIC (Schwarz, 1978) can be used to select the best quantization $\boldsymbol{q}$:

$$\begin{aligned}\hat{\boldsymbol{q}} &= \operatorname*{argmin}_{\boldsymbol{q}\in\mathcal{Q}} \text{BIC}(\hat{\boldsymbol{\theta}}_{\boldsymbol{q}}) \qquad (5)\\ &= \operatorname*{argmin}_{\boldsymbol{q}\in\mathcal{Q}} \left\{-2\ell_{\boldsymbol{q}}(\hat{\boldsymbol{\theta}}_{\boldsymbol{q}};(\mathbf{x},\mathbf{y})) + \nu_{\boldsymbol{q}}\ln(n)\right\}\end{aligned}$$

where $\nu_{\boldsymbol{q}}$ is the number of continuous parameters to be estimated in the $\boldsymbol{\theta}$-parameter space. We shall insist here on the fact that choosing the BIC as our model selection tool is unrelated to the proposed algorithm. The practitioner can swap this criterion with any other information criterion on training data such as AIC (Akaike, 1973) or, as *Credit Scoring* people like, the Gini index on a test set. Note however that, regardless of the criterion used, an exhaustive search of $\hat{\boldsymbol{q}} \in \mathcal{Q}$ is an intractable task due to its highly combinatorial nature. For example, with $d = 10$ categorical features with $l_j = 4$ levels each, $|\mathcal{Q}|$ is given by the sum of the Stirling numbers of the second kind over $m_j = 1 \dots l_j$ to the power $d$, which is approximately $6{\cdot}10^{11}$. Anyway, the optimization in (5) requires a new specific strategy, which is the main contribution of the present work, and that we describe in the next section.

### 2.2.3. Remark on model identifiability

The shifting of cutpoints (2) anywhere strictly between two successive raw values of a given continuous feature induce the same quantization. Thus, the identifiability of such quantizations is obtained from the dataset $\mathbf{x}$ by fixing arbitrary cutpoints between successive data values, feature by feature.

# 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 $\boldsymbol{q}_j$ to simplify the search of $\hat{\boldsymbol{q}}$. Indeed, the derivatives of $\boldsymbol{q}_j$ are zero almost everywhere and consequently a gradient descent cannot be directly applied to find an optimal quantization.

### 3.1.1. Smooth approximation of the quantization mapping

A classical approach is to replace the binary functions $q_{j,h}$ (see Equation (1)) by smooth parametric ones with a simplex condition, namely with $\boldsymbol{\alpha}_j = (\boldsymbol{\alpha}_{j,1}, \dots, \boldsymbol{\alpha}_{j,m_j})$:

$$\boldsymbol{q}_{\boldsymbol{\alpha}_j}(\cdot) = \left(q_{\boldsymbol{\alpha}_{j,h}}(\cdot)\right)_{h=1}^{m_j} \text{ with } \begin{cases}\sum_{h=1}^{m_j} q_{\boldsymbol{\alpha}_{j,h}}(\cdot) = 1,\\ 0 \le q_{\boldsymbol{\alpha}_{j,h}}(\cdot) \le 1,\end{cases}$$

where functions $q_{\boldsymbol{\alpha}_{j,h}}(\cdot)$, properly defined hereafter for both continuous and categorical features, represent a fuzzy quantization in that, here, each level $h$ is weighted by $q_{\boldsymbol{\alpha}_{j,h}}(\cdot)$ instead of being selected once and for all as in (1). The resulting fuzzy quantization for all components depends on the global parameter $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_d)$ and is denoted by $\boldsymbol{q}_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \left(\boldsymbol{q}_{\boldsymbol{\alpha}_j}(x_j)\right)_{j=1}^{d} \in \widetilde{\mathcal{Q}}$. This approximation will be justified in Section 3.1.3.

**For continuous features**, we set for $\boldsymbol{\alpha}_{j,h} = (\alpha^0_{j,h}, \alpha^1_{j,h}) \in \mathbb{R}^2$

$$q_{\boldsymbol{\alpha}_{j,h}}(\cdot) = \frac{\exp(\alpha^0_{j,h} + \alpha^1_{j,h}\cdot)}{\sum_{g=1}^{m_j}\exp(\alpha^0_{j,g} + \alpha^1_{j,g}\cdot)}.$$

**For categorical features**, we set for $\boldsymbol{\alpha}_{j,h} = (\alpha_{j,h}(1), \dots, \alpha_{j,h}(l_j)) \in \mathbb{R}^{l_j}$

$$q_{\boldsymbol{\alpha}_{j,h}}(\cdot) = \frac{\exp\left(\alpha_{j,h}(\cdot)\right)}{\sum_{g=1}^{m_j}\exp\left(\alpha_{j,g}(\cdot)\right)}.$$

### 3.1.2. Parameter estimation

With this new fuzzy quantization, the logistic regression for the predictive task is then expressed as

$$\ln\left(\frac{p_{\boldsymbol{\theta}}(1|\boldsymbol{q}_{\boldsymbol{\alpha}}(\boldsymbol{x}))}{1 - p_{\boldsymbol{\theta}}(1|\boldsymbol{q}_{\boldsymbol{\alpha}}(\boldsymbol{x}))}\right) = \theta_0 + \sum_{j=1}^{d} \boldsymbol{\theta}'_j \cdot \boldsymbol{q}_{\boldsymbol{\alpha}_j}(x_j), \qquad (6)$$

where $\boldsymbol{q}$ has been replaced by $\boldsymbol{q}_{\boldsymbol{\alpha}}$ from Equation (3). Note that as $\boldsymbol{q}_{\boldsymbol{\alpha}}$ is a sound approximation of $\boldsymbol{q}$ (see Section 3.1.3), this logistic regression in $\boldsymbol{q}_{\boldsymbol{\alpha}}$ is consequently a good approximation of the logistic regression in $\boldsymbol{q}$ from Equation (3). The relevant log-likelihood is here

$$\ell_{\boldsymbol{q}_{\boldsymbol{\alpha}}}(\boldsymbol{\theta};(\mathbf{x},\mathbf{y})) = \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i|\boldsymbol{q}_{\boldsymbol{\alpha}}(\boldsymbol{x}_i)) \qquad (7)$$

and can be used as a tractable substitute for (4) to solve the original optimization problem (5), where now both $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ have to be estimated, which is discussed in the next section. We wish to maximize the log-likelihood (6) which would yield parameters $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}})$; To "push" $\widetilde{\mathcal{Q}}$ further into $\mathcal{Q}$, we deduce $\boldsymbol{q}^{\text{MAP}}$ from a *maximum a posteriori* procedure applied to $\boldsymbol{q}_{\hat{\boldsymbol{\alpha}}}$:

$$\hat{q}^{\text{MAP}}_{j,h}(x_j) = 1 \text{ if } h = \operatorname*{argmax}_{1\le h'\le m_j} q_{\hat{\boldsymbol{\alpha}}_{j,h'}}, 0 \text{ otherwise.} \qquad (8)$$

If there are two levels $h$ that satisfy (8), we simply take the level that corresponds to smaller values of $x_j$ to be in accordance with the definition of $C_{j,h}$ in Equation (2). This *maximum a posteriori* principle are exemplified in Figure 2 on simulated data by the plain vertical lines (see Section 4).

### 3.1.3. Validity of the relaxation

From a deterministic point of view, we have $\mathcal{Q} \subset \widetilde{\mathcal{Q}}$: First, the *maximum a posteriori* step (8) produces contiguous intervals (*i.e.* there exists $C_{j,h}$; $1 \leq j \leq d, 1 \leq h \leq m_j$, s.t. $\boldsymbol{q}^{\text{MAP}}$ can be written as in 1) (Samé et al., 2011). Second, in the continuous case, the higher $\alpha^1_{j,h}$, the less smooth the transition from one quantization $h$ to its "neighbor"[1] $h+1$, whereas $\frac{\alpha^0_{j,h}}{\alpha^1_{j,h}}$ controls the point in $\mathbb{R}$ where the transition occurs (Chamroukhi et al., 2009). Concerning the categorical case, the rationale is even simpler as $q_{\lambda\alpha_{j,h}}(x_j) \rightarrow 1$ if $h = \text{argmax}_{h'} \, q_{\alpha_{j,h'}}(x_j)$, 0 otherwise as $\lambda \rightarrow +\infty$ (Reverdy & Leonard, 2016).

From a statistical point of view, under standard regularity conditions and with a suitable estimation procedure (see later for the proposed estimation procedure), the maximum likelihood framework ensures the consistency of $(\boldsymbol{q}_{\hat{\boldsymbol{\alpha}}}, \hat{\boldsymbol{\theta}})$ towards $(\boldsymbol{q}, \boldsymbol{\theta})$. This is further ensured by the *maximum a posteriori* step (8).

However, and as is usual, the log-likelihood $\ell_{\boldsymbol{q}_{\boldsymbol{\alpha}}}(\boldsymbol{\theta}, (\mathbf{x}, \mathbf{y}))$ cannot be directly maximized w.r.t. $(\boldsymbol{\alpha}, \boldsymbol{\theta})$, so that we need an iterative procedure. To this end, the next section introduces a neural network of particular architecture.

From an empirical point of view, we will see in Section 4 and in particular in Figure 2, that the smooth approximation $\boldsymbol{q}_{\boldsymbol{\alpha}}$ converges towards "hard" quantizations[1] $\boldsymbol{q}$.

## 3.2. A neural network-based estimation strategy

### 3.2.1. Neural network architecture

To estimate parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ in model (6), a particular neural network architecture can be used. We shall insist that this network is only a way to use common deep learning frameworks, namely Tensorflow (Abadi et al., 2015) through the high-level API Keras (Chollet et al., 2015) instead of building a gradient ascent algorithm from scratch to optimize (7). The most obvious part is the output layer that must produce $p_{\boldsymbol{\theta}}(1|\boldsymbol{q}_{\boldsymbol{\alpha}}(\boldsymbol{x}))$ which is equivalent to a densely connected layer with a sigmoid activation $\sigma(\cdot)$.

For a continuous feature $x_j$ of $\boldsymbol{x}$, the combined use of $m_j$ neurons including affine transformations and softmax activation obviously yields $\boldsymbol{q}_{\boldsymbol{\alpha}_j}(x_j)$. Similarly, an input categorical feature $x_j$ with $l_j$ levels is equivalent to $l_j$ binary input neurons (presence or absence of the factor level). These $l_j$ neurons are densely connected to $m_j$ neurons without any bias term and a softmax activation. The softmax outputs are next aggregated via the summation in model (6), say $\Sigma_{\boldsymbol{\theta}}$ for short, and then the sigmoid function $\sigma$ gives the final output. All in all, the proposed model is straightforward to optimize with a simple neural network, as shown in Figure 1.

### 3.2.2. Stochastic gradient descent as a quantization provider

By relying on stochastic gradient ascent, the smoothed likelihood (7) can be maximized over $(\boldsymbol{\alpha}, \boldsymbol{\theta})$. Due to its convergence properties (Bottou, 2010), the results should be close to the maximizers of the original likelihood (4) if the model is well-specified, when there is a true underlying quantization. However, in the mis-specified model case, there is no such guarantee. Therefore, to be more conservative, we evaluate at each training epoch $(t)$ the quantization $\boldsymbol{q}^{\text{MAP}(t)}$ resulting from the *maximum a posteriori* procedure explicited in Equation (8), then classically estimate the logistic regression parameter *via* maximum likelihood, as done in Equation (4):

$$\hat{\boldsymbol{\theta}}^{(t)} = \underset{\boldsymbol{\theta}}{\text{argmax}} \, \ell_{\boldsymbol{q}^{\text{MAP}(t)}}(\boldsymbol{\theta}; (\mathbf{x}, \mathbf{y}))$$

and the resulting $\text{BIC}(\hat{\boldsymbol{\theta}}^{(t)})$ as in (5). If $T$ is a given maximum number of iterations of the stochastic gradient ascent algorithm, the quantization retained at the end is then determined by the optimal epoch

$$t_* = \underset{t \in \{1, \ldots, T\}}{\text{argmin}} \, \text{BIC}(\hat{\boldsymbol{\theta}}^{(t)}). \tag{9}$$

The number of iterations $T$ can be seen as a computational budget: contrary to classical early stopping rules (*e.g.* based on validation loss) used in neural network fitting, this network only acts as a stochastic quantization provider for (9) which will naturally prevent overfitting. We reiterate that, in (9), the BIC can be swapped for the user's favourite model choice criterion. Lots of optimization algorithms for neural networks have been proposed, which all come with their hyperparameters. We chose the "RMSProp" method, which showed good results, is one of the standard methods, and tuned only its learning rate.

### 3.2.3. Choosing an appropriate number of levels

The number of intervals or factor levels $\boldsymbol{m} = (m_j)_1^d$ were supposed up to now known but in practice also have to be estimated. In fact, they play an overriding role in the bias-variance "tuning" effect which motivated this work in Section 1. By relying on the *maximum a posteriori* procedure developed in Equation (8) parallel to the neural network candidate generator, we might drop a lot of unseen factor levels,

[1] Up to a permutation on the labels $h = 1 \ldots m_j$ to recover the ordering in $C_{j,h}$ (see Equation (2)).

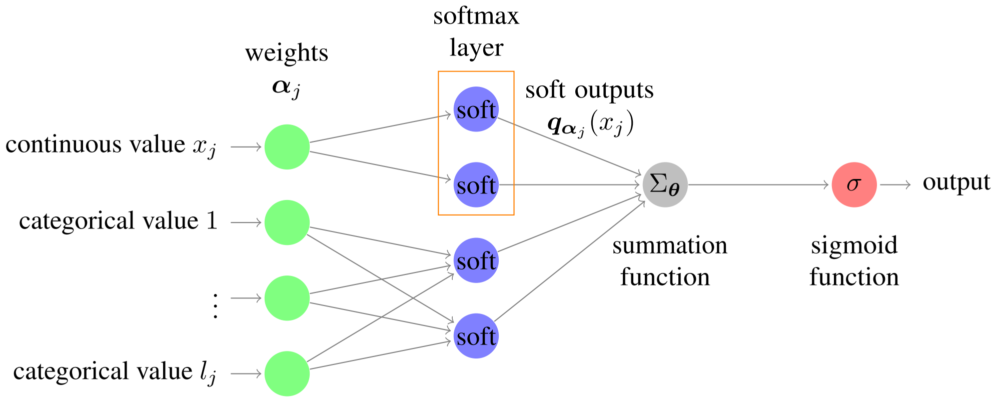


*Figure 1.* Proposed shallow architecture to maximize (7).

*e.g.* if $q_{\boldsymbol{\alpha}_{j,h}}(x_{i,j}) \ll 1$ for all training observations $x_{i,j}$, the level $h$ "vanishes", *i.e.* $\hat{q}_{j,h} = 0$. Thus, it is not necessary to go through such a loop and in practice, we recommend to start with a user-chosen $\boldsymbol{m} = \boldsymbol{m}_{\max}$ and we will see in the experiments of Section 4 that the proposed approach is able to explore small values of $\boldsymbol{m}$ and to select a value $\hat{\boldsymbol{m}}$ drastically smaller than $\boldsymbol{m}_{\max}$. This phenomenon, which reduces the computational burden of the quantization task, is also illustrated in the next section. The hyper-parameter $\boldsymbol{m}_{\max}$ is problem-dependent and should be adjusted by the practitioner to meet his/her interpretability requirements.

## 4. Numerical experiments

This section is divided into three 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. Second, the predictive quality of the new learned representation approach is illustrated on several classical benchmark datasets from the UCI library. Third, we use it on *Credit Scoring* datasets provided by Crédit Agricole Consumer Finance (CACF), a major European company in the consumer credit market. The Python notebooks of all experiments, excluding the confidential real data of CACF, are available online[1].

### 4.1. Simulated data: empirical consistency and robustness

We focus here on discretization of continuous features (similar experiments could be conducted on categorical ones). Two continuous features $x_1$ and $x_2$ are sampled from the uniform distribution on $[0, 1]$ and discretized using

$$\boldsymbol{q}_1(\cdot) = \boldsymbol{q}_2(\cdot) = (\mathbb{1}_{(-\infty,1/3]}(\cdot), \mathbb{1}_{(1/3,2/3]}(\cdot), \mathbb{1}_{(2/3,\infty]}(\cdot)).$$

Here, following (2), we have $d = 2$ and $m_1 = m_2 = 3$ and the cutpoints are $c_{j,1} = 1/3$ and $c_{j,2} = 2/3$ for $j = 1, 2$. Setting $\boldsymbol{\theta} = (0, -2, 2, 0, -2, 2, 0)$, the target feature $y$ is then sampled from $p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{q}(\boldsymbol{x}))$ via the logistic model (3).

From the *glmdisc* algorithm, we studied three cases:

(A) First, the quality of the cutoff estimator $\hat{c}_{j,2}$ of $c_{j,2} = 2/3$ is assessed when the starting maximum number of intervals per discretized continuous feature is set to its true value $m_1 = m_2 = 3$;

(B) Second, we estimated the number of intervals $\hat{m}_1$ of $m_1 = 3$ when the starting maximum number of intervals per discretized continuous feature is set to $m_{\text{max}} = 10$;

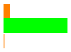
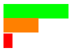
(C) Last, we added a third feature $x_3$ also drawn uniformly on $[0, 1]$ but uncorrelated to $y$ and estimated the number $\hat{m}_3$ of discretization intervals selected for $x_3$. The reason is that a non-predictive feature which is discretized or grouped into a single value is *de facto* excluded from the model, and this is a positive side effect.

From a statistical point of view, experiment (A) assesses the empirical consistency of the estimation of $C_{j,h}$ motivated in Section 3.2.2, whereas experiments (B) and (C) focus on the consistency of the estimation of $m_j$ motivated in Section 3.2.3. The results are summarized in Table 1 where either 95% confidence intervals ((Sun & Xu, 2014), hereafter CI) or bar plots are given, with a varying sample size. Two iterations of experiment (A) are displayed on Figure 2: at first (Figure 2a), the proposed neural network fails to recover the true underlying discretization but after 300 iterations (Figure 2b), the "smooth" discretization $\boldsymbol{q}_{\boldsymbol{\alpha}}$ and its *maximum a posteriori* $\hat{\boldsymbol{q}}$ get closer to the data generating mechanism, resulting in a very good estimation of $c_{j,2}$ (Column (A) of Table 1). Also note that the slight underestimation ($\hat{m}_1 = 2$ for 9 experiments out of 100) in (B) for $n = 1{,}000$ is a classical consequence of the BIC criterion on small samples. As for (C) and as expected, spurious

[1] https://adimajo.github.io

correlations with a small sample allow $x_3$ to enter the model with either $\hat{m}_3 = 2$ intervals (32 experiments out of 100) or $\hat{m}_3 = 3$ intervals (8 experiments out of 100). However, with a larger sample, feature $x_3$ is rightfully omitted from the final model, *i.e.* with $\hat{m}_3 = 1$ interval (88 experiments out of 100).

*Table 1.* For different sample sizes $n$, (A) CI of $\hat{c}_{j,2}$ for $c_{j,2} = 2/3$. (B) Bar plot of $\hat{m} = 2, 3, 4$ (resp.) for $m_1 = 3$. (C) Bar plot of $\hat{m}_3 = 1, 2, 3$ (resp.) for $m_3 = 1$.

| $n$ | (A) $\hat{c}_{j,2}$ | (B) | $\hat{m}_1$ | (C) | $\hat{m}_3$ |
|---|---|---|---|---|---|
| 1,000 | $[0.656, 0.666]$ | 9<br>90<br>1 |  | 60<br>32<br>8 |  |
| 10,000 | $[0.666, 0.666]$ | 0<br>100<br>0 | | 88<br>12<br>0 | |

## 4.2. Benchmark data

To test further the effectiveness of *glmdisc* in a predictive setting, we gathered 6 datasets from the UCI library: the Adult dataset ($n = 48{,}842$, $d = 14$), the Australian dataset ($n = 690$, $d = 14$), the Bands dataset ($n = 512$, $d = 39$), the Credit-screening dataset ($n = 690$, $d = 15$), the German dataset ($n = 1{,}000$, $d = 20$) and the Heart dataset ($n = 270$, $d = 13$). Each of these datasets has mixed (continuous and categorical) features and a binary response to predict. To get more information about these datasets, their respective features, and the predictive task associated with them, the interested reader may refer to the UCI website[2].

Now that we made sure that our approach is empirically consistent, *i.e.* it is able to find the true quantization in a well-specified setting, we wish to verify now that embedding the learning of a good quantization in the predictive task *via glmdisc* is better than other methods that rely on *ad hoc* criteria. As we were primarily interested in logistic regression, we will compare our approach to a "naïve" additive linear logistic regression (on non-quantized features - hereafter ALLR), a logistic regression on continuous discretized data using the now standard MDLP algorithm from (Fayyad & Irani, 1993) and categorical grouped data using $\chi^2$ tests of independence between each pair of factor levels and the target in the same fashion as the ChiMerge discretization algorithm proposed by Kerber (1992) (hereafter MDLP/$\chi^2$). As the original use case stems from *Credit Scoring*, we use the performance metric usually monitored by *Credit Scoring* practitioners, which is the Gini coefficient, directly related to the Area Under the ROC Curve (Gini $= 2 \times \text{AUC} - 1$). In this Section and the next, Gini indices are reported on a random 30 % test set. Table 2 shows our approach yields significantly better results on these rather small datasets where the added flexibility of quantization might help the predictive task.

*Table 2.* Gini indices (the greater the value, the better the performance) of our proposed quantization algorithm *glmdisc* and two baselines: ALLR and MDLP / $\chi^2$ tests obtained on several benchmark datasets from the UCI library.

| Dataset | ALLR | MDLP/$\chi^2$ | *glmdisc* |
|---|---|---|---|
| Adult | 81.4 (1.0) | **85.3 (0.9)** | 80.4 (1.0) |
| Australian | 72.1 (10.4) | 84.1 (7.5) | **92.5 (4.5)** |
| Bands | 48.3 (17.8) | 47.3 (17.6) | **58.5 (12.0)** |
| Credit | 81.3 (9.6) | 88.7 (6.4) | **92.0 (4.7)** |
| German | 52.0 (11.3) | 54.6 (11.2) | **69.2 (9.1)** |
| Heart | 80.3 (12.1) | 78.7 (13.1) | **86.3 (10.6)** |

## 4.3. *Credit Scoring* data

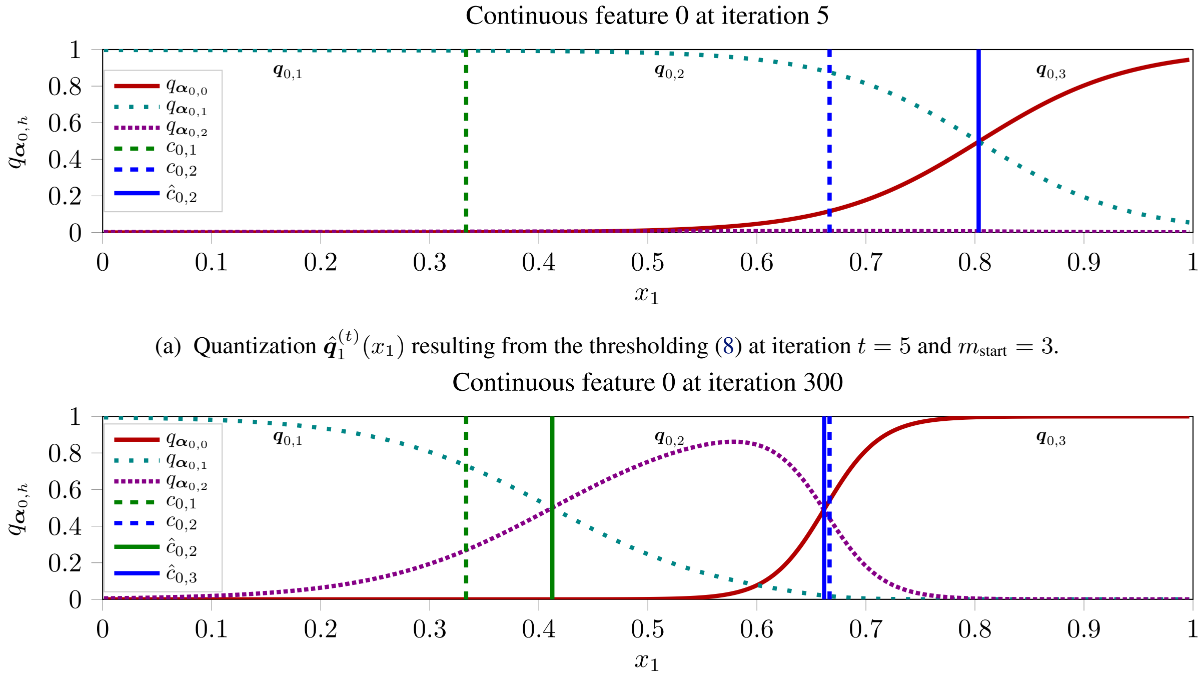
Discretization and grouping are preprocessing steps relatively "manually" performed in the field of *Credit Scoring*, using $\chi^2$ tests for each feature or so-called Weights of Evidence (Zeng, 2014). This back and forth process takes a lot of time and effort and provides no particular statistical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

Table 3 shows Gini coefficients of several portfolios for which there are $n = 50{,}000$, $n = 30{,}000$, $n = 50{,}000$, $n = 100{,}000$, $n = 235{,}000$ and $n = 7{,}500$ clients respectively and $d = 25$, $d = 16$, $d = 15$, $d = 14$, $d = 14$ and $d = 16$ 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, the naïve additive linear logistic regression (ALLR) and *ad hoc* methods (MDLP/$\chi^2$) introduced in the previous section to our *glmdisc* proposal. 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.* Our approach shows approximately similar results than MDLP/$\chi^2$, potentially due to the fact that contrary to the two previous experiments with simulated or UCI data, the classes are imbalanced ($< 3\%$ defaulting loans), which would require special treatment while back-propagating the gradients (Anand et al., 1993). Note however that it is never significantly worse; for the Electronics dataset and as was the case for most UCI datasets, *glmdisc* is significantly superior, which in the *Credit Scoring* business might end up saving millions to the financial institution.

Regarding complexity, there are at most $\mathcal{O}(m_j^2)$ $\chi^2$ tests performed in all benchmarks for categorical features as initially, all pairwise tests have to be computed. The MDLP

[2] (Dheeru & Karra Taniskidou, 2017) : http://archive.ics.uci.edu/ml



(a) Quantization $\hat{\boldsymbol{q}}_1^{(t)}(x_1)$ resulting from the thresholding (8) at iteration $t = 5$ and $m_{\text{start}} = 3$.

(b) Quantizations $\hat{\boldsymbol{q}}_1^{(t)}(x_1)$ resulting from the thresholding (8) at iteration $t = 300$ and $m_{\text{start}} = 3$.

*Figure 2.* Quantizations $\hat{\boldsymbol{q}}_1^{(t)}(x_1)$ of experiment (A) resulting from the thresholding (8).

*Table 3.* Gini indices (the greater the value, the better the performance) of our proposed quantization algorithm *glmdisc*, the two baselines of Table 2 and the current scorecard (manual / expert representation) obtained on several portfolios of Crédit Agricole Consumer Finance.

| Portfolio | ALLR | Current | MDLP/$\chi^2$ | *glmdisc* |
|---|---|---|---|---|
| Automobile | **59.3 (3.1)** | 55.6 (3.4) | **59.3 (3.0)** | 59.1 (3.0) |
| Renovation | 52.3 (5.5) | 50.9 (5.6) | 54.0 (5.1) | **56.7 (4.8)** |
| Standard | 39.7 (3.3) | 37.1 (3.8) | **45.3 (3.1)** | 44.0 (3.1) |
| Revolving | 62.7 (2.8) | 58.5 (3.2) | **63.2 (2.8)** | 62.3 (2.8) |
| Mass retail | 52.8 (5.3) | 48.7 (6.0) | 61.4 (4.7) | **61.8 (4.6)** |
| Electronics | 52.9 (11.9) | 55.8 (10.8) | 56.3 (10.2) | **72.6 (7.4)** |

algorithm has to first sort the training samples ($\mathcal{O}(n \ln n)$ operations) and then recursively assess the entropy produced by cutting at each "boundary point", *i.e.* where consecutive training points, say $x_{i,j}, x_{i',j}$, have different targets ($y_i \neq y_{i'}$). There are $\mathcal{O}(b_j^2)$ such operations where $b_j$ is the number of these "boundary points" (Ramírez-Gallego et al., 2016). Our approach, the *glmdisc* algorithm, requires that we fit a softmax with $m_j$ output classes per feature and training epoch $(t)$ which is quite low. About the length of the gradient ascent chain, there is no stopping rule except the time budget T. However, the required T value to obtain relevant candidates is low: approx. 20-40 iterations for the experiments of Section 4. Figure 2 uses a small learning rate to showcase both the empirical consistency of the relax and the effect of the MAP scheme in exploring a lower number of quantization levels $m_j$. 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 each dataset of Table 3, making it in this regard also comparable to MDLP/$\chi^2$ methods.

## 5. Concluding remarks

Feature quantization (discretization for continuous features, grouping of factor levels for categorical ones) in a supervised multivariate classification setting is a recurring problem in many industrial contexts. It was formalized as a highly combinatorial representation learning problem and a new algorithmic approach, named *glmdisc*, has been proposed as a sensible approximation of a classical statistical information criterion.

This algorithm relies on the use of a softmax approximation of each discretized or grouped feature. This proposal can alternatively be replaced by any other univariate multiclass predictive model, which makes it flexible and adaptable to other problems. Prediction of the target feature, given quantized features, was exemplified with logistic regression, although here as well, it can be swapped with any other supervised classification model, provided it is the same as the output layer of the proposed neural network. Thus, the extension to penalized logistic regression or any

Generalized Linear Model is straightforward. Its good computational properties were put to use 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 automated 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 a rule of thumb, a month is generally allocated to data pre-processing for a single data scientist working on a single scorecard that can now be invested in tasks that add more value, *e.g.* more data, better data quality.

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 (pairwise interactions) or more 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.

## References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

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

Anand, R., Mehrotra, K. G., Mohan, C. K., and Ranka, S. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969, 1993.

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

Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Chamroukhi, F., Samé, A., Govaert, G., and Aknin, P. A regression model with a hidden logistic process for feature extraction from time series. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pp. 489–496. IEEE, 2009.

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

Dheeru, D. and Karra Taniskidou, E. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Fayyad, U. and Irani, K. Multi-interval discretization of continuous-valued attributes for classification learning. In *13th International Joint Conference on Artificial Intelligence*, pp. 1022–1029, 1993.

Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

Kass, G. V. An exploratory technique for investigating large quantities of categorical data. *Applied statistics*, pp. 119–127, 1980.

Kerber, R. Chimerge: Discretization of numeric attributes. In *Proceedings of the tenth national conference on Artificial intelligence*, pp. 123–128. Aaai Press, 1992.

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

Ramírez-Gallego, S., García, S., Mouriño-Talín, H., Martínez-Rego, D., Bolón-Canedo, V., Alonso-Betanzos, A., Benítez, J. M., and Herrera, F. Data discretization: taxonomy and big data challenge. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(1): 5–21, 2016.

Reverdy, P. and Leonard, N. E. Parameter estimation in softmax decision-making models with linear objective functions. *IEEE Transactions on Automation Science and Engineering*, 13(1):54–67, 2016.

Samé, A., Chamroukhi, F., Govaert, G., and Aknin, P. Model-based clustering and segmentation of time series with changes in regime. *Advances in Data Analysis and Classification*, 5(4):301–321, 2011.

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

Sun, X. and Xu, W. Fast implementation of delong's algorithm for comparing the areas under correlated receiver oerating characteristic curves. *IEEE Signal Processing Letters*, 21(11):1389–1393, 2014.

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

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

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