# Checkered Regression

Mastane Achab[*]

This paper introduces the checkered regression model, a nonlinear generalization of logistic regression. More precisely, this new binary classifier relies on the multivariate function $\frac{1}{2}\left(1 + \tanh(\frac{z_1}{2}) \times \cdots \times \tanh(\frac{z_m}{2})\right)$, which coincides with the usual sigmoid function in the univariate case $m = 1$. While the decision boundary of logistic regression consists of a single hyperplane, our method is shown to tessellate the feature space by any given number $m \geq 1$ of hyperplanes. In order to fit the model's parameters to some labeled data, we describe a classic empirical risk minimization framework based on the cross entropy loss. A multiclass version of our approach is also proposed.

## 1 Introduction

Logistic regression (LR) is one of the most standard approaches for binary classification: it simply learns a linear prediciton rule, through a convex minimization problem in the case of the cross entropy loss (see e.g. [2], [5]). Nevertheless, it suffers from a lack of representational power: indeed, it cannot handle nonlinear relations between features and labels. For that reason, artificial neural networks (a.k.a. deep learning models) are nowadays preferred over linear methods such as LR across a broad spectrum of machine learning applications, ranging from image or speech recognition to natural language processing ([6], [4]). But in general, the optimization of a deep neural network is a highly non-convex problem for which we still have little theoretical understanding. As an alternative to deep learning, this paper proposes a new binary classifier that strictly generalizes LR: we call it the *checkered regression* (CR) model. Loosely speaking, CR can be seen as a single hidden-layer neural network with tanh activations that are multiplied together, instead of being additively combined as is customary. This is somehow similar to the NALU module proposed in [8] that computes the elementwise product of tanh and sigmoid activations. Contrary to LR, the scope of CR expands beyond linear separability. As shall be seen in the next section, the predictions of a CR model can tile the feature space into a checkerboard-like pattern. The paper is organized as follows. After defining our model in Section 2, we discuss its optimization in Section 3. Finally, we illustrate our approach with a basic numerical experiment in Section 4.

[*] mastane.achab@gmail.com

## 2 The checkered regression model

This section formally introduces the checkered regression model along with a few elementary properties. Let $m \geq 1$ and $\mathbf{1} = (1, \ldots, 1)$ be the all-ones vector of size $m$. We start with the definitions below.

**Definition 1.** (Checkoid function). *The checkoid function $\Xi_m$ is defined for all $z = (z_1, \ldots, z_m) \in \mathbb{R}^m$ by*

$$\Xi_m(z) = \frac{1}{2}\left(1 + \prod_{k=1}^{m} \tanh\left(\frac{z_k}{2}\right)\right) = \frac{\sum_{v \in \{0,1\}^m \ s.t.\ \mathbf{1}^\intercal v \equiv 0[2]} e^{-v^\intercal z}}{(1+e^{-z_1}) \times \cdots \times (1+e^{-z_m})} \ .$$

**Definition 2.** (Checkered regression). *Let $\mathcal{X} \subseteq \mathbb{R}^d$ $(d \geq 1)$. The checkered regression model with parameters $\omega = (\omega_1, \ldots, \omega_m) \in \mathbb{R}^{dm}$ is given by the posterior probabilities*

$$p_\omega(0|x) = 1 - p_\omega(1|x) = \Xi_m(\omega_1^\intercal x, \ldots, \omega_m^\intercal x) \quad \textit{for all } x \in \mathcal{X} \ .$$

For $m = 1$, $\Xi_1 = \sigma$ is the sigmoid function and the checkered regression is simply a logistic regression. If $m = 2$, the checkoid function is also equal to

$$\Xi_2(z_1, z_2) = \frac{\tanh\left(\frac{z_1}{2}\right) + \tanh\left(\frac{z_2}{2}\right)}{2\tanh\left(\frac{z_1+z_2}{2}\right)} \ .$$

For general $m \geq 1$, we give next two properties of the checkoid function and CR.

**Proposition 1.** (Symmetry of $\Xi_m$). *Let $z = (z_1, \ldots, z_m) \in \mathbb{R}^m$, $k \in \{1, \ldots, m\}$ and $z' = (z'_1, \ldots, z'_m)$ with $z'_k = -z_k$ and $z'_j = z_j$ for $j \neq k$. Then,*

$$\Xi_m(z') = 1 - \Xi_m(z).$$

*Proof.* By the oddness of the hyperbolic tangent. □

**Lemma 1.** (Hamming distance parity). *Consider a checkered regression model with parameters $\omega = (\omega_1, \ldots, \omega_m)$. Let $x$ and $x'$ be two points in $\mathbb{R}^d$ both outside the $m$ hyperplanes of the CR model, i.e. such that $\omega_k^\intercal x \neq 0$ and $\omega_k^\intercal x' \neq 0$ for all $1 \leq k \leq m$. Then, the two following conditions are equivalent.*

*(i) The CR model predicts that $x$ and $x'$ share the same label:*

$$sign\left(p_\omega(0|x) - \frac{1}{2}\right) = sign\left(p_\omega(0|x') - \frac{1}{2}\right) \ .$$

*(ii) The Hamming distance*

$$\sum_{k=1}^{m} \mathbb{I}\left\{sign\left(\omega_k^\intercal x\right) \neq sign\left(\omega_k^\intercal x'\right)\right\}$$

*is even.*

*Proof.* By successive applications of Proposition 1. □

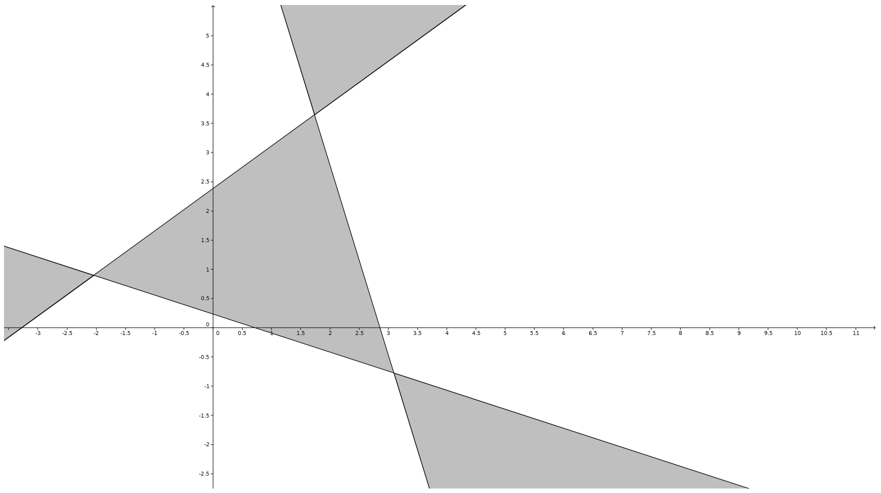

Figure 1: A checkered hyperplane tessellation of the Euclidean plane.

In the univariate case, Proposition 1 corresponds to the common identity $\sigma(-z) = 1 - \sigma(z)$ satisfied by the sigmoid. More interestingly, Lemma 1 shows that the decision regions of a CR model form a hyperplane tessellation of the feature space $\mathcal{X}$, where the tiles are binary labeled in a checkered fashion.

**Multiclass CR** We now define a muticlass version of our model that can be used in classification problems having more than two classes. Let $\mathcal{Y} = \{0, \dots, c-1\}$ be the set of classes with $c \geq 3$.

**Definition 3.** (Multiclass checkered regression). *The multiclass checkered regression model with parameters $\Omega = (\Omega_1, \dots, \Omega_m) \in \mathbb{R}^{dm(c-1)}$ is given by the posterior probabilities*

$$\forall y \in \mathcal{Y}, \quad p_\Omega(y|x) = \Xi_{m,y}(\Omega_1 x, \dots, \Omega_m x) ,$$

*where the multiclass checkoid function $\Xi_{m,y}$ is defined for all $Z \in \mathbb{R}^{(c-1)\times m}$ as*

$$\Xi_{m,y}(Z) = \frac{\sum_{v \in \mathcal{Y}^m \ s.t. \ \mathbf{1}^\intercal v \equiv y[c]} e^{-tr(E_v Z)}}{\left(1 + \sum_{j=1}^{c-1} e^{-Z_{j,1}}\right) \times \cdots \times \left(1 + \sum_{j=1}^{c-1} e^{-Z_{j,m}}\right)} ,$$

*where $E_v$ is the $m \times (c-1)$ matrix such that $[E_v]_{k,j} = \mathbb{I}\{j = v_k\}$ for $1 \leq k \leq m, 1 \leq j \leq c-1$.*

For $m = 1$, multiclass CR coincides with the well-known multiclass (or 'softmax') LR model. As a remark, the multiclass checkoid functions may as well be expressed through generalized tanh functions with the $c$-th roots of unity. For simplicity, we restrict our attention to binary classification in the remaining of the paper.

# 3 Optimization

Given a random pair $(X, Y)$ valued in $\mathcal{X} \times \{0, 1\}$ and a collection $((X_1, Y_1), \dots, (X_n, Y_n))$ of i.i.d. copies of $(X, Y)$, we follow the empirical risk minimization paradigm ([3]). In other words, we adjust the parameters $\omega$ of our CR model to match the true posterior probabilities

$$\mathbb{P}(Y = 0|X = x) = 1 - \mathbb{P}(Y = 1|X = x),$$

by minimizing some empirical risk. By analogy with LR, we take the cross entropy loss:

$$\min_{\omega} \widehat{L}(\omega) := \frac{1}{n} \sum_{i=1}^{n} -Y_i \log p_{\omega}(1|X_i) - (1 - Y_i) \log p_{\omega}(0|X_i) \ , \tag{1}$$

which also corresponds to maximum likelihood estimation.

**Gradient** A tempting way of solving the optimization problem (1) is via stochastic gradient descent (SGD). At each iteration of SGD, we need to compute the gradient (with respect to $\omega$) of the loss induced by a single training pair $(X_i, Y_i)$. The next result allows to do this differentiation.

**Proposition 2.** (Partial derivatives). *Consider the logarithmic loss: for all* $z = (z_1, \dots, z_m) \in \mathbb{R}^m$,

$$\ell(z) = -\log(\Xi_m(z)) \ .$$

*Then for any* $1 \le k \le m$, *the* $k$*-th partial derivative of* $\ell$ *is*

$$\frac{\partial \ell}{\partial z_k}(z) = \sigma(z_k) \cdot \left(1 - \frac{\Xi_{m-1}(z_{-k})}{\Xi_m(z)}\right) \ ,$$

*where* $z_{-k} = (z_j)_{j \neq k}$ .

By combining Propositions 1 and 2, we obtain the partial derivatives of the log-loss $\tilde{\ell}(z) = -\log(1 - \Xi_m(z))$ on the other class:

$$\frac{\partial \tilde{\ell}}{\partial z_k}(z) = -(1 - \sigma(z_k)) \cdot \left(1 - \frac{\Xi_{m-1}(z_{-k})}{1 - \Xi_m(z)}\right) \ .$$

In particular, the partial derivatives of $\ell$ and $\tilde{\ell}$ are all bounded in the interval $(-1, 1)$. The next paragraph links our approach to the concept of submodularity.

**Submodularity** We recall from [1] that a continuous real-valued function $H$ defined on the product $\mathscr{X} = \prod_{k=1}^{m} \mathscr{X}_k$ of $m$ compact subsets $\mathscr{X}_k$ of $\mathbb{R}$ is *submodular* if and only if for all $(z, z') \in \mathscr{X}^2$ such that $\{z, z'\} \neq \{\min(z, z'), \max(z, z')\}$,

$$H(z) + H(z') \ge H(\min(z, z')) + H(\max(z, z')) \ , \tag{2}$$

with component-wise min and max operations. If the inequality is strict in Eq. (2), then $H$ is *strictly submodular.* In the case $m = 2$, the following lemma states that the logarithmic loss of the checkoid belongs to this class of functions.

**Lemma 2.** (Strict submodularity). *The log-loss* $\ell(z_1, z_2) = -\log(\Xi_2(z_1, z_2))$ *is strictly submodular.*

*Proof.* For any $1 \leq j \neq k \leq m$, the cross-second-order derivative

$$\frac{\partial^2 \ell}{\partial z_j \partial z_k}(z) = -\frac{\sigma(z_j)(1-\sigma(z_j)) \cdot \sigma(z_k)(1-\sigma(z_k))}{\Xi_m(z)^2} \cdot \prod_{l \neq j,k} \tanh\left(\frac{z_l}{2}\right)$$

is strictly negative if $m = 2$, which is a sufficient condition for strict submodularity ([1]). □

Symmetrically, it can be shown that $-\tilde{\ell}$ is strictly submodular, where $\tilde{\ell}(z_1, z_2) = -\log(1 - \Xi_2(z_1, z_2))$ is the log-loss on the other class. This is an interesting property: indeed, submodular functions can be minimized and approximately maximized efficiently. Though this result is limited to the bivariate setting, it may suggest that there exists a more general structure to be discovered for any arbitrary number of variables.

## 4 Numerical illustration

We fit a CR model with $m = 3$ hyperplanes to the (checkered) dataset depicted in Figure 2. The feature vectors $x = (x_1, x_2, 1)$ are augmented with a third coordinate in order to learn bias parameters. After 10 epochs, SGD (with learning rate set to 0.1) attains 99% classification accuracy on the training data. The same performance is also obtained with a model having more hyperplanes than needed (for instance $m = 20$).

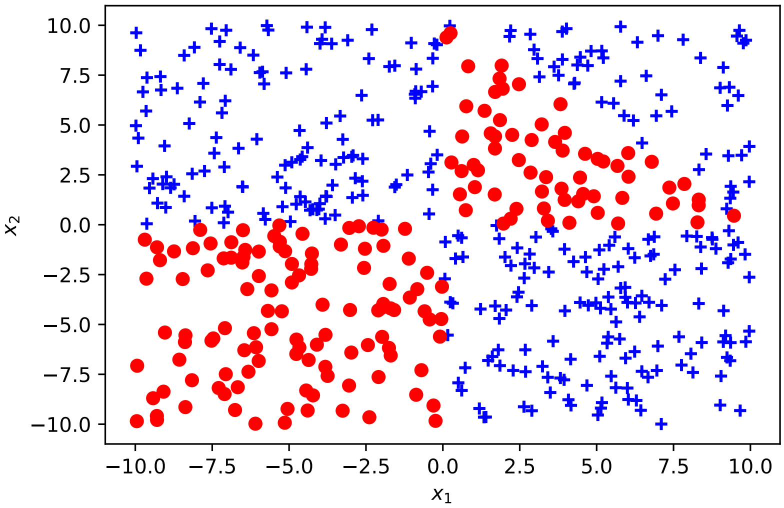

Figure 2: Training data ($n = 500$ points).

### Acknowledgment

The author thanks Jaouad Mourtada for his precious pointer to the notion of submodularity.

# References

[1] F. Bach. Submodular functions: from discrete to continuous domains. *Mathematical Programming*, 175(1):419–459, 2019.

[2] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[3] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

[4] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[5] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[6] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[7] Y. Plan and R. Vershynin. Dimension reduction by random hyperplane tessellations. *Discrete & Computational Geometry*, 51(2):438–461, 2014.

[8] A. Trask, F. Hill, S. E. Reed, J. Rae, C. Dyer, and P. Blunsom. Neural arithmetic logic units. *Advances in neural information processing systems*, 31, 2018.