Conic Particules Gradient Descent for Kernel logistic Regression

Antoine SIMOES[1] Yohann DE CASTRO[1,2]

[1] Institut Camille Jordan (CNRS UMR 5208), Ecole Centrale Lyon, 36 avenue Guy de Collongue, 69134 Écully, France

[2]Institut Universitaire de France (IUF)

Résumé – Dans cet article, nous développons l'intérêt des modèles continus parcimonieux pour la regression logistique à noyau dans le cadre de classification binaire. Contrairement à la logistique ridge regression, notre méthode ne nécessite pas autant de paramètres que d'observations. Notre classifieur se base sur des avancées récentes dans l'étude de modèles pénalisés grâce à la norme de variation totale qui nous permet d'obtenir une estimée avec peu de paramètres, intrinsèque à la parcimonie de la fonction cible. Cette technique est particulièrement intéressante quand la taille de échantillon est très grande. Notre estimée est obtenue grâce à une descente de gradient sur l'espace des mesures discrètes, et plus particulièrement à une descente de gradient conique particulaire. Nous présentons ici une étude méthodologique des différences et avantages entre la régression ridge à noyau et notre technique, appelée ici Beurling Logistic ou BLogistic.

Abstract – In this article, we investigate the benefits of continuous sparse models for the kernel logistic regression in binary classification. Contrary to kernel ridge logistic regression, our method does not fit a parameter whose size grows linearly with the number of data. Our classifier is based on recent advances in total variation norm regularization and enables us to directly estimate fewer parameters, intrinsic to the sparsity of the target function. This technique is particularly interesting when the sample is large. Our estimate is conducted via a gradient descent on the space of discrete measure, referred to as the conic particle gradient descent in the literature. We present a comprehensive methodological study of the differences and advantages between kernel ridge regression and our technique, referred to as the Beurling Logistic or BLogistic.

1 Introduction

Classification is a standard task in supervised learning, where one aims at predicting labels y from features x. Logistic regression [10] is by and large the most frequently used model to estimate the probability of a binary response. Under this model, there exists a function $f^\star$ such that

$$\mathbb{P}(Y = y \mid X = x) = \sigma(yf^\star(x)) \tag{1}$$

where $x \in \mathcal{X}$ is a feature in real some feature space $\mathcal{X} \subset \mathbb{R}^d$, $y \in \{-1, +1\}$ the response variable and σ is the sigmoid function $\sigma(t) := \log(1 + \exp(-t))$. Considering now that we observed a set of features $(z_i)_{1\leq i\leq n} = (x_i, y_i)_{1\leq i\leq n} \in \mathcal{X} \times \{-1, +1\}$ for a linear model $f_\theta(x) = \langle \theta, x\rangle$ and minimizing the empirical logistic loss

$$L_n(f_\theta) := \frac{1}{n}\sum_{i=1}^{n} \sigma(y_i f_\theta(x_i)) \,, \tag{2}$$

one obtains the standard logistic estimator. In high-dimensions, when d is large, sparsity promoting regularization such as the ℓ_1-norm are often used to gain in generalization and select relevant predictors [7], which leads to the logistic lasso estimator

$$\min_{\theta} \Big\{ L_n(f_\theta) + \lambda L_1(\theta) \Big\} \,, \tag{3}$$

where $\lambda L_1(\theta)$ is the LASSO regularizing term.

An extension to non-parametric models is often done by considering the kernel logistic regression. In this setting, the learned function is $f_h(x) = \langle h, \varphi(x)\rangle$ where $h \in \mathcal{H}$ is an element of some Hilbert $\mathcal{H}$ (called Reproducing Kernel Hilbert Space or RKHS) and $\varphi : \mathcal{X} \to \mathcal{H}$ is the so called feature map. To obtain a tractable estimator, one considers the logistic kernel ridge regression which reads

$$\min_{h\in\mathcal{H}} \Big\{ L_n(f_h) + \lambda L_2(h) \Big\} \,, \tag{4}$$

where $\lambda L_2(h)$ is the kernel RIDGE regularizing term. In this case, the solution reads

$$h^{\text{ridge}} = \sum_{i=1}^{n} \theta_i \varphi(x_i) \,, \tag{5}$$

where $\theta \in \mathbb{R}^n$ [6]. When the sample size n is large, this estimator reaches its limits, while having as many parameters (θ_i) as the sample size. In that case, we should want an estimator composed with less parameters, by transposing the real case and the LASSO ideas, the solution that we consider here is to change the penalization term to

$$\min_{h\in\mathcal{H}} \Big\{ L_n(f_h) + \lambda L_1(h) \Big\} \,, \tag{6}$$

where $\lambda L_1(h)$ denotes an ℓ_1 penalization term.

Those kinds of model have already been studied [8] and are known as off-the-grid methods. An interesting feature is that the sample size n does not governs the size of the estimator. This technique allows to find an estimator with eventually far less parameters

$$\hat{f}_\lambda = \sum_{i=1}^{p} a_i \varphi(t_i) \,, \tag{7}$$

where the $(a_i, t_i)_{1\leq i\leq p}$ are the amplitude and position parameters of the estimator.

Recently, optimization methods of (9) have been proposed through the lens of Wasserstein gradient flows on the space of measures. In particular, L. Chizat [4] developed an algorithm called *conic particle gradient descent* which has global convergence of the projected gradient flow [4, Theorem 4.1] and local convergence of the projected gradient descent. In this article, we will focus on the so-caled *BLogistic* estimator (9) and its benefits when compared to RIDGE estimator (4).

As we have seen, on one hand, for real case studies and high dimensional problems, the LASSO (3) provides an alternative to ℓ_2 penalization in order to have a low dimensional estimator [7]. On the other hand, there exist works on kernel-based methods that ensure convergence and statistical error bounds [9] under Tikhonov regularization (4). Statistical error bounds have also been studied under L_1 regularization on the space of measures point of view [8] with a will of having an estimator composed of few parameters for the case of large samples; there are also studies about implementation [3] which prove the convergence of the gradient flow. But all of these works focused on the mean square error loss and we provide in this work a study of the logistic loss with L_1 regularization for kernel-based methods.

In this article, we develop the idea of sparsity in some kernel-based model via a representative measure set. Then we show in a toy model the interest of the kernel logistic regression via conic particle gradient descent algorithm that we improve. Finally, we obtain a new estimator that admits less parameters and provides a closer estimation of our Bayes function.

2 Conic Particle Gradient Descent

2.1 Sparse hypothesis and ℓ_1-regularization

A standard hypothesis of real-valued high dimensional problem is the sparsity of the parameter. This hypothesis assumes a low dimensional solution to the problem. The RKHS equivalent to the real sparsity is a little more complex and we the kernel mean embedding function (KME). Under some assumptions of integrability over φ, for every $\mu \in \mathcal{M}(\mathcal{X})$ we can define the KME as

$$\Phi(\mu) = \int_{x\in\mathcal{X}} \varphi(x) d\mu(x) \tag{8}$$

This mathematical object is well studied throught the knowledge of properties on the kernel [1]. That mapping leads to a sense of sparsity for $f \in \mathcal{H}$ if we can write $f = \Phi(\sum_{j=1}^{p} a_j \delta_{t_j})$ where δ_t denotes a Dirac in t. This writing embodies perfectly the sense of sparsity as we define earlier, such a function can be written $f = \sum_{j=1}^{p} a_j \varphi(t_j)$ for some signed amplitudes a. This leads to the folowing hypothesis of sparsity

Hypothesis 1 (H1). *We assume that $f^\star$ is $p^\star$-sparse:* $\exists (a_j^\star)_{1\le j\le p^\star} \in \mathbb{R}^{p^\star}$, $(t_j^\star)_j$, $\mu^\star := \sum_{j=1}^{p^\star} a_j^\star \delta_{t_j^\star}$: $f^\star = \Phi(\mu^\star)$.

Under Hypothesis 1, the Kernel Ridge Logistic (KRL) and its representer theorem seems to be exceeded. Their estimator having n parameters, the parameters of the weight a_i over each observation and it never tries to match any kind of sparsity such as the number of Dirac in the Bayes function. The objective here is to find a new way to catch the right number of Dirac $p^\star$, their position and amplitude in order to obtain really few parameters. In the real case, some results achieve that switching from a ℓ_2 penalization to a ℓ_1 penalization implies some sparsity on the estimator [7]. Finding an ℓ_1-norm in this Hilbert space is possible thanks to the KME and the Total Variation (TV) norm in the measure set, from those two remarks we get the kernel logistic regression with ℓ_1 penalization model as

$$\underset{f\in\mathcal{H}}{\text{minimize}}\, J(f) := \mathcal{R}(f) + \lambda \Omega_{TV}(f) \tag{9}$$

where $\mathcal{R}(f) = \frac{1}{n}\sum_{i=1}^{n} \log(1+\exp(-y\langle f, \varphi(x)\rangle_{\mathcal{H}})$ is a the logistic risk term, $\Omega_{TV}(f) := \inf\{\|\mu\|_{\text{TV}} \mid \mu \in \mathcal{M}(\mathcal{X}) : f = \Phi(\mu)\}$ an ℓ_1 penalization term as in Beurling work [2] and λ the penalization parameter. This function Ω_{TV} is also called in the literature as the variation norm and there exists statistical guarantees under Hypothesis 1 [8] for the mean square error loss. Indeed, under additional assumptions of sources separation (which just mean that the $t_j^\star$ are not too close) and that there exists some observation sufficiently close, we can obtain a control on the error of estimation of $a_i^\star$ and $t_i^\star$ of the order of $\sqrt{p^\star}\lambda$, for λ of the order of $1/\sqrt{n}$. It means that we are able to find the position $t_i^\star$ and that we put mass around this position that is close to $a_i^\star$. These guarantees ensure that one can find an estimator with as few as of $f^\star$.

2.2 Conic Particle Gradient Descent

We study Conic Particle Gradient Descent (CPGD) [3] in the case of the logistic loss. Its interest relies on the idea that we are looking for estimator in the close form $\hat{f}_\lambda = \sum_{i=1}^{p} a_i \varphi(t_i)$ for some $p, (a_i), (t_i)$ as the sought after solution shares this description by atomic measures. This close form seems to be reasonable since we are looking for such a Bayes function and that represents function that admits a sparse measure representation. For a certain p fixed, the problem can now be seen as a $\mathbb{R}^{2p}$ problem, the parameter being $(a_i)_{1\le i\le p}$ and $(t_i)_{1\le i\le p}$ and then we can rewrite the J function as a $\mathbb{R}^{2p}$ function

$$F_p\big((a_i,t_i)_{1\le i\le p}\big) := J\Big(\Phi(\sum_{i=1}^{p} a_i \delta_{t_i})\Big), \tag{10}$$

and, for fixed p, the minimization of this function can be solved by a Gradient Descent algorithm while we can compute its gradient.

Proposition 2.1 ($\nabla_{a_l} F_p, \nabla_{t_l} F_p$)
With the same notation as above, we get

$$\begin{aligned}
\nabla_{a_l} F_p((a_j,t_j)_j) &= \\
&\frac{1}{n}\sum_{i=1}^{n} -y_i k(x_i,t_l)\sigma\Big(-y_i\sum_{j=1}^{p} a_j k(x_i,t_j)\Big) + \lambda\epsilon_l \\
\nabla_{t_l} F_p((a_j,t_j)_j) &= \\
&\frac{1}{n}\sum_{i=1}^{n} -y_i a_l \nabla_t k(x_i,t_l)\sigma\Big(-y_i\sum_{j=1}^{p} a_j k(x_i,t_j)\Big),
\end{aligned} \tag{11}$$

where ϵ_l corresponds to the sign of the l-th particle and $k(x_1,x_2) = \langle\varphi(x_1),\varphi(x_2)\rangle$ denotes the kernel of the RKHS

$\mathcal{H}$. Since we consider a signed particle, we have a_j that are signed and we may want the particle to keep a sign fixed during the algorithm in order to avoid oscillation around 0 and to simplify the computation of the gradient of the amplitude. In order to conserve a fixed sign for all particles, we will consider some mirror descent on the amplitude. This leads to the following retraction on our parameters

$$\begin{aligned} a_l &\leftarrow a_l \exp(-\nabla_{a_l} F_p(a_i, t_i)) \\ t_l &\leftarrow t_l - \nabla_{t_l} F_p(a_i, t_i)/a_l \end{aligned} \tag{12}$$

the exponential term in the retraction of the amplitude comes from the mirror descent as in [5] and the will of a fixed sign for a precise particle along the algorithm. For the position retraction, we divide the gradient by the amplitude because of the conical geometry studied in [5], [3].

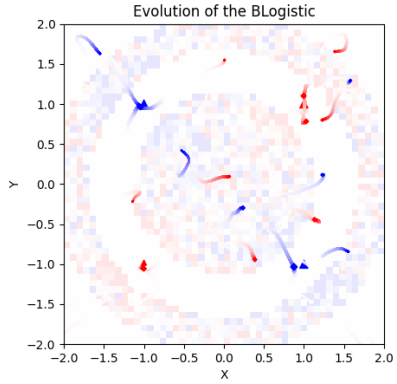

Figure 1: Figure of evolution of the CPGD ($\lambda = 10^{-4}$)

This figure represents the evolution and movement of particles along CPGD where colours become more and more intense after each step. We initialize CPGD with randomly chosen particles, work only with a gaussian kernel and let the CPGD running 10^3 iterations.

Definition 3 (Residual)**.** We name residual a rescaled version of the gradient of the loss risk term, that gives us

$$\eta(\mu) = \frac{\Phi^\star \nabla \mathcal{R} \Phi(\mu)}{\lambda}, \tag{13}$$

where $\Phi^\star$ is the dual operator of Φ, which can be computed in closed form .

For this theory, a solver is a critical point, and so, the residual over this point has to respect the nullity of the sub-differential. The derivative of the risk term is the residual; the derivative of the TV-norm penalization term is well known as $\{\pm 1\}$ depending on the sign of a particle and $[-1, 1]$ where there is no particles for a particular measure.

Proposition 3.1 (KKT condition)
Let μ be sparse measure $\mu = \sum_{i=1}^{p} a_i \delta_{t_i}$, we denote $T = \{t_1, .., t_p\}$ and $T_+ = \{t_i \mid a_i > 0, i \in \{1, .., p\}\}$ (resp. $T_- = \{t_i \mid a_i < 0, i \in \{1, .., p\}\}$) the set of position of the positive particles (resp. the set of position of the negative ones). If the measure μ satisfies the optimization problem 9, then it satisfies the following conditions:

$$\begin{cases} \eta(\mu)(t) = 1, \forall t \in T_+ \\ \eta(\mu)(t) = -1, \forall t \in T_- \\ \eta(\mu)(t) \in]-1, 1[, \forall t \notin T \end{cases} \tag{14}$$

The Gradient Descent is such as a particle that fall into the attraction basin of a Dirac will move in this direction and try to match a trade-off between the real particle and the mass we can accept to put on the particle that is limited by the ℓ_1-norm of penalization.

This leads us to the following algorithm

Algorithme 1 : Conic Particle Gradient Descent

input : Parameterize the regularization parameter λ

Initialization of the particles: $(a_i), (t_i)$;
while not stopping_criterion **do**
  compute $\nabla_{a_l} F_p, \nabla_{t_l} F_p$
  Actualisation of the particles according to the parameter retraction
  Refresh the stopping_criterion
end

with the retraction as we define in 12.

4 Simulated data

We did the simulation in a toy model in order to highlight the remarks we made earlier.

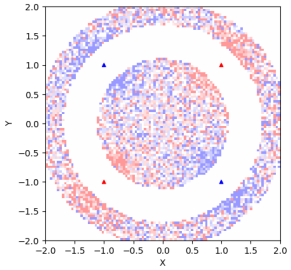

Figure 2: Observation and Bayes function

On Figure (2), we print out the observations that we simulate and the Bayes function we try to learn (represented by the four target points out of the data). For the rest of the article, blue colour corresponds to negative particles and red to positive ones .

First of all, $f^\star$ is written as the sum of four Dirac, the 4 triangle that we see in the coordinates $(\pm 1, \pm 1)$ and we take $n = 3000$ observations. In that case, we obtain the limit of the capacity of the classical kernel logistic ridge regression while it manages matrices of unreasonable size during computation. Secondly, the observation set $\mathcal{X}$ is composed of a reunion of two discs and those two discs are close enough of the Diracs to feel the impact of them on the observation but such that they don't include those Diracs.

We can represent the Bayes function in the toy model and the function we learn thanks to CPGD in figure (3).

We clearly see the four Dirac that corresponds to the $t_i^\star$ through the gaussian kernel on the lefter plot. We recognize those area in the BLogistic function while we can't recognize anything pattern in the KRL model. This incapacity of KRL to match the function reflects the dependency of the estimator that we get through the representer theorem.

In all simulation, we use a fix number of iteration as the stopping criterion but we may think for other one. The interest

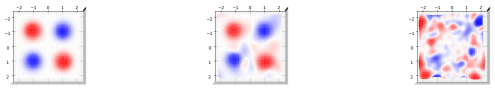

Figure 3: Probability to obtain 1 or -1 for the Bayes function $f^\star$ on the left, the estimator obtaines after the CPGD algorithm on the middle and Kernel Ridge Logistic on the right

of this algorithm remains to be with a lot less parameters and to match directly to the Bayes function.

4.1 Death and birth of particles

According to the algorithm, some particles tend to die along CPGD. Indeed, the gradient pushes the amplitude to zero. Since, at each step we need to compute some Gram matrix of the kernel for all the particles together, we added a condition in our algorithm to detect if some particle is going to die or not. This condition corresponds to the study of the second term in a Taylor approximation of the method. We can see this criterion on the residual with the KKT condition. For some particle, if the algorithm tends to decrease a small amplitude where the residual has already a value between -1 and 1, that correspond to a particle which doesn't succeed in the task of finding a Dirac.

The criterion to detect a dying particle relies on a Taylor formula, we only can remove particles one by one but since the condition is quite cheap to verify we can look at each step if a particle verifies it or not.

At the opposite, there exists a risk that any particle aggregate around a Dirac and that we didn't detect it. In order to verify if we explore the whole possible set, which could be possible by initialize a lot of particle in a grid sufficiently small but that would be horribly expensive in computation and memory. We implement a condition to see if the actual exploration is sufficient.

The idea is the following, we draw at random some new possible position and for each of them we look if the residual respects the KKT condition (14). If yes, the particle is not necessary but if the residual has a absolute value larger than 1, we create there a particle of the same sign as the residual. This condition gives a way to create new particles such that we ensure a loss diminution and so without any risk of non-converging algorithm. At this step we can draw as many particle as we want but that will increase the cost of this operation since we have to verify each particle independently but an exploration is mandatory to ensure the detection of all Dirac. It is where the algorithm suffer from the curse of dimension, since with a larger dimension gets the probability for a random particle to fall near enough to a Dirac become less probable. Since this computation is more expensive, we processed it only at fixed epochs.

Algorithme 2 : Conic Particle Gradient Descent with dying and birth of particle

input : Parametrize the regularization parameter λ

Initialization of the particles: $(a_i), (t_i)$;
while not stopping_criterion **do**
  compute $\nabla_{a_l} F_p, \nabla_{t_l} F_p$
  Actualisation of the particles according to the parameter retraction
  Remove a dying particle if needed
  At fixed epochs see if there is new born particles
  Refresh the stopping_criterion
end

5 Conclusion

This article deals with the over dependency of KRidge on the observation. Thanks to a new penalization and ideas developed in BLASSO, we study a new estimator that decrease significantly the number of parameters, and, in some critic cases may improve the distance to the target function. We used CPGD to solve our model and ensure its sparsity. In future works, we would like to develop either the theoretical aspect of the model with developing a certificate for its convergence either in the algorithmic aspect with the creation smarter stopping criterion and imagine a way to merge two close enough particles.

References

[1] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer, New York, 2004.

[2] Arne Beurling. *Sur les intégrales de Fourier absolument convergents et leur application à une transformation functionelle*. 1939.

[3] Lenaic Chizat. Sparse optimization on measures with over-parameterized gradient descent. *arXiv preprint arXiv:1907.10300*, 2019.

[4] Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046, 2018.

[5] De Castro, Gadat, and Marteau. Fastpart: Over-parameterized stochastic gradient descent for sparse optimisation on measures. *under review*, 2024.

[6] Zhang Fan, Li and Zou. *Statistical Foundations of Data Science*. CRC Press, 2020.

[7] Koh and Boyd. An interior point method for large scale ℓ_1-regularized logistic regression. *Journal of Machine Learning Research*, 2007.

[8] Clarice Poon, Nicolas Keriven, and Gabriel Peyré. The geometry of off-the-grid compressed sensing. *Foundations of Computational Mathematics*, pages 1–87, 2021.

[9] Scholkopf and Smola. *Learning with kernels*. MIT Press, 2002.

[10] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.