Clustering en ligne : le point de vue PAC-bayésien

Le Li [1], Benjamin Guedj [2] & Sébastien Loustau [3]

[1] *Université d'Angers & iAdvize, le@iadvize.com*
[2] *Équipe-projet MODAL, Inria, benjamin.guedj@inria.fr*
[3] *Université d'Angers, loustau@math.univ-angers.fr*

Résumé. Nous nous intéressons dans ce travail à la construction et à la mise en oeuvre d'une méthode de *clustering* en ligne. Face à des flux de données massives, le *clustering* est une gageure tant d'un point de vue théorique qu'algorithmique. Nous proposons un nouvel algorithme de *clustering* en ligne, reposant sur l'approche PAC-bayésienne. En particulier, le nombre de *clusters* est estimé dynamiquement (c'est-à-dire qu'il peut changer au cours du temps), et nous démontrons des bornes de regret parcimonieuses. De plus, un algorithme via RJMCMC, appelé Paco est présenté, et ses performances sur données simulées seront commentées.

Mots-clés. Bornes de regret parcimonieuses, Clustering en ligne, Reversible Jump MCMC, Théorie PAC-bayésienne.

Abstract. We address the online clustering problem. When faced with high frequency streams of data, clustering raises theoretical and algorithmic pitfalls. Working under a sparsity assumption, a new online clustering algorithm is introduced. Our procedure relies on the PAC-Bayesian approach, allowing for a dynamic (*i.e.*, time-dependent) estimation of the number of clusters. Its theoretical merits are supported by sparsity regret bounds, and an RJMCMC-flavored implementation called Paco is proposed along with numerical experiments to assess its potential.

Keywords. Online clustering, PAC-Bayesian theory, Reversible Jump Markov Chain Monte Carlo, Sparsity regret bounds.

1 Introduction

Online learning has been extensively studied these last decades in game theory and statistics, see [1]. The problem could be described as a sequential game: a blackbox reveals at each time t some $z_t \in \mathcal{Z}$. Then, the forecaster predicts the next value based on the past observations and possibly other available information. The difference with the classical statistical framework lies in the fact that the sequence (z_t) is not assumed to be a realization of some stochastic process. Research efforts in online learning began in the framework of prediction with experts advices. In this setting, the forecaster has access

to a set of experts' predictions. His goal is to build a sequence of predictions which are nearly as good as the best expert's predictions in the first T time rounds.

Online learning technique has also been applied to the regression framework. In this setting, the forecaster gives a prediction of dependent variable, using only newly revealed regressor and past observations. A possible goal is to build a forecaster whose performance is nearly as good as the best linear forecaster. It has been addressed by numerous contributions to the literature. For example, algorithms close to the ridge regression with a remainder term growing logarithmically with time horizon T have been proposed. Other contributions have investigated the Gradient-Descent algorithm and the Exponentiated Gradient Forecasters. In the so-called high dimensional setting, a sparsity regret bound with a remainder term growing logarithmically with T and dimension d of regressor is proved by Gerchinovitz [5].

The ambition of our work is to transpose the aforecited framework to the clustering problem. Online clustering has attracted some attention from the machine learning and streaming communities. The latter study the so-called data streaming clustering problem which amounts to cluster online data to a fixed number of groups in a single pass, or a small number of passes, while using little memory. From a machine learning perspective, to the best of our knowledge, Loustau [4] is the first attempt to perform online clustering with an unfixed K, which serves as a starting point to our work. Our strategy consists in two steps:

1. The Gibbs quasi-posterior which depends on the choice of a sparsity-inducing quasi-prior. This approach is motivated by the PAC-Bayesian theory which has been extensively studied by numerous researchers in the batch setting and developed by Audibert [3], Loustau [4] and Gerchinovitz [5] in the online setting, among others.

2. The Reversible Jump MCMC algorithm which allows the effective simulations from the Gibbs quasi-posterior of the complex-structured space.

Our PAC-Bayesian Online clustering method is fully presented in [2].

2 Notation

Let $(x_t)_{1:T}$ be an online dataset, where $x_t \in \mathbb{R}^d$ and $\mathcal{C} = \cup_{k=1}^{p} \mathbb{R}^{dk}$ for some integer $p \geq 1$. Our goal is to learn a time-dependent parameter K_t and a partition of the observed points into K_t ($K_t \leq p$) cells, for any $t = 1, \ldots, T$. To this aim, the output of our algorithm at time t is a vector $\hat{\mathbf{c}}_t = (\hat{c}_{t,1}, \hat{c}_{t,2}, \ldots, \hat{c}_{t,K_t}) \in \mathcal{C}$, depending on the past information $(x_s)_{1:(t-1)}$ and $(\hat{\mathbf{c}}_s)_{1:(t-1)}$. A partition is then fulfilled by assigning $(x_s)_{1:(t-1)}$ to its closest center. When x_t is revealed, the instantaneous loss is computed as

$$\ell(\hat{\mathbf{c}}_t, x_t) = \min_{1 \leq k \leq K_t} |\hat{c}_{t,k} - x_t|_2^2,$$

where $|\cdot|_2$ is the ℓ_2-norm in $\mathbb{R}^d$.

In the sequel, denote by π a quasi-prior on $\mathcal{C}$ and let $\lambda > 0$ be some (inverse temperature) parameter. The output $\hat{\mathbf{c}}_{t+1}$ is constructed as follows: at each time t, we observe x_t and a random partition $\hat{\mathbf{c}}_{t+1} \in \mathcal{C}$ is sampled from the Gibbs quasi-posterior $\hat{\rho}_{t+1}$ in Algorithm 1 below.

Algorithm 1 The PAC-Bayesian online clustering algorithm

1: **Input parameters**: $p > 0, \pi$, $\lambda > 0$ and $S_0 \equiv 0$
2: **Initialization**: Draw $\hat{\mathbf{c}}_1 \sim \pi$
3: **For** $t = 1, \ldots, (T-1)$
4: Get the data x_t
5: Draw $\hat{\mathbf{c}}_{t+1} \sim \hat{\rho}_{t+1}(\mathbf{c})$ where $\mathrm{d}\hat{\rho}_{t+1}(\mathbf{c}) \propto \exp\big(-\lambda S_t(\mathbf{c})\big)\mathrm{d}\pi(\mathbf{c})$, and

$$S_t(\mathbf{c}) = S_{t-1}(\mathbf{c}) + \ell(\mathbf{c}, x_t) + \frac{\lambda}{2}\big(\ell(\mathbf{c}, x_t) - \ell(\hat{\mathbf{c}}_t, x_t)\big)^2.$$

6: **End for**

Note that the last term in step 5 is a consequence of the non-convexity of the loss ℓ (see [3]) and that the partition $\hat{\mathbf{c}}_{t+1}$ is a realisation of $\hat{\rho}_{t+1}$.

3 Sparsity regret bound and implementation

We present here our main theoretical result.

Theorem 1. *For any sequence $(x_t)_{1:T} \in \mathbb{R}^{dT}$ and any $p \geq 1$, there exists a quasi-prior π and a λ such that the procedure described in Algorithm 1 satisfies*

$$\sum_{t=1}^{T} \mathbb{E}_{(\hat{\rho}_1, \hat{\rho}_2, \ldots, \hat{\rho}_t)} \ell(\hat{\mathbf{c}}_t, x_t) \leq \inf_{k \in \{1,\ldots,p\}} \left\{ \inf_{\mathbf{c} \in \mathcal{C}(k,R)} \sum_{t=1}^{T} \ell(\mathbf{c}, x_t) + C_1 k \sqrt{T} \log(T) \right\} + (C_2 + \log p)\sqrt{T}.$$

where $\mathcal{C}(k,R) = \big\{\mathbf{c} = (c_j)_{j=1,\ldots,k} \in \mathbb{R}^{dk},$ such that $|c_j|_2 \leq R \quad \forall j\big\}$ for $k \in \{1, \ldots, p\}$ and $R \geq \max_{t=1,\ldots,T} |x_t|_2$. C_1 and C_2 are constants.

This theorem indicates that if there exist $k^\star \in \{1, \ldots, p\}$ and $\mathbf{c}^\star \in \mathcal{C}(k^\star, R)$ which achieve the infimum above, then the regret between the expected cumulative loss of our algorithm and the oracle cumulative loss is upper bounded by a term of order $\sqrt{T}\log T$. The implementation required to sample at each t from the Gibbs quasi-posterior $\hat{\rho}_{t+1}$. Since $\hat{\rho}_t$ is defined on the massive and complex-structured space $\mathcal{C}$ (let us recall that $\mathcal{C}$ is a union of heterogeneous spaces), direct sampling from $\hat{\rho}_t$ is not an option and is much rather an algorithmic challenge. Our approach consists in approximating $\hat{\rho}_t$ through a version of RJMCMC (Algorithm 2, more details see [2]).

Algorithm 2 PACO

Initialization: (λ_t)
For $t \in [\![1, T]\!]$
Initialization: $\left(k^{(0)}, \mathbf{c}^{(0)}\right) \in [\![1, p]\!] \times \mathbb{R}^{dk^{(0)}}$
For $n \in [\![0, N-1]\!]$
    Given $k^{(n)}$, draw $k' \sim q(k^{(n)}, \cdot)$, where $q(k^{(n)}, \cdot)$ is a conditional distribution on $[\![k^{(n)} - 1, k^{(n)} + 1]\!]$.
    Let $\mathfrak{c}_{k'} \leftarrow$ standard k-means output with k' centers.
    Let $\tau_{k'} = 1/\sqrt{pt}$.
    Sample $v_1 \sim \rho_{k'}(\cdot, \mathfrak{c}_{k'}, \tau_{k'})$, where $\rho_{k'}(\cdot, \mathfrak{c}_{k'}, \tau_{k'})$ is a distribution on $\mathbb{R}^{dk'}$ with parameters $\mathfrak{c}_{k'}$ and $\tau_{k'}$.
    Let $(v_2, \mathbf{c}') = g(v_1, \mathbf{c}^{(n)})$, where $g : (x, y) \in \mathbb{R}^{dk'} \times \mathbb{R}^{dk^{(n)}} \to (y, x) \in \mathbb{R}^{dk^{(n)}} \times \mathbb{R}^{dk'}$ is an one to one, first order derivative mapping.
    Accept the move $(k^{(n)}, \mathbf{c}^{(n)}) = (k', \mathbf{c}')$ with probability

$$\alpha\left[(k^{(n)}, \mathbf{c}^{(n)}), (k', \mathbf{c}'))\right] = \min\left\{1, \frac{\hat{\rho}_t(\mathbf{c}')q(k', k^{(n)})\rho_{k^{(n)}}(v_2, \mathfrak{c}_{k^{(n)}}, \tau_{k^{(n)}})}{\hat{\rho}_t(\mathbf{c}^{(n)})q(k^{(n)}, k')\rho_{k'}(v_1, \mathfrak{c}_{k'}, \tau_{k'})} \left|\frac{\partial g(v_1, \mathbf{c}^{(n)})}{\partial v_1 \partial \mathbf{c}^{(n)}}\right|\right\}$$

    Else $(k^{(n+1)}, \mathbf{c}^{(n+1)}) = (k^{(n)}, \mathbf{c}^{(n)})$.
End for
Let $\hat{\mathbf{c}}_t = \mathbf{c}^{(N)}$.
End for

The performance of Algorithm 2 on a simulated dataset is shown in Figure 1, where the black lines represent the true number of cells augmenting each 20 time trial while the red crosses represent the estimates. We can see that, after a cold start, PACO almost perfectly identifies the true number of cells.

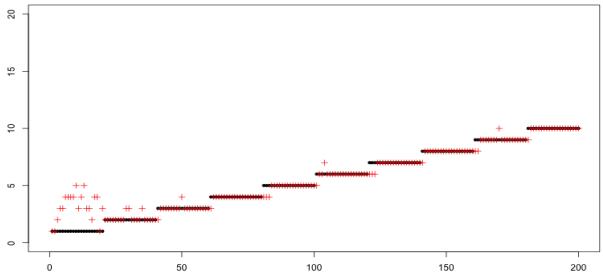

Figure 1: Estimated number of cells by PACO as a function of t. Black lines represent the true number and red dots the estimates.

Bibliographie

[1] N. Cesa-Bianchi and G. Lugosi (2006), *Prediction, learning and Games*, Cambridge University Press, New York.
[2] L. Li, B. Guedj and S. Loustau (2016), *PAC-Bayesian Online Clustering*, HAL preprint https://hal.inria.fr/hal-01264233.
[3] J. Y. Audibert (2009), *Fast learning rates in statistical inference through aggregation*, The Annals of Statistics, 37(4): 1591–1646.
[4] S. Loustau (2014), *Online clustering of individual sequence*, HAL preprint https://hal.archives-ouvertes.fr/hal-00943384.
[5] S. Gerchinovitz (2011), *Prédiction de suites individuelles et cadre statistique classique : étude de quelques liens autour de la régression parcimonieuse et des techniques d'agrégation*, PhD thesis, Université Paris-Sud.