PAC-BAYESIAN ONLINE CLUSTERING

Apprentissage en Ligne et agrégation

Assitan Diarra
Flora Ziadi

MS Data Science
Année 2017 – 2018

Plan

- L’algorithme de clustering en ligne PAC-Bayesien
- Sparsity Regrets Bounds
- Comparaison avec l’Algorithme EWA (Agrégation à poids exponentiels)
- L’algorithme PACO
- Application numérique
- Annexes

Algorithme de clustering en ligne : PAC-Bayesien

- Objectif : Evaluer à l'instant $t\epsilon[1,T]$, l'ensemble des clusters $\hat{c}_t = (\hat{c}_{t,1}, \hat{c}_{t,2}, \dots, \hat{c}_{t,K_t}) \in \mathbb{R}^{dK_t}$, où K_t représente le nombre de clusters.

- $(x_t)_{1:T}$ la base de données en ligne $(x_t \in \mathbb{R}^{d})$

- Lorsqu'un nouveau x_t est révélé, on calcule **la fonction de perte** entre le cluster le plus proche (le centre) et x_t .

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

- On définit les deux lois de probabilités suivantes :
  - q permet de simuler le nombre de cluster K_t à l'instant t sur l'ensemble $[1,p]$,
  - $\pi(c)$ permet de simuler la répartition (quasi) à priori de l'ensemble des clusters $\hat{c}_t$.

Algorithme de clustering en ligne : PAC-Bayesien

Algortihme 1: Clustering en ligne PAC-Bayesien

Input:

- Le nombre maximum de clusters p
- La distribution a priori π
- Le paramètre $\lambda > 0$
- On initialise la perte cumulée à 0 $S_0 = 0$

Initialisation:

- On simule l'ensemble des clusters $\hat{c}_1$ par la distribution à priori $\pi = \hat{\rho}_1$

For $t \epsilon [1, T-1]$:

- On a une nouvelle ligne d'observations x_t
- On calcule la **fonction de perte cumulée S_t** , qui correspond à la perte depuis le début de l'algorithme.

$$S_t(c) = S_{t-1}(c) + l(c, x_t) + \frac{\lambda}{2}\left(l(c, x_t) - l(\hat{c}_t, x_t)\right)^2$$

- On génère l'ensemble des clusters $\hat{c}_{t+1}$ selon la **distribution a posteriori de Gibbs $\hat{\rho}_{t+1}$**

$$d\hat{\rho}_{t+1}(c) \propto \exp\left(-\lambda S_t(c)\right) d\pi(c)$$

Elle dépend de la distribution a priori π et de la perte cumulée S_t.

Sparsity Regrets Bounds

- Objectif : Déterminer la limite autour de la somme des erreurs du clustering en ligne.

- Le regret représente la différence entre l'erreur de notre prédiction et la meilleure prédiction possible. Le premier théorème de l'article énonce que ce regret peut être borné par une pénalité composée de :
  - La fonction de Kullback-Leibler $K(\rho,\pi)$
  - L'écart quadratique de nos erreurs suivant les distributions a posteriori de Gibbs utilisées lors de notre algorithme. Ce terme est une conséquence de la non-convexité de la perte l (Audibert 2009), en d'autres terme si la perte l est convexe, ce terme est alors nul.

- L'expression du Théorème 1 est la suivante :

$$\sum_{t=1}^{T} E_{(\hat{\rho}_1,\dots,\hat{\rho}_t)} l(\hat{c}_t, x_t) \leq \inf_{\rho \epsilon \mathrm{P}_\pi(C)} \left\{ E_{c\sim\rho}\left[\sum_{t=1}^{T} l(c, x_t)\right] + \frac{K(\rho,\pi)}{\lambda} + \frac{\lambda}{2} E_{(\hat{\rho}_1,\dots,\hat{\rho}_T)} E_{c\sim\rho} \sum_{t=1}^{T} [\, l(c, x_t) - l(\hat{c}_t, x_t)]^2 \right\}$$

Le but ici est de minimiser cette pénalité pour laquelle sera démontré la sous-linéarité en T.

Sparsity Regrets Bounds

- En prenant les hypothèses suivantes, on arrive à expliciter la pénalité.
  - q qui simule le nombre de clusters k suivra une loi exponentielle décroissante discrète,
  - π_k est définit comme le produit de k distributions uniformes sur des boules dans R^d centrées en 0 et de rayon R.

- De plus, si $\lambda = (d+2)/(2\sqrt{T}R^2)$, $R \geq \max_{t=1,\dots,T} |x_t|_2$ et si la borne inférieure est atteinte (c^*, k^*) il est possible d'écrire cette limite telle que :

$$\sum_{t=1}^{T} E_{(\hat{\rho}_1,\dots,\hat{\rho}_t)}\, l(\hat{c}_t, x_t) - \sum_{t=1}^{T} l(c^*, x_t) \leq J\, k^* \sqrt{T}\, \log(T)$$

  Avec J dépendant de d, R, p.

  On remarque ainsi que la borne est sous-linéaire en T.

- **Algorithme 2 :** The adaptive PAC-Bayesian online clustering algorithm
  Cette algorithme est identique à l'algorithme 1, **on remplace seulement λ par λ_t.**

  Comme T est inconnu, il est donc nécessaire de rendre variable λ de manière à ce que la correction de la distribution diminue en fonction du temps.

Sparsity Regrets Bounds

- Pourquoi parlons nous de parcimonie ? Un modèle statistique parcimonieux est un modèle pour lequel un nombre relativement faible de paramètres joue un rôle important.

- En définissant une loi q pour simuler le nombre de clusters et une loi π_k pour simuler la distribution, l'article montre que la pénalité exprimée précédemment est linéairement croissante avec k et croissante avec p, où k est le nombre de clusters et p la valeur maximum prise par k.

- Or, plus k et p augmentent, plus les termes de la pénalités ainsi que la borne de regret augmentent également. Ainsi les choix vers des modélisations plus complexes seront sanctionnées par un accroissement de la pénalité que nous cherchons ici à minimiser.

- Afin de favoriser un modèle parcimonieux et limiter la pénalité, l'algorithme PAC Bayésien devra utiliser une distribution a priori favorisant les modèles avec un nombre plus faible de paramètres (k et p petits).

Comparaison avec l'algorithme EWA

Les hypothèses

Hypothèses	Article PAC Bayésien	Cours ENSAE
Prédicteurs	Les prédicteurs sont les clusters.	M méthodes de prédiction dans le cas fini et une famille de prédicteurs dans le cas général.
Fonction de perte	La fonction de perte s'exprime avec la norme $\|\| \|\|_2^2$	La fonction de perte est « seulement» bornée par B. $0 \leq l(.,.) \leq B$
Distribution a priori	Distingue le nombre de clusters et leurs distributions permettant un paramètrage plus fin de nos prédictions.	Pas de précision sur la distribution a priori.
Distribution a posteriori	Distribution a posteriori de Gibbs avec la somme cumulée des pertes incrémentée d'une composante quadratique.	Distribution a posteriori de Gibbs ainsi que de l'hypothèse de Hoeffding

Dans la slide suivante, nous comparons les limites obtenues qui dépendent en toute logique des hypothèses ci-dessus.

Comparaison avec l'algorithme EWA

Regrets Bounds (sous hypothèse de la loi uniforme)

Composantes	Article PAC Bayésien	Cours ENSAE
La divergence de Kullback-Leibler $K(.,.)$ Sous l'hypothèse d'une distribution uniforme, on arrive à expliciter $K(.,.)$	$\mathrm{K}(\rho,\pi) \leq d\sum_{j=1}^{k}\log\left(\frac{2R}{\varepsilon_j}\right)+\eta(\mathrm{k}-1)+\log p$ (cf. preuve du corollaire 1 p.20)	La divergence de Kullback-Leibler $K(.,.)$ s'exprime simplement comme $\log\ (M)$
Terme quadratique Ce terme disparaît sous hypothèse de convexité de la perte l	Le terme s'exprime comme $\frac{\lambda}{2}E_{(\hat{\rho}_1,\ldots,\hat{\rho}_T)}E_{c\sim\rho}\sum_{t=1}^{T}[l(c,\hat{x}_t)-(l(\hat{c}_t,x_t)]^2 \leq \frac{\lambda T(C_1)^2}{2}$ Où $C_1=(2R+\max_{t=1,\ldots,T}\lvert x_t\rvert_2)^2$ (cf. preuve du corollaire 1 p.20)	En prenant B la borne supérieure de la fonction de perte et à l'aide de l'hypothèse d'Hoeffding le second terme de la « regret bound » s'exprime comme $\frac{\eta T B^2}{8}$ Où η est le paramètre inverse de température.

L'algorithme PACO

- Limite de l'algorithme 1 et 2 : En pratique, la distribution quasi-posteriori de Gibbs $\hat{\rho}_t$est en général impossible à simuler.

- Solution : Approximer $\hat{\rho}_t$ à l'aide de l'algorithme MCMC sous la contrainte de favoriser les déplacements locaux de la chaîne de Markov (RJMCMC).

- L'agorithme RJMCMC (Reversible Jump Makov Chain Monte Carlo) : En statistique computationnelle, cet algorithme permet de **simuler des valeurs d'une distribution sur des espaces de dimensions variables** et est une extension de l'algorithme de Metropolis-Hastings.

L'algorithme PACO

Présentation de l'algorithme PACO :

- Nous définissons $(k^{(n)}, c^{(n)})_{0 \le n \le N}$ l'état de la Chaine de Markov de longueur N
  - $k^{(n)} \in [\![1, p]\!]$ le nombre de clusters à l'étape n
  - $c^{(n)}$ les centres des clusters à l'étape n
- Ainsi, $c^{(n)} \in R^{dk^{(n)}}$ et $c' \in R^{dk'}$ peuvent appartenir à des espaces différents.
- On introduit alors deux vecteurs $v_1 \in R^{d1}$ et $v_2 \in R^{d2}$ $(d_1, d_2 \geq 1)$ pour compenser la différence des dimensions :

$$dk^{(n)} + d_1 = dk' + d_2$$

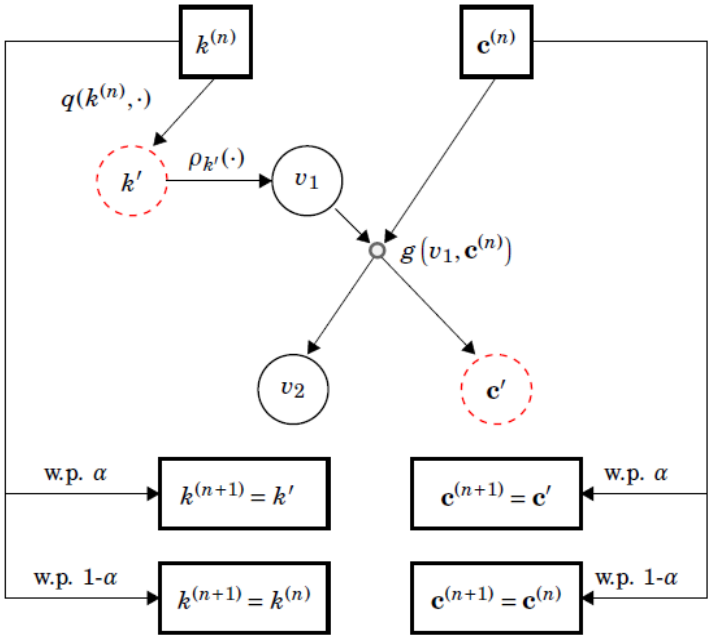


Figure 1: General structure of PACO.

L'algorithme PACO

Présentation de l'algorithme PACO :

1. Générer k' tel que $k' \in \left[k^{(n)} - 1 \,;\, k^{(n)} + 1 \right]$
2. Générer un vecteur ϑ_1 de dimension dk' à partir de $\rho_{k'}$ sur $R^{dk'}$
3. Calculer la paire $(\vartheta_2, c') = g\left(\vartheta_1, c^{(n)}\right)$ en choisissant $g : (x, y) \in R^{dk'} * R^{dk^{(n)}} \rightarrow (y, x) \in R^{dk^{(n)}} * R^{dk'}$
4. La proposition est alors acceptée avec la probabilité :

$$\alpha = \min\left\{1, \frac{\hat{\rho}_t(c')q(k', k^{(n)})\rho_{k^{(n)}}(\vartheta_2)}{\hat{\rho}_t(c^{(n)})q(k^{(n)}, k')\rho_{k'}(\vartheta_1)} \left| \frac{\partial\, g\left(\vartheta_1, c^{(n)}\right)}{\partial \vartheta_1\, \partial\, c^{(n)}} \right| \right\}$$

Algorithm 3 PACO

Initialization: (λ_t)
For $t \in [1, T]$
Initialization: $(k^{(0)}, \mathbf{c}^{(0)}) \in [1, p] \times \mathbb{R}^{dk^{(0)}}$
For $n \in [1, N-1]$
  Sample $k' \in [k^{(n)} - 1, k^{(n)} + 1]$ from $q(k^{(n)}, \cdot) = \frac{1}{3}$.
  Let $c' \leftarrow$ standard k-means output.
  Let $\tau' = 1/\sqrt{pt}$.
  Sample $v_1 \sim \rho_{k'}(\cdot, c_{k'}, \tau_{k'})$, .
  Let $(v_2, \mathbf{c}') = g(v_1, \mathbf{c}^{(n)})$.
  Accept the move $(k^{(n)}, \mathbf{c}^{(n)}) = (k', \mathbf{c}')$ with probability

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

  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

L'algorithme PACO

Convergence de PACO vers la distribution quasi posteriori de Gibbs :

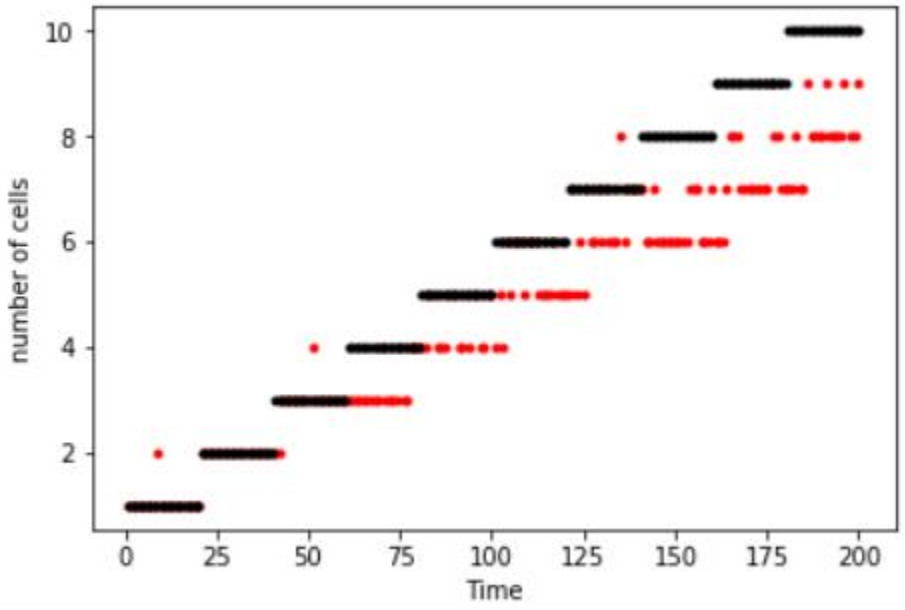
- L'objectif est de montrer que l'algorithme PACO construit une chaine de Markov dont la **distribution invariante** est précisément la distribution quasi posteriori de Gibbs $\hat{\rho}_t$ quand $N \to +\infty$.

- Le théorème 3 énonce que l'algorithme de PACO fournit une série $(c^{(n)})_{1:N}$ qui respecte ces conditions.
  - $\hat{\rho}_t$ –irréductible
  - Apériodique
  - Harris récurrente

Application numérique : Implémentation de l'algorithme PACO

- Calibration du rayon $\mathcal{R}$ et du paramètre λ:
  - $\mathcal{R} \geq max \ \|x\|_2$
  - $\lambda = 0.6 * \frac{d+2}{2\sqrt{t}}$
- Choix du langage: python
- Les paramètres de l'algorithme :
  - Les données sous forme de numpy array
  - Le rayon $\mathcal{R}$
  - Le nombre de cluster maximum
  - Le nombre d'itérations
- Les sorties de l'algorithme:
  - Les centres des clusters
  - Le nombre de cluster à chaque instant

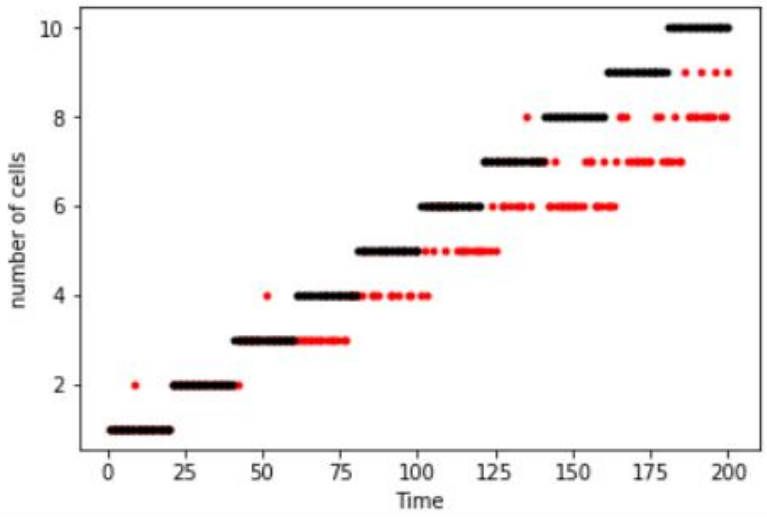
Application numérique : Résultats de la simulation de l'algorithme PACO



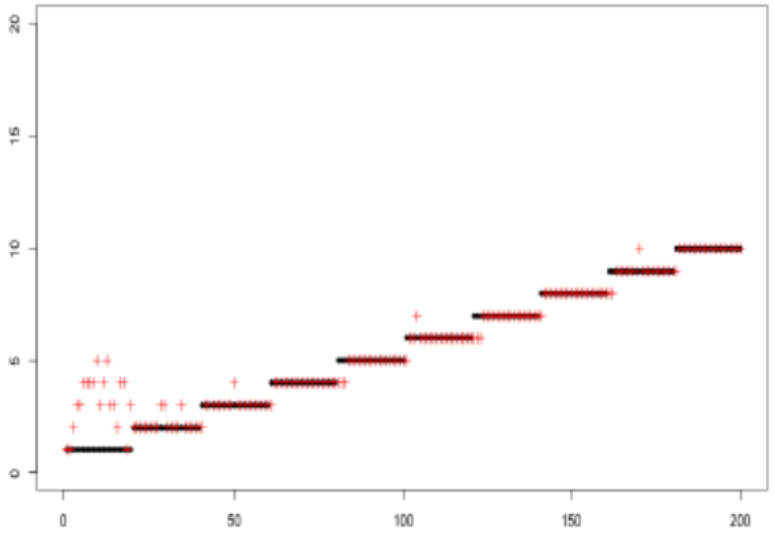
- Paramètres:
  - Modèle 6
  - $\mathcal{R} = max\ \|x\|_2$
  - Le nombre de clusters maximum: 50
  - Le nombre d'itérations: 100

Application numérique : Comparaison des résultats

Résultats obtenus



Résultats de l'article



Annexes

- Comparaison détaillée avec le cours
- Apprentissage en ligne
- Quelques définitions

Annexes : Comparaison détaillée avec le cours

Algorithme EWA (dans le cas général) :

- $\mathcal{F}$ une famille de prédicteurs $X \to Y$
- $(x_1, y_1), \ldots, (x_T, y_T)$ quelconques
- Le regret est définit à la date T de la façon suivante :

$$R_T(F) = \sum_{t=1}^{T} l(\hat{y}_t, y_t) - \inf_{f \in F} \sum_{t=1}^{T} \underbrace{l(f(x_t), y_t)}_{g_t(\mathrm{f})}$$

- Le premier terme correspond à l'erreur cumulée et le second correspond l'erreur cumulée obtenue par le prédicteur qui minimise cette erreur.

- L'objectif est de fabriquer une stratégie pour laquelle on contrôle le regret, c'est-à-dire trouver une borne supérieure $v(T)$ tel que :

$$R_T(F) \leq v(T)$$

Annexes : Comparaison détaillée avec le cours

Algorithme EWA

- p_i loi sur $\mathcal{F}$, $\eta > 0$
- A la date t :
  - x_t est donné
  - On génère les prédicteurs selon la loi p_t, puis on applique f sur les données : $f \sim p_t$, $\hat{y}_t = f(x_t)$
  - y_t est révélé
  - On mesure l'erreur qu'on réinjecte dans p_{t+1} :

$$p_{t+1}(df) = \frac{\exp(-\eta g_t(\mathrm{f}))p_t(df)}{\int \exp(-\eta g_t(\mathrm{h}))p_t(dh)}$$

⇨ $p_{t+1}(df) \propto \exp(-\eta g_t(\mathrm{f}))p_t(df)$

PAC-Bayesian Online Clustering

- $\pi = \hat{\rho}_1$ loi sur $\mathcal{P}(\mathcal{C})$, $\lambda > 0$
- A la date t :
  - x_t est donné
  - On génère les clusters selon la loi $\hat{\rho}_t$:

$$\hat{c}_{t+1} \sim \hat{\rho}_{t+1}(c)$$

  - On mesure l'erreur qu'on réinjecte dans $\hat{\rho}_{t+1}$:

$$\hat{\rho}_{t+1} = \frac{\exp\left(-\lambda S_t(\mathrm{c})\right) d\pi(c)}{\int \exp\left(-\lambda S_t(\mathrm{c})\right) d\pi(c)}$$

⇨ $\hat{\rho}_{t+1}(c) \propto \exp\left(-\lambda S_t(\mathrm{c})\right) d\pi(c)$

Annexes : Comparaison détaillée avec le cours

Algorithme EWA

- La divergence de Kullback-Leibler est la suivante :

$$\mathcal{K}(\mu,\upsilon) = \begin{cases} \int \log\left(\frac{d\upsilon}{d\mu}(f)\mu(df)\right) \\ +\infty \end{cases}$$

avec μ, υ deux probabilités sur $\mathcal{F}$.

- **Lemme :** Si $h : \mathcal{F} \to \mathbb{R}$ bornée et π loi sur $\mathcal{F}$, on a :

$$\sup_{\mu}\left[\int h(f)\mu(df) - \mathcal{K}(\mu,\pi)\right] = \log\int \exp\big(h(f)\big)\,\pi(df)$$

$$\Leftrightarrow \inf_{\mu}\left[\int h(f)\mu(df) - \mathcal{K}(\mu,\pi)\right] = -\log\int \exp\big(-h(f)\big)\,\pi(df)$$

et la borne inf est atteinte pour $\mu = \pi_h$

$$\pi_h = \frac{\exp\big(h(f)\big)\,\pi(df)}{\int \exp\big(h(f)\big)\,\pi(df}$$

PAC-Bayesian Online Clustering

- La divergence de Kullback-Leibler est la suivante :

$$\mathcal{K}(\rho,\pi) = \begin{cases} \int_{\Theta} \log\left(\frac{d\rho}{d\pi}\right) d\pi & si\ \rho\epsilon\mathcal{P}_{\pi}(\Theta) \\ +\infty & sinon \end{cases}$$

avec ρ, π deux probabilités sur $\mathcal{P}(\Theta)$.

- **Lemme :** Si $h : \Theta \to \mathbb{R}$ bornée et ρ loi sur $\mathcal{P}(\Theta)$, on a :

$$\inf_{\rho\epsilon\mathcal{P}_{\pi}(\Theta)}\left[\int_{\Theta} h d\rho + \mathcal{K}(\rho,\pi)\right] = -\log\int_{\Theta}\exp(-h)\,d\pi$$

La borne inf est atteinte pour $\hat{\rho}$ (distribution quasi posteriori de Gibbs)

$$d\hat{\rho} = \frac{\exp(-h)\,d\pi}{\int \exp(-h)\,d\pi}$$

Annexes : Comparaison détaillée avec le cours

On obtient au final le théorème suivant :

Algorithme EWA

- **Théorème :** si la perte l est bornée tel que $0 \leq l(.,.) \leq B$, alors l'algorithme EWA satisfait la condition suivante :

$$\mathrm{E}\left(\sum_{t=1}^{T} l(\hat{y}_t, y_t)\right) \leq$$

$$\inf_{\mu}\left\{\int \sum_{t=1}^{T} l(f(x_t), y_t)\, \mu(df) + \frac{\mathcal{K}(\mu, p_1)}{\eta} + \frac{\eta\, T\, B^2}{8}\right\}$$

PAC-Bayesian Online Clustering

- **Théorème :** L'algortihme PAC-Bayesian Online Clustering satisfait la condition suivante :

$$\sum_{t=1}^{T} E_{(\hat{\rho}_1,\ldots,\hat{\rho}_t)} l(\hat{c}_t, x_t) \leq \inf_{\rho \epsilon \mathrm{P}_\pi(C)} \left\{ E_{c\sim\rho}\left[\sum_{t=1}^{T} l(c, x_t)\right] + \frac{K(\rho,\pi)}{\lambda} + \frac{\lambda}{2} E_{(\hat{\rho}_1,\ldots,\hat{\rho}_T)}\, E_{c\sim\rho} \sum_{t=1}^{T} [\, l(c, x_t) - l(\hat{c}_t, x_t)]^2 \right\}$$

- Sous l'hypothèse de la distribution uniforme π_k définit précédemment à la slide 7, on peut majorer le terme quadratique et ainsi réécrire le théorème de la façon suivante :

$$\sum_{t=1}^{T} E_{(\hat{\rho}_1,\ldots,\hat{\rho}_t)} l(\hat{c}_t, x_t) \leq \inf_{\rho \epsilon \mathrm{P}_\pi(C)} \left\{ E_{c\sim\rho}\left[\sum_{t=1}^{T} l(c, x_t)\right] + \frac{K(\rho,\pi)}{\lambda} + \frac{\lambda T (C_1)^2}{2} \right\}$$

Annexes : Apprentissage en ligne

- Une boîte noire révèle à chaque instant t des observations $z_t \in Z$ et les prédictions $\hat{z}_t$ sont faites sur les observations passées z_{t-1} et d'autres informations disponibles.

Prédiction avec conseil d'expert :

- Le prévisionniste dispose d'un ensemble de prévision d'experts $\{f_{e,t} \in D : e \in \mathcal{E}\}$, où $\mathcal{E}$ est un ensemble fini d'experts.
- L'objectif est de construire une séquence de prédictions $(\hat{z}_t)_{1:T}$ qui est presque aussi bonne que les prédictions d'experts, i.e. qui satisfait uniformément la limite du regret suivante :

$$\sum_{t=1}^{T} l(\hat{z}_t, z_t) - \min_{e \epsilon \mathcal{E}} \left\{ \sum_{t=1}^{T} l\left(f_{e,t}, z_t\right) \right\} \leq \Delta_T(\varepsilon)$$

  Où l est la fonction de perte et $\Delta_T(\varepsilon)$ est le terme restant.
- $\Delta_T(\varepsilon)$ doit-être le plus petit possible et en particulier sous-linéaire en T.
- Si $|\varepsilon| < \infty$, l est borné et convexe dans sont premier terme, alors $\Delta_T(\varepsilon) = \sqrt{(T/2) log|\varepsilon|}$

Regression en ligne :

- Il s'agit ici d'un problème d'apprentissage en ligne où l'algorithme est mis à jour au fur et à mesure que les données deviennent disponibles.
- $z_t = (x_t, y_t) \in \mathbb{R}^d \times \mathbb{R}$, on prédit $\hat{y}_t$ à partir de x_t et des observations passées $(x_s, y_s)_{1:t-1}$

PAC-Bayesian :

- La méthode utilisée pour calculer la limite de l'algorithme est le PAC-Bayesian.

Annexes : quelques définitions

- **Sparsity regret bounds :** a sparse model statistical model is one in which only a relatively small number of parameters (or predictors) play an important role. Conexity greatly simplifies the computation as does the sparsity assumption itself.

- **MCMC :** Un algorithme Markov Chain Monte Carlo (MCMC) est un algorithme stochastique qui permet de simuler une distribution à l'aide d'une chaîne de Markov.

- **Algorithme de Metropolis-Hastings :**

Algorithm A.24 –Metropolis–Hastings–

Given $x^{(t)}$,

1. Generate $Y_t \sim q(y|x^{(t)})$.
2. Take

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability } \rho(x^{(t)}, Y_t), \\ x^{(t)} & \text{with probability } 1 - \rho(x^{(t)}, Y_t), \end{cases}$$

where [A.24]

$$\rho(x, y) = \min \left\{ \frac{f(y)}{f(x)} \frac{q(x|y)}{q(y|x)}, 1 \right\} .$$

Annexes : quelques définitions

- **Harris récurrente :** Une chaîne de Harris est une chaîne de Markov où la chaîne retourne à une partie particulière de l'espace d'état un nombre illimité de fois.

- **Distribution invariante d'une Chaine de Markov :**

  Si P est une matrice stochastisque et irréductible, il existe donc un vecteur de probabilité unique π tel que $\pi P = \pi$ et $\pi 1 = 1$. π est strictement positif et appelé le vecteur de probabilités invariantes de la chaîne de Markov.

$$\pi_n = \pi_0 P^n = \pi P^n = (\pi P) P^{n-1} = \pi P^{n-1} = \pi$$

Références

- Cours Apprentissage en ligne et Aggrégation, Pierre Alquier
- MCMC : http://www.math-info.univ-paris5.fr/~ebirmele/depots/Enseignements/MCMC.pdf
- Online Regression definition : https://en.wikipedia.org/wiki/Online_machine_learning
- Variance Inequality (2ème terme de la limite est nul si convexe) : https://arxiv.org/pdf/0909.1468.pdf
- Definition PAC Bayesian vs Inference Bayesian : https://arxiv.org/pdf/1307.2118.pdf
- Sparsity definition : https://web.stanford.edu/~hastie/StatLearnSparsity_files/SLS_corrected_1.4.16.pdf