UNIVERSITÉ DE NANTES

Master 1 Ingénierie Statistique

Optimisation stochastique

Paul Rochet

Contents

1 Introduction **2**
1.1 Définitions et notations . . . 2
1.2 Méthode d'acceptation/rejet . . . 3
1.3 Chaînes de Markov . . . 5

2 Méthodes de Monte-Carlo **8**
2.1 Exploration à l'aveugle . . . 8
2.2 Exploration aléatoire localisée . . . 10
2.3 Le cas discret . . . 11

3 Méthodes adaptatives **13**
3.1 Recuit simulé . . . 13
3.2 Algorithmes génétiques . . . 15

4 Gradient stochastique **17**
4.1 Convergence . . . 18
4.2 Application à la régression . . . 21

5 Algorithme EM **23**

1 Introduction

1.1 Définitions et notations

On s'intéresse au problème de minimiser une fonction J sur un domaine $\mathcal{X}$ et à valeurs dans $\mathbb{R}$. On supposera toujours que $\mathcal{X}$ est un sous-ensemble non-vide de $\mathbb{R}^d$, mesurable et de mesure de Lebesgue positive. On rappelle les définitions suivantes dont les notations seront utilisées implicitement dans la suite du cours.

1. La *borne inférieure* ou *infimum* de J,

$$m := \inf_{x \in \mathcal{X}} J(x) = \inf\{J(x) : x \in \mathcal{X}\} = \sup\{y \in \mathbb{R} : J(x) \geq y, \forall x \in \mathcal{X}\} \in [-\infty, +\infty[.$$

   Cette valeur n'est pas forcément atteinte, autrement dit, il n'existe pas nécessairement $x^* \in \mathcal{X}$ tel que $J(x^*) = m$ (typiquement si l'infimum est $-\infty$). Par contre, on peut trouver une suite $(x_n)_{n \in \mathbb{N}}$ d'éléments de $\mathcal{X}$ telle que $\lim_{n \to \infty} J(x_n) = m$. Clairement, l'infimum est unique. S'il est atteint, on peut parler de minimum.

2. L'*argument minimum* est l'ensemble des minimiseurs de J sur $\mathcal{X}$,

$$\mathcal{X}^* := \arg\min_{x \in \mathcal{X}} J(x) = \{x \in \mathcal{X} : J(x) = m\} \subseteq \mathcal{X}.$$

   Rigoureusement, l'argument minimum est l'ensemble des valeurs de $\mathcal{X}$ auxquelles J atteint son minimum (l'ensemble vide si l'infimum n'est pas atteint). Lorsque le minimum m est atteint en un unique point x^*, on utilise fréquemment l'abus de notation

$$\arg\min_{x \in \mathcal{X}} J(x) = x^*$$

   en identifiant l'argument minimum à l'unique *minimiseur* x^* au lieu du singleton $\{x^*\}$.

Minimiser la fonction $J(.)$ est bien sûr équivalent à maximiser $x \mapsto -J(x)$. On a alors

- $\inf_{x \in \mathcal{X}} J(x) = -\sup_{x \in \mathcal{X}}\{-J(x)\}$.
- $\arg\min_{x \in \mathcal{X}} J(x) = \arg\max_{x \in \mathcal{X}}\{-J(x)\}$.

Les algorithmes d'optimisation stochastique consistent à rechercher un minimum d'une fonction par une exploration aléatoire du domaine. Ces méthodes sont souvent simples à mettre en place et efficaces pour des problèmes de grande dimension et/ou pour des fonctions "peu" régulières, qui ont par exemple beaucoup de minima locaux. La convergence rapide de ces algorithmes se fait au détriment d'une convergence plus faible (par exemple, convergence presque sûre, en probabilité ou avec forte probabilité) au contraire des méthodes déterministes.

Un intérêt principal des algorithmes aléatoires est de ne nécessiter que l'évaluation de J (et généralement pas de gradient, exact ou approché, ou de matrice Hessienne), en plus de générer des variables aléatoires. Ainsi, les méthodes d'exploration aléatoires sont à privilégier lorsque l'évaluation de la fonction J (ou de ses dérivées) est coûteuse. En contrepartie, les garanties d'efficacité sont généralement plus faibles.

Une méthode d'optimisation stochastique va chercher le plus souvent à construire une suite de variables aléatoires $X_1, X_2, ...$ à valeurs dans $\mathcal{X}$ et qui converge en un sens probabiliste vers un minimiseur de J. Comme il n'existe en général pas de critère d'arrêt universel, on peut évaluer l'état de convergence de l'algorithme à partir du nombre N d'itérations, des progrès marginaux $J(X_n) - J(X_{n-1})$ ou encore des écarts $\|X_n - X_{n-1}\|$, si on sait que le minimiseur est unique.

1.2 Méthode d'acceptation/rejet

La première étape d'un algorithme d'optimisation stochastique est de savoir générer des points aléatoirement sur le domaine $\mathcal{X}$. La loi uniforme sur $\mathcal{X}$ est un choix naturel si aucune information n'est disponible sur le comportement de J (et que le domaine $\mathcal{X}$ est de mesure finie), mais même cette loi simple n'est pas forcément facile à générer, en particulier pour des problèmes d'optimisation sous contraintes. Le moyen standard pour y parvenir est de générer des variables aléatoires uniformément sur un rectangle (ou pavé en dimension d quelconque) qui contient le domaine $\mathcal{X}$ et de ne conserver que les variables qui tombent dans $\mathcal{X}$. Pour qu'elle soit le plus efficace possible, il est nécessaire de choisir le pavé de volume minimal pour minimiser le nombre d'itérations.

Cette approche est un cas particulier de la méthode dite d'acceptation/rejet. Plus généralement, cette méthode permet de générer une variable aléatoire X de densité f à partir d'une variable aléatoire Y de densité g telle que le rapport f/g soit borné par une constante M connue. L'algorithme est le suivant:

i) Générer U de loi uniforme sur $[0, 1]$ et Y de densité g indépendants.

ii) Calculer le ratio $R = f(Y)/g(Y)$.

iii) Si $R > M.U$, retourner $X = Y$, sinon reprendre à la première étape.

La variable X obtenue est exactement de densité f. L'efficacité de l'algorithme d'un point de vue computationnel dépend de la majoration M du rapport f/g qui doit être idéalement la plus petite possible.

Proposition 1.1 *Soit $U_1, U_2, ...$ une suite iid de variables aléatoires de lois uniformes sur $[0, 1]$ et $Y_1, Y_2, ...$ une suite iid de densité g indépendantes des U_i. On définit*

$$\tau := \inf\{n : f(Y_n)/g(Y_n) > MU_n\}.$$

Alors, la variable aléatoire $X := Y_\tau$ a pour densité f. Le nombre moyen d'itérations nécessaires pour construire X est $\mathbb{E}(\tau) = M$.

Preuve. Par indépendance, le couple (Y_i, U_i) a pour densité $(y, u) \mapsto g(y)\mathbb{1}\{u \in [0,1]\}$. On définit les variables $Z_i = \mathbb{1}\{f(Y_i)/g(Y_i) > MU_i\}$ qui sont iid de loi de Bernoulli de paramètre

$$\begin{aligned}
\mathbb{P}\big(f(Y_1)/g(Y_1) > MU_1\big) &= \iint g(y)\mathbb{1}\{u \in [0,1], f(y)/g(y) > Mu\}dydu \\
&= \int g(y)\Big[\int_0^{f(y)/Mg(y)} du\Big]dy \\
&= \int g(y)\frac{f(y)}{Mg(y)}dy = \frac{1}{M}\int f(y)dy = \frac{1}{M}.
\end{aligned}$$

Soit $X = Y_\tau$, on écrit pour tout ensemble mesurable A, $\mathbb{P}(X \in A) = \sum_{n=1}^{\infty} \mathbb{P}\big(Y_n \in A, \tau = n\big)$ avec

$$\begin{aligned}
\mathbb{P}\big(Y_n \in A, \tau = n\big) &= \mathbb{P}(Y_n \in A, Z_n = 1, Z_1, ..., Z_{n-1} = 0\big) \\
&= \Big(1 - \frac{1}{M}\Big)^{n-1}\mathbb{P}\big(Y_n \in A, f(Y_n)/g(Y_n) > MU_n\big) \\
&= \Big(1 - \frac{1}{M}\Big)^{n-1}\frac{1}{M}\int_A f(y)dy.
\end{aligned}$$

On déduit par la formules des probabilités totales:

$$\mathbb{P}(X \in A) = \frac{1}{M}\sum_{n=1}^{\infty}\Big(1 - \frac{1}{M}\Big)^{n-1}\int_A f(y)dy = \int_A f(y)dy. \qquad \square$$

Remarque. Une preuve plus rapide basée sur un argument intuitif consiste à remarquer que la densité du couple (Y_τ, U_τ) est la densité de (Y, U) conditionnée à l'événement $\{f(Y)/g(Y) > MU\}$. Cette densité vaut donc

$$(y, u) \mapsto \frac{g(y)\mathbb{1}\{f(y)/g(y) > Mu\}}{\mathbb{P}(f(Y)/g(Y) > MU)} \propto g(y)\mathbb{1}\{f(y)/g(y) > Mu\}.$$

On retrouve alors le résultat en intégrant en u sur $[0,1]$.

On remarque que la condition $f \leq Mg$ implique $M \geq 1$ avec égalité si et seulement si $f \overset{p.p.}{=} g$. La majoration M doit être connue mais la plus petite possible pour limiter le temps de calcul.

Application. Soit $\mathcal{X}$ un sous-ensemble mesurable de $\mathbb{R}^d$ de mesure de Lebesgue $\mathrm{vol}(\mathcal{X})$ finie et strictement positive. On veut générer une variable aléatoire de loi uniforme sur $\mathcal{X}$ qui a pour densité $\mathbb{1}\{. \in \mathcal{X}\}/\,\mathrm{vol}(\mathcal{X})$ sur $\mathbb{R}^d$. On choisit

$$\mathcal{R} = \{x \in \mathbb{R}^d : a_i < x_i < b_i, \forall i = 1, ..., d\}$$

tel que $\mathrm{vol}(\mathcal{X} \cap \mathcal{R}) = \mathrm{vol}(\mathcal{X})$ (autrement dit, $\mathcal{X}$ est contenu dans $\mathcal{R}$ à un ensemble de mesure nulle près). On construit alors Y par l'algorithme suivant:

i) On génère $U_1, ..., U_d$ iid de loi uniforme sur $[0, 1]$.

ii) On construit $V_i = a_i(1 - U_i) + b_i U_i$ pour $i = 1, ..., d$ et $V = (V_1, ..., V_d)^\top$.

iii) Si $V \in \mathcal{X}$, on définit $Y = V$. Sinon, on reprend à l'étape i).

Corollaire 1.2 *La variable aléatoire Y obtenue est de loi uniforme sur $\mathcal{X}$. Le nombre moyen d'itérations nécessaires pour générer Y est donné par*

$$\frac{\mathrm{vol}(\mathcal{R})}{\mathrm{vol}(\mathcal{X})} = \frac{\prod_{i=1}^d (b_i - a_i)}{\mathrm{vol}(\mathcal{X})}.$$

L'attrait majeur de cet algorithme est qu'il peut être implémenté facilement pour des problèmes d'optimisation sous contrainte. Concrètement, on génére une suite de variables aléatoires uniformes sur $\mathcal{R}$ jusqu'à ce que toutes les contraintes soient vérifiées. Cependant, la méthode est rarement utilisable en grande dimension où le rapport $\mathrm{vol}(\mathcal{R})/\,\mathrm{vol}(\mathcal{X})$ peut être très grand. Prenons par exemple comme domaine la boule unité $\mathcal{X} = \mathcal{B}(0, 1)$ de $\mathbb{R}^d$. Pour générer une variable aléatoire de loi uniforme sur $\mathcal{B}(0, 1)$, il suffit de générer $U_1, ..., U_d$ iid de loi uniforme sur $[-1, 1]$ jusqu'à ce que $\sum_{i=1}^d U_i^2 \leq 1$. La probabilité de cet événement est donnée par

$$\mathbb{P}\Big(\sum_{i=1}^d U_i^2 \leq 1\Big) = \frac{\mathrm{vol}\left(\mathcal{B}(0,1)\right)}{\mathrm{vol}\left([-1,1]^d\right)} = \frac{\pi^{d/2}}{2^d \Gamma(d/2+1)} = o\big((2.07/\sqrt{d})^d\big)$$

par la formule de Stirling.

1.3 Chaînes de Markov

La plupart des méthodes d'optimisation stochastique sont basées sur la construction d'une chaîne de Markov X_n dont la loi limite est adaptée au problème. Typiquement, la loi doit attribuer d'autant plus de poids à un point $x \in \mathcal{X}$ que la valeur $J(x)$ est petite. Un processus vérifie la propriété Markovienne (à l'ordre 1) si sa loi à l'étape $n+1$ ne dépend que de sa position à l'étape n, et pas de ce qui s'est passé avant.

Définition 1.3 *Une chaîne de Markov (sous-entendu d'ordre* 1*, homogène et en temps discret) est une suite de variables aléatoires* $X_n, n \in \mathbb{N}$ *dont la loi conditionnellement au passé est invariante dans le temps et ne dépend que du passé immédiat. Formellement, pour tout ensemble mesurable* A*,*

$$\mathbb{P}(X_n \in A | X_0, ..., X_{n-1}) = \mathbb{P}(X_n \in A | X_{n-1}) = \mathbb{P}(X_1 \in A | X_0 = x) \ , \ \forall n = 1, 2, ...$$

On distingue deux types de chaînes de Markov selon si l'espace des valeurs de X_n (l'espace d'états) est discret ou continu. On rappelle brièvement quelques définitions et propriétés.

Cas discret. Supposons que X_n est à valeurs dans un ensemble fini $\mathcal{X} = \{x_1, ..., x_k\}$. La loi de X_n conditionnellement à X_{n-1} est alors entièrement déterminée par les valeurs

$$p_{ij} := \mathbb{P}(X_n = x_j | X_{n-1} = x_i), i, j = 1, ..., k.$$

La loi de la chaîne de Markov $X_n, n = 0, 1, ...$ dépend donc uniquement de la loi de X_0, que l'on identifie à un vecteur ligne $\nu^{(0)} = (\nu_1^{(0)}, ..., \nu_k^{(0)})$, et la matrice de transition $P = (p_{ij})_{i,j=1,...,k}$. De plus, la loi de X_1 est donnée par

$$\nu_j^{(1)} = \mathbb{P}(X_1 = x_j) = \sum_{i=1}^{k} \mathbb{P}(X_1 = x_j | X_0 = x_i)\mathbb{P}(X_0 = x_i) = \sum_{i=1}^{k} p_{ij}\nu_i^{(0)} \ , j = 1, ..., k,$$

ce qui s'écrit sous forme matricielle $\nu^{(1)} = \nu^{(0)}P$. En réitérant le calcul, on obtient la loi de X_n: $\nu^{(n)} = \nu^{(0)}P^n$.

On peut généraliser cette définition à un espace d'état inifini dénombrable avec cette fois une "matrice" de transition P indexée par $\mathbb{N}$ ou $\mathbb{Z}$.

Définition 1.4 *Une chaîne de Markox* X_n *à espace d'état dénombrable est irréductible si pour tout couple* i, j*, il existe un chemin* $i_1, ..., i_n$ *tel que les probabilités de transition* $p_{ii_1}, p_{i_1 i_2}, ..., p_{i_n j}$ *soient strictement positives.*

Une chaîne est irréductible si le graphe de P, qui possède une arête de i à j si et seulement si $p_{ij} > 0$, est fortement connexe. Autrement dit, il est possible de passer d'un état i à état j (en plusieurs étapes si nécessaire) pour tout i, j.

Théorème 1.5 *Une chaîne de Markov* X_n *irréductible à espace d'état fini admet une unique loi invariante* ν*, qui est l'unique vecteur propre de* $P^\top$ *associé à la valeur propre* 1*.*

On remarque que si X_n converge en loi, alors sa loi limite ν est nécessairement un point fixe par la transition P: $\nu = \nu P$. Autrement dit, ν définit un vecteur propre de $P^\top$ associé à la

valeur propre 1. L'existence d'un tel vecteur propre est assurée par le fait que, P vérifie $P\mathbf{1} = \mathbf{1}$ par construction (et que P et $P^\top$ ont les mêmes valeurs propres). Lorsqu'elle existe, la loi limite μ d'une chaîne de Markov est appelée *loi stationnaire* ou encore *loi invariante* du fait qu'elle stabilise le processus en loi dans le sens où, si X_{n-1} est de loi μ alors X_n l'est également.

Le cas d'un espace d'état $\mathcal{X}$ infini dénombrable est plus difficile à traiter mais des conditions existent sur les probabilités de transition p_{ij} pour assurer l'existence et unicité de la loi invariante.

Exemple. La marche aléatoire simple sur $\mathbb{Z}^d$, $X_{n+1} = X_n + Y_{n+1}$ avec les incréments Y_i indépendants et uniformément distribués sur le simplexe $\{(1,0,...,0),....,(0,...,0,1)\} \subset \mathbb{Z}^d$ est une chaîne de Markov à espace d'état dénombrable. Elle est clairement irréductible (il est toujours possible d'atteindre un état en partant de n'importe quel autre en un nombre fini d'étape). On peut montrer qu'elle n'est récurrente que pour $d \leq 2$ et pour Y_i de loi uniforme. Elle n'admet en revanche pas de loi stationnaire.

Cas continu. Si X_n est à valeurs dans $\mathcal{X}$ un espace continu, la loi de X_n conditionnellement à X_{n-1} est définie par un noyau de transition $K(.|x)$ qui associe à tout $x \in \mathcal{X}$ une loi de probabilité sur $\mathcal{X}$. Dans le cas de variables à densité, $K(.|x)$ désigne la densité de X_n conditionnellement à $X_{n-1} = x$ (qui on rappelle ne dépend pas de n). Soit X_0 de densité f_0, la densité de X_n est alors donnée par

$$f_n(y) = \int_{\mathcal{X}} K^n(y|x) f_0(x) dx \ , \ y \in \mathcal{X}$$

où $K^n(y|x) = \int_{\mathcal{X}^{n-1}} K(y|x_1)K(x_1|x_2)...K(x_{n-1}|x)dx_1...dx_{n-1}$. Ici, une loi stationnaire pour le noyau K est un point fixe f, solution de:

$$f(y) = \int_{\mathcal{X}} K(y|x) f(x) dx \ , y \in \mathcal{X}.$$

Exemple. Le processus $X_n = aX_{n-1} + \epsilon_n, a \in \mathbb{R}, n = 1, 2, ...$ où les ϵ_n sont iid de loi $\mathcal{N}(0,1)$ et indépendants du passé est une chaîne de Markov continue. Son noyau de transition est donné par $K(y|x) = \phi(y - ax)$ où ϕ est la densité Gaussienne standard. Soit X_0 une valeur initiale, on montre par récurrence que

$$X_n = a^n X_0 + \sum_{k=0}^{n-1} a^k \epsilon_{n-k}.$$

Si $a \in]-1,1[$, la loi de X_n dépend de la loi de X_0 mais cette dépendance s'estompe quand $n \to \infty$, avec comme loi limite $X_n \xrightarrow{loi} \mathcal{N}\big(0, 1/(1-a^2)\big)$. En revanche, pour $a \notin]-1,1[$, le processus n'admet pas de loi stationnaire.

2 Méthodes de Monte-Carlo

2.1 Exploration à l'aveugle

L'algorithme le plus naïf d'exploration aléatoire est l'exploration dite "à l'aveugle" pour lequel on évalue la fonction J en des points aléatoires de même loi et tirés indépendamment sur le domaine $\mathcal{X}$.

Exploration aléatoire à l'aveugle: Soit $Y_1, Y_2, ...$ une suite de variables aléatoires iid à valeurs dans $\mathcal{X}$, on définit la suite $(X_n)_{n\in\mathbb{N}}$ par $X_1 = Y_1$ et pour tout $n \geq 2$

$$X_n = \begin{cases} Y_n & \text{si } J(Y_n) \leq J(X_{n-1}) \\ X_{n-1} & \text{sinon.} \end{cases}$$

Autrement dit, X_n est la valeur (d'indice minimal) parmi les n premiers Y_i pour lequel $J(Y_i)$ est minimal, en particulier

$$X_n \in \arg \min_{y\in\{Y_1,...,Y_n\}} J(y).$$

Remarque 2.1 *Le processus $(X_n)_{n\in\mathbb{N}}$ est une chaîne de Markov homogène d'ordre 1 du fait que la loi de X_n conditionnellement au passé $X_1, ..., X_{n-1}$ ne dépend que du passé immédiat X_{n-1} et, conditionnellement à $X_{n-1} = x$, cette loi ne varie pas (homogénéité).*

Proposition 2.2 *On suppose $m = \inf_{x\in\mathcal{X}} J(x) > -\infty$. Pour tout $\eta > 0$, on note $D_\eta = \{x \in \mathcal{X} : J(x) \leq m + \eta\}$ et on suppose que les Y_i vérifient*

$$\forall \eta > 0 \ , \ p_\eta := \mathbb{P}(Y_i \in D_\eta) > 0.$$

Alors $J(X_n)$ converge presque sûrement vers m quand $n \to \infty$.

Preuve. On remarque que

$$\mathbb{P}(X_n \notin D_\eta) = \mathbb{P}\Big(\forall i \leq n \ , \ Y_i \notin D_\eta\Big) = (1 - p_\eta)^n$$

par indépendance des Y_i. Comme $J(x) \geq m$ pour tout $x \in \mathcal{X}$, on peut réécrire

$$\mathbb{P}(X_n \notin D_\eta) = \mathbb{P}(J(X_n) > m + \eta) = \mathbb{P}(|J(X_n) - m| > \eta)$$

et on déduit la convergence de $J(X_n)$ vers m en probabilité:

$$\forall \eta > 0 \ , \ \mathbb{P}(|J(X_n) - m| > \eta) = (1 - p_\eta)^n \underset{n\to\infty}{\longrightarrow} 0.$$

Or, la suite $J(X_n)$ est décroissante (par construction) et minorée par m, elle converge donc presque sûrement, sa limite ne pouvant être autre que sa limite en probabilité m. □

La condition $\mathbb{P}(Y \in D_\eta) > 0, \forall \eta > 0$ est facilement réalisable en pratique si J est continue et $\mathcal{X}$ est un ouvert. En effet, l'ensemble D_η contient l'ensemble ouvert $J^{-1}\big(]m, m+\eta[\big)$ qui est non-vide (sauf cas trivial $J = m$) et est donc de mesure de Lebesgue strictement positive pour tout $\eta > 0$. Il suffit alors de choisir une variable aléatoire Y continue de densité strictement positive sur $\mathcal{X}$, par exemple une loi uniforme si $\mathcal{X}$ est borné, où un vecteur Gaussien restreint à $\mathcal{X}$ sinon. Réciproquement, si J n'est pas continue et que son minimum est atteint pour des points isolés de son graphe, ces points sont indétectables par exploration aléatoire sans information supplémentaire sur J (prenons par exemple la fonction $J(x) = \mathbb{1}\{x \neq 0\}$).

On s'intéresse maintenant à la convergence de la suite X_n vers un minimiseur de J. On note $\|A - x\|$ la distance d'un point x à un ensemble non-vide A, donnée par

$$\|A - x\| = \inf_{a \in A} \|a - x\| \in [0, +\infty[.$$

Proposition 2.3 *On se place sous les hypothèses de la proposition précédente, et on suppose de plus que $\mathcal{X}^* := \arg\min_{x \in \mathcal{X}} J(x)$ est non-vide et vérifie*

$$\forall \epsilon > 0 \ , \quad \inf_{x : \|\mathcal{X}^* - x\| > \epsilon} J(x) > m. \tag{1}$$

Alors, $\|\mathcal{X}^* - X_n\|$ converge presque sûrement vers 0 quand $n \to \infty$. En particulier, si le minimiseur est unique, alors X_n converge presque sûrement vers celui-ci.

Preuve. Soit $\epsilon > 0$, on pose

$$\eta_\epsilon := \inf_{x : \|\mathcal{X}^* - x\| > \epsilon} J(x) - m$$

qui est strictement positif par hypothèse. En particulier, $\|\mathcal{X}^* - x\| > \epsilon \Longrightarrow J(x) - m \geq \eta_\epsilon$ ou encore

$$\big\{\omega : \|\mathcal{X}^* - X_n(\omega)\| > \epsilon\big\} \subseteq \big\{\omega : J(X_n(\omega)) \geq m + \eta_\epsilon\big\}.$$

On déduit

$$\mathbb{P}\Big(\limsup_{n \to \infty} \big\{\|\mathcal{X}^* - X_n\| > \epsilon\big\}\Big) \leq \mathbb{P}\Big(\limsup_{n \to \infty} \big\{J(X_n) \geq m + \eta_\epsilon\big\}\Big) \underset{n \to \infty}{\longrightarrow} 0$$

par la convergence presque sûre de $J(X_n)$ vers m (proposition 2.2). $\square$

Remarque 2.4 *La propriété $\|\mathcal{X}^* - X_n\| \to 0$ n'implique pas forcément que X_n converge vers un minimiseur de J s'il n'y a pas unicité.*

La condition technique (1) de la proposition signifie simplement qu'il n'est pas possible de s'approcher de m autrement que pour x s'approchant de $\mathcal{X}^*$. La continuité seule de J n'est pas suffisante pour la vérifier. Un contre-exemple est donné par la fonction $J : x \mapsto e^{-(x^2+1/x^2)}$ (prolongée par continuité en 0) définie sur $\mathcal{X} = \mathbb{R}$.

En revanche, la condition (1) est vérifiée sous l'hypothèse supplémentaire de compacité de $\mathcal{X}$. Pour le voir, procédons par l'absurde: soit $\epsilon > 0$ et $(x_n)_{n\in\mathbb{N}}$ telle que $\|\mathcal{X}^* - x_n\| > \epsilon$ et $J(x_n) \to m$ quand $n \to \infty$. Par compacité de $\mathcal{X}$, il existe une sous-suite $(x_{n_k})_{k\in\mathbb{N}}$ convergente, on note l sa limite. Par continuité, on sait que $\|\mathcal{X}^* - l\| \geq \epsilon$ alors que $J(x_{n_k})$ converge vers $m = J(l)$, ce qui conduit à une contradiction. La condition est également vérifiée si J est coercive (on peut alors facilement se ramener au cas compact).

L'efficacité de l'algorithme de Monte-Carlo décroît avec la dimension d de l'espace. Prenons $\mathcal{X}$ la boule unité $\mathcal{B}(0,1)$ de $\mathbb{R}^d$ et $J : x \mapsto \|x\|$ qui atteint son minimum en $x^* = (0, ..., 0)^\top$. On considère l'algorithme d'exploration aléatoire à l'aveugle avec Y_i des variables aléatoires iid uniformes sur $\mathcal{X}$. Soit $\epsilon < 1$ la marge d'erreur tolérée dans la recherche du minimiseur. Pour chaque Y_i, la probabilité de tomber à une distance au plus ϵ de 0 est donnée par

$$\mathbb{P}(\|Y_i\| \leq \epsilon) = \frac{\mathrm{vol}\left(\mathcal{B}(0,\epsilon)\right)}{\mathrm{vol}\left(\mathcal{B}(0,1)\right)} = \epsilon^d.$$

Le temps d'attente moyen avant d'arriver à distance ϵ du minimum vaut $(1/\epsilon)^d$ qui croît exponentiellement vite en la dimension. De même, la probabilité d'être à distance au plus ϵ de 0 après N itérations de l'algorithme est $1 - (1-\epsilon^d)^N$ ce qui signifie que, pour α un seuil de tolérance de la probabilité d'erreur,

$$N = \left\lceil \frac{\log(\alpha)}{\log(1-\epsilon^d)} \right\rceil \underset{d\to\infty}{\sim} \frac{1}{\epsilon^d}$$

itérations sont nécessaires.

2.2 Exploration aléatoire localisée

Une variante de la méthode d'exploration naïve consiste, à l'étape n, à chercher aléatoirement localement autour de X_{n-1}, de manière à provilégier les valeurs petites connues de J. Cette approche est comparable aux méthodes déterministes de descente, mais où l'intérêt d'un incrément aléatoire est d'éviter de rester coincé autour d'un minimum local.

Exploration localisée à incréments aléatoires: Soit $X_1 \in \mathcal{X}$ tiré aléatoirement et $\Delta_1, \Delta_2, ...$ une suite de variables aléatoires (typiquement centrées en 0), on définit récursivement pour $n \geq 2$,

$$X_n = \begin{cases} X_{n-1} + \Delta_n & \text{si } X_{n-1} + \Delta_n \in \mathcal{X} \text{ et } J(X_{n-1} + \Delta_n) < J(X_{n-1}) \\ X_{n-1} & \text{sinon.} \end{cases}$$

Pour plus de flexibilité, les incréments Δ_i ne sont pas forcément iid mais peuvent être adaptés au comportement récent de X_n. Les lois typiques pour les incréments Δ_i sont la loi uniforme sur la boule $\mathcal{B}(0, R)$ avec R fixé ou aléatoire, ou encore des vecteurs Gaussiens centrés, de matrice de covariance arbitraire pouvant être adaptée au domaine.

Remarque 2.5 *Ici aussi, le processus* $(X_n)_{n\in\mathbb{N}}$ *est une chaîne de Markov d'ordre* 1. *Elle est homogène si les incréments sont iid.*

L'avantage d'une recherche localisée est d'obtenir une convergence potentiellement plus rapide, au risque de converger vers un minimum local. On rappelle qu'un point x_1 est un minimiseur local de J s'il existe un voisinage V de x_1 dans $\mathcal{X}$ tel que $\forall x \in V$, $J(x) \geq J(x_1)$. La valeur $J(x_1)$ est alors un minimum local. L'existence de minima locaux qui ne sont pas des minima globaux est un des principaux obstacles en optimisation. Ici, si les incréments sont à support borné (par exemple une boule centrée de rayon $r > 0$), et que l'algorithme se rapproche d'un point x_1 qui minimise J dans un rayon supérieur à r, il peut arriver qu'il ne soit plus possible de sortir de ce voisinage rendant la convergence vers un minimum global impossible. Considérer des incréments à support non borné comme des variables normales résout le problème "en théorie", même s'il faut se méfier de la décroissante très rapide de la densité Gaussienne qui se comporte virtuellement comme une variable à support compact). Enfin, choisir des incréments à très forte variance retrouve des performances comparables à la recherche à l'aveugle.

La solution la plus efficace pour repérer/éviter la convergence vers un minimum local est sans doute de répéter la procédure en partant de points initiaux différents (par exemple tirés aléatoirement uniformément sur $\mathcal{X}$ si l'ensemble est borné).

La recherche locale est à privilégier quand J est régulière (par exemple convexe), auquel cas la convergence vers le minimum global est mieux ciblée et donc généralement plus rapide. Les propriétés théoriques de convergence de l'algorithme d'exploration aléatoire locale sont semblables à celles de la recherche à l'aveugle.

2.3 Le cas discret

On s'intéresse maintenant au cas où le domaine de définition $\mathcal{X}$ est fini ou dénombrable. La non-continuité du domaine $\mathcal{X}$ les problèmes d'optimisation discrets sont généralement liés à des problèmes de combinatoire potentiellement très difficiles.

Sur un domaine discret $\mathcal{X}$, les notions classiques de régularité (continuité, convexité de la fonction J) ont un sens différent. Un problème d'optimisation sur un ensemble discret n'a d'intérêt que si on peut définir une notion de proximité entre deux points de l'ensemble (typiquement $\mathcal{X}$ est un espace métrique) et si J vérifie une certaine régularité sur $\mathcal{X}$, en prenant des

valeurs proches pour des points $x, x' \in \mathcal{X}$ à distance faible. Si ce n'est pas le cas, une méthode d'exploration à l'aveugle est la plus adaptée.

S'il existe une notion naturelle de distance minimale entre deux points du domaine $\mathcal{X}$, la représentation de $\mathcal{X}$ par un graphe peut constituer un outil conceptuel utile. Une exploration aléatoire du domaine peut alors se faire de façon localisée en passant par des points adjacents. La représentation se fait de la façon suivante.

- Le sommets du graphe sont les éléments de $\mathcal{X}$.
- Il y a une arête (non dirigée) entre deux sommets différents x, x' si et seulement si x et x' diffèrent par une opération élémentaire (càd s'ils sont à distance minimale).

La distance minimale qui correspond aux arêtes du graphe doit être adaptée au problème de minimisation, et plus précisément à la fonction J à minimiser. Concrètement, on cherche à ce que le graphe rende J la plus régulière (continue) possible.

Exemples

1. Soit le problème classique de logistique d'attributions de tâches $t_1, ..., t_k$ à des machines $m_1, ..., m_k$. On note δ_{ij} la durée (le coût) de la tâche t_i effectuée par la machine m_j. En supposant qu'aucune machine ne puisse effectuer plus d'une tâche, une attribution des tâches correspond à une permutation x de $\{1, ..., k\}$, c'est-à-dire une bijection de $\{1, ..., k\}$ dans lui-même. On peut chercher à minimiser le temps d'exécution $J(x) = \max_{i=1,...,k} \delta_{ix(i)}$ ou encore le coût total $\sum_{i=1}^{k} \delta_{ix(i)}$. Dans ce cas, le nombre minimal de transpositions nécessaires pour passer d'une permutation à une autre définit une distance sur l'ensemble des permutations, par rapport à laquelle J est (normalement) assez régulière.

2. Le problème du voyageur de commerce consiste à chercher l'itinéraire le plus court permettant de visiter un ensemble de n villes et revenir au point de départ. Un itinéraire correspond à une permutation sur $\{1, ..., n\}$ où $x(i)$ est la i-ème ville visitée. La ville de départ peut être fixée sans perte de généralité, il y a donc $(n-1)!$ itinéraires possibles. La fonction objectif à minimiser s'écrit comme la somme des distances

$$J(x) = \sum_{i=1}^{n-1} d(x(i), x(i+1)) + d(x(n), x(1)).$$

   On peut considérer que deux itinéraires sont à distance minimale (adjacents) si les ordres de passages sont identiques à l'exception de deux villes.

3. En apprentissage, la classification binaire non-supervisée consiste à diviser un ensemble fini A en deux groupes homogènes. Pour $x \subseteq A$, on note $\overline{a}_x = \frac{1}{\#(x)} \sum_{a \in x} a$ la valeur moyenne

sur x. On cherchera à minimiser un critère d'homogénéité, par exemple

$$J(x) = \sum_{a \in x} (a - \overline{a}_x)^2 + \sum_{a \in A \setminus x} (a - \overline{a}_{A \setminus x})^2 \ , \ x \subseteq A.$$

Deux ensembles x, x' qui diffèrent par peu de points correspondront à des valeurs de J proches. La classification en plus de deux groupes nécessite de regarder non plus les sous-ensembles de A mais les partitions, ce qui augmente considérablement le nombre de possibilités.

3 Méthodes adaptatives

3.1 Recuit simulé

Pour qu'un algorithme d'exploration aléatoire soit efficace, on s'attend à ce que la loi de X_n tende à charger des valeurs pour lesquelles J est minimale quand $n \to \infty$. Dans le cas extrême, si la loi de X_n a pour support $\mathcal{X}^* = \arg\min_{x \in \mathcal{X}} J(x)$ à partir d'un certain rang, alors l'algorithme a convergé et fournit une solution exacte au problème d'optimisation. Bien sûr, cette condition est difficilement atteignable, surtout sur des domaines continus. L'algorithme du recuit-simulé permet de générer un processus qui converge en loi vers une approximation de la loi uniforme sur $\mathcal{X}^*$.

Définition 3.1 *Soit une température $T > 0$, on suppose que la fonction $\exp(-J(x)/T)$ est intégrable sur $\mathcal{X}$. La mesure de Gibbs associée à J est la loi de densité sur $\mathcal{X}$ donnée par*

$$\pi_T(x) = \frac{\exp(-J(x)/T)}{\int \exp(-J(s)/T) ds} \ , \ x \in \mathcal{X}.$$

On peut bien sûr définir cette loi sur un ensemble discret en remplaçant l'intégrale $\int \exp(-J(s)/T) ds$ par une somme sur tous les éléments de l'ensemble.

Quand T tend vers 0, la mesure de Gibbs se rapproche d'une certaine façon de la loi uniforme sur $\mathcal{X}^* = \arg\min_{x \in \mathcal{X}} J(x)$. En effet, π_T est constante sur $\mathcal{X}^*$ et

$$\forall x^* \in \mathcal{X}^*, \forall x \in \mathcal{X} \setminus \mathcal{X}^*, \ \lim_{T \to 0} \frac{\pi_T(x^*)}{\pi_T(x)} = \lim_{T \to 0} \left(e^{J(x) - J(x^*)}\right)^{1/T} = +\infty.$$

Pour générer des variables de densité π_T approximativement, on utilise l'algorithme de Métropolis-Hastings qui est basé sur une méthode d'acceptation/rejet sur des chaînes de Markov. Cet algorithme ne nécessite pas de connaître la constante de normalisation $\int \exp(-J(s)/T) ds$, ce qui est primordial ici.

Lemme 3.2 *Deux densités f, g définies sur un même ensemble $\mathcal{X}$ sont proportionnelles (càd vérifient $\forall x \in \mathcal{X}, f(x) = \lambda g(x)$ pour un $\lambda \in \mathbb{R}$) si et seulement si elles sont égales.*

Preuve. Il suffit d'intégrer: $1 = \int f(x)dx = \int \lambda g(x)dx = \lambda \int g(x)dx = \lambda$. □

La densité de la mesure de Gibbs est donc l'unique densité proportionnelle à la fonction $\exp(-J(x)/T)$, ce que l'on note

$$\pi_T(x) \propto \exp(-J(x)/T)\ ,\ x \in \mathcal{X}.$$

Soit f la densité que l'on veut simuler. On considère une densité g symétrique facile à simuler, qui va servir pour la loi des incréments (typiquement g est la densité uniforme sur une boule centrée en 0 ou une densité Gaussienne centrée). L'idée de l'algorithme de Metropling-Hastings (dans une version simplifiée) est de générer, à l'étape n, une variable Δ_n de densité g et d'actualiser la chaîne par $X_n = X_{n-1} + \Delta_n$ conditionnellement à un événement de probabilité

$$p(X_{n-1}, X_n) := \min\big\{1, f(X_n)/f(X_{n-1})\big\}$$

issu de variables indépendantes. Autrement dit, la chaîne est actualisée si $f(X_n) > f(X_{n-1})$ (c'est-à-dire X_n est plus "vraisemblable" que X_{n-1} pour f) mais n'est actualisée qu'avec probabilité $f(X_n)/f(X_{n-1})$ sinon. Pour générer un événement de probabilité $p(X_{n-1}, X_n)$, il suffit de générer une variable aléatoire U_n de loi uniforme sur $[0, 1]$ indépendamment, on actualise la chaîne si $U_n \leq p(X_{n-1}, X_n)$.

Proposition 3.3 *Conditionnellement à l'événement $U_n \leq p(X_{n-1}, X_n)$, X_n décrit une chaîne de Markov de noyau de transition*

$$x \mapsto K(x|x_{n-1}) \propto g(x - x_{n-1}) \min\Big\{1, \frac{f(x)}{f(x_{n-1})}\Big\},$$

et de loi stationnaire de densité f.

Preuve. Remarquons que X_n est bien une chaîne de Markov puisque sa loi condionnellement au passé ne dépend que de X_{n-1} et est invariante dans le temps. Par construction, la loi à l'étape n du couple (X_n, U_n) sachant X_{n-1} a pour densité

$$(x, u) \mapsto \frac{g(x - X_{n-1})\mathbb{1}\{u \leq p(X_{n-1}, x)\}}{\mathbb{P}\big(U_n \leq p(X_{n-1}, X_n)\big)} \propto g(x - X_{n-1})\mathbb{1}\big\{u \leq f(x)/f(X_{n-1})\big\}$$

pour $x \in \mathcal{X}, u \in [0, 1]$. On obtient la densité $K(.|X_{n-1})$ de X_n sachant X_{n-1} en intégrant en u:

$$\begin{aligned} K(x|X_{n-1}) &\propto \int_0^1 g(x - X_{n-1})\mathbb{1}\big\{u \leq f(x)/f(X_{n-1})\big\}du \\ &\propto g(x - X_{n-1}) \int_0^{\min\{1, f(x)/f(X_{n-1})\}} du \\ &\propto g(x - X_{n-1}) \min\big\{1, f(x)/f(X_{n-1})\big\}. \end{aligned}$$

Montrons que f est stationnaire par K:

$$\int K(x|y)f(y)dy \propto f(x)\int g(x-y)\min\Big\{1,\frac{f(y)}{f(x)}\Big\}dy \propto f(x)\int K(y|x)dy \propto f(x),$$

où on a utilisé que g est symétrique et $K(.|x)$ est une densité pour tout $x \in \mathcal{X}$. $\square$

On voit bien que la construction de la chaîne X_n nécessite d'évaluer f à une constante multiplicative près puisque la règle d'acceptation ne dépend de f qu'à travers $f(X_n)/f(X_{n-1})$.

L'algorithme de recuit-simulé génère par la méthode de Metropolis-Hastings une chaîne de Markov de loi stationnaire π_{T_n} en suivant un schéma de températures $(T_n)_{n\in\mathbb{N}}$ décroissantes. Il tient son nom d'un procédé en métallurgie qui consiste à alterner des cycles de refroidissement d'un métal pour obtenir une bonne composition.

Recuit simulé: Soit $X_0 \in \mathcal{X}$ une valeur initiale, g une densité symétrique et $(T_n)_{n\in\mathbb{N}}$ une suite décroissante positive qui tend vers zéro. A chaque étape:

1. On génère Δ de densité g.
2. Si $J(X_{n-1}+\Delta) < J(X_{n-1})$ on actualise la chaîne par $X_n = X_{n-1}+\Delta$.
3. Sinon, on génère U de loi uniforme sur $[0,1]$: si $U < \exp\big((J(X_{n-1}) - J(X_{n-1}+\Delta))/T_n\big)$, on actualise la chaîne, sinon on reprend à l'étape 1.

On peut choisir une suite $(T_n)_{n\in\mathbb{N}}$ qui décroît par paliers. Le choix de la suite $(T_n)_{n\in\mathbb{N}}$ a une très grande influence sur l'efficacité de l'algorithme mais il n'existe pas de méthode universelle pour la choisir. L'idée est de trouver un compromis entre une décroissante suffisamment rapide pour accélérer la convergence vers le minimum global de J tout en gardant des paliers T_n constants suffisamment longs pour permettre la convergence de l'algorithme de Metropolis-Hastings.

Remarque 3.4 *Contrairement aux méthodes dites de Monte-Carlo décrites dans le chapitre précédent, l'algorithme du recuit-simulé génère une chaîne de Markov inhomogène, du fait de l'évolution de la température.*

3.2 Algorithmes génétiques

Les algorithmes génétiques font partie des méthodes dites d'évolution qui s'inspirent du phénomène de sélection naturelle. L'idée est de faire évoluer une population de M individus, correspondant à M trajectoires simultanées, qui vont interagir entre elles "intelligemment" pour permettre à certains individus de converger plus rapidement vers une solution du problème. Dans cette partie, on suppose que la fonction objectif J est positive sur $\mathcal{X}$.

Un algorithme génétique nécessite de définir des processus de *sélection*, *croisement* et *mutation* adaptés au problème et qui seront implémentés à chaque itération.

1. **Sélection:** On tire dans la population actuelle $X_n^1, ..., X_n^M$ un groupe de $m < M$ individus en privilégiant généralement les petites valeurs de la fonction objectif J. Il y a plusieurs méthodes possibles, par exemple:

   - On tire m individus ayant le meilleur score d'adaptation (c'est-à-dire les valeurs de $J(x)$ les plus faibles).
   - On tire aléatoirement m individus selon des probabilités p_i ordonnées dans l'ordre inverse des $J(X_n^i)$. Le choix des probabilités est arbitraire, on peut aussi utiliser par exemple la mesure de Gibbs sur l'échantillon avec une température bien choisie ou encore des probabilités proportionnelles à $1/J(X_n^i)$ si J est positive.
   - On tire aléatoirement m individus uniformément dans la population. Cette méthode de sélection est neutre car elle ne tient pas compte de J. Dans ce cas, l'efficacité de l'agorithme dépendra uniquement des étapes de croisement et mutation.

2. **Croisement:** On crée aléatoirement M paires dans les individus sélectionnés à l'étape précédente (un individu peut appartenir à plusieurs paires). On définit pour chaque paire une dépendance qui mélange les attributs des parents. L'exécution en pratique dépend largement du problème et peut être déterminée indépendamment de la fonction objectif. Typiquement, on peut prendre une combinaison convexe des deux individus si on est dans un espace continu.

3. **Mutation:** On modifie (légèrement) chaque individu obtenu après croisement. Par exemple une (petite) perturbation aléatoire dans le cas continu, ce qui s'apparente à une exploration aléatoire localisée. Dans le cas discret, cela revient à choisir un individu "proche" (possiblement lui-même) en un sens adapté au problème. La mutation correspond souvent à un changement très léger qui a pour but d'éviter d'obtenir des individus trop similaires au sain d'une population, qui peut entraîner des convergences vers des minima locaux.

Remarque 3.5 *Il est très facile de créer des variantes pour chaque processus qui vont fonctionner plus ou moins bien selon les cas, la version décrite ici n'est qu'une des nombreuses possibilités.*

Exemples

1. Soit une fonction J définie sur un espace continu $\mathcal{X} \subset \mathbb{R}^d$. Pour des parents x, x', on définit le croisement par la combinaison convexe $(1-\lambda)x + \lambda x'$ avec $\lambda \in]0,1[\setminus\{0\}$ (λ peut être calibré de manière à privilégier les petites valeurs de $J(x)$). Pour la partie mutation on ajoute une perturbation aléatoire de loi normale centrée. Cela revient à générer simultanément M trajectoires d'exporation aléatoire localisée qui sont moyénnées deux à deux aléatoirement à chaque étape.

2. On cherche à retrouver une chaîne de caractères x^* (par exemple un mot de passe) inconnue. Pour toute chaîne de caractères x de même longueur, on suppose qu'on peut évaluer le nombre $J(x)$ de caractères incorrects. On part de plusieurs chaînes tirées au hasard et on définit à chaque étape le croisement de deux chaînes x, x' en prenant aléatoirement les caractères de x ou x' avec probabilité 1/2 (ou une probabilité qui dépend intelligemment de $J(x)$ et $J(x')$). Pour l'étape de mutation, on peut changer un caractère aléatoirement (ou même ne rien faire).

La plupart des méthodes d'exploration aléatoire peuvent être implémentées sous la forme d'algorithmes génétiques. Cela revient à générer plusieurs trajectoires en parallèle en conservant (sélection) et combinant (croisement) les meilleures réalisations à chaque étape, pour accélérer la convergence.

4 Gradient stochastique

Dans ce chapitre, on s'intéresse à des problèmes d'optimisation pour lesquels la fonction objectif J s'écrit comme une somme de fonctions différentiables $J_i : \mathbb{R}^d \to \mathbb{R}$,

$$J(x) = \frac{1}{N} \sum_{i=1}^{N} J_i(x),$$

où le nombre d'éléments N est typiquement très grand. Ce contexte se retrouve particulièrement dans des méthodes d'apprentissage en grande dimension où N représente le nombre d'observations.

Remarque 4.1 *On peut généraliser ce cadre à une fonction objectif qui s'écrit comme l'espérance d'une fonction aléatoire $J(x) = \mathbb{E}(J_I(x))$, indexée par une variable aléatoire I (x est déterministe). Le cas précédent est alors un cas particulier où I est de loi uniforme sur $\{1, ..., N\}$.*

La difficulté de ce genre de problème vient de la complexité d'évaluation de la fonction J du fait du grand nombre de termes N, qui induit un gros coût de calcul même si les J_i sont régulières et simples à évaluer. Alors qu'un algorithme déterministe de descente de gradient nécessiterait d'évaluer le gradient de J_i en tout point, l'algorithme de gradient stochastique restreint l'évaluation du gradient à un point tiré aléatoirement dans l'échantillon.

Gradient stochastique: Soit X_0 un point de départ quelconque et une suite γ_n qui décroît vers zéro. A chaque étape:

1. On tire un indice aléatoire i_n uniformément dans $\{1, ..., N\}$.
2. On définit $X_n = X_{n-1} - \gamma_n \nabla J_{i_n}(X_{n-1})$.

4.1 Convergence

Un grand nombre d'itérations (souvent de l'ordre de N) sont généralement nécessaires pour atteindre une valeur satisfaisante. La convergence est vérifiée (théoriquement) sous des conditions fortes de régularité des fonctions J_i et un contrôle de la suite $(\gamma_n)_{n\in\mathbb{N}}$.

Remarque 4.2 *Un algorithme simple de descente de gradient déterministe construit récursivement*

$$X_n = X_{n-1} - \gamma \nabla J(X_{n-1}),$$

avec $\nabla J = \frac{1}{N}\sum_{i=1}^N \nabla J_i$. *Ici, un pas* $\gamma > 0$ *constant peut suffire pour obtenir la convergence du fait que* $\nabla J(X_n) \to 0$ *si* X_n *converge vers un minimiseur* x^*. *Pour la version stochastique, la condition n'est plus vraie si* J *est remplacée par* ∇J_i *(minimiser* $J = \sum_{i=1}^N J_i$ *n'implique pas de minimiser simultanément tous les* J_i*) ce qui est compensé en choisissant un pas* γ_n *qui tend vers zéro.*

Théorème 4.3 *On fait les hypothèses suivantes:*

i) Les J_i *sont différentiables et de gradients uniformément bornés:*

$$\exists M < \infty \ , \ \forall i = 1, ..., N \ , \ \sup_{x\in\mathcal{X}} \|\nabla J_i(x)\| < M.$$

ii) $J = \frac{1}{N}\sum_{i=1}^N J_i$ *admet un unique minimiseur* x^* *qui est dans l'intérieur du domaine* $\mathcal{X}$.

iii) Il existe $\eta > 0$ *tel que* $\langle x - x^*, \nabla J(x)\rangle \geq \eta\|x - x^*\|^2$ *pour tout* $x \in \mathcal{X}$.

iv) $\sum_{n=1}^{+\infty} \gamma_n = +\infty$ *et* $\sum_{n=1}^{+\infty} \gamma_n^2 < +\infty$.

Alors $\lim_{n\to\infty} \mathbb{E}\|X_n - x^*\|^2 = 0$.

Preuve. On a

$$\begin{aligned}
\|X_n - x^*\|^2 &= \|X_{n-1} - x^* - \gamma_n \nabla J_{i_n}(X_{n-1})\|^2 \\
&= \|X_{n-1} - x^*\|^2 + \gamma_n^2 \|\nabla J_{i_n}(X_{n-1})\|^2 - 2\gamma_n \big\langle X_{n-1} - x^*, \nabla J_{i_n}(X_{n-1})\big\rangle
\end{aligned}$$

D'après l'hypothèse i),

$$\mathbb{E}\big(\|X_n - x^*\|^2 \big| X_{n-1}\big) \leq \|X_{n-1} - x^*\|^2 + \gamma_n^2 M^2 - 2\gamma_n \big\langle X_{n-1} - x^*, \mathbb{E}\big(\nabla J_{i_n}(X_{n-1})\big| X_{n-1}\big)\big\rangle.$$

Sachant X_{n-1}, l'aléa ne vient que de i_n, d'où $\mathbb{E}\big(\nabla J_{i_n}(X_{n-1})\big|X_{n-1}\big) = \nabla J(X_{n-1})$. Par iii),

$$\begin{aligned}
\mathbb{E}\big(\|X_n - x^*\|^2 \big| X_{n-1}\big) &\leq \|X_{n-1} - x^*\|^2 + \gamma_n^2 M^2 - 2\gamma_n \eta \|X_{n-1} - x^*\|^2 \\
&\leq (1 - 2\eta\gamma_n)\|X_{n-1} - x^*\|^2 + \gamma_n^2 M^2
\end{aligned}$$

ce qui donne en espérance

$$\mathbb{E}\|X_n - x^*\|^2 \leq (1 - 2\eta\gamma_n)\mathbb{E}\|X_{n-1} - x^*\|^2 + \gamma_n^2 M^2. \tag{2}$$

On pose $a_n = \mathbb{E}\|X_n - x^*\|^2$ et $b_{nk} = \prod_{j=0}^{k}(1 - 2\eta\gamma_{n-j})$. On suppose sans perte de généralité que $1 - 2\eta\gamma_n > 0$ (ce qui est vrai à partir d'un certain rang), en itérant l'inégalité précédente, on trouve

$$a_n \leq b_{nn}.a_0 + M^2\left[\sum_{k=1}^{n-1} b_{nk-1}\gamma_{n-k}^2 + \gamma_n^2\right]$$

Pour tout $1 < K_n < n$,

$$\sum_{k=1}^{n-1} b_{nk-1}\gamma_{n-k}^2 = \sum_{k=1}^{K_n} b_{nk-1}\gamma_{n-k}^2 + \sum_{k=K_n+1}^{n-1} b_{nk-1}\gamma_{n-k}^2 \leq \sum_{k=1}^{K_n} \gamma_{n-k}^2 + b_{nK_n} \sum_{k=K_n+1}^{n-1} \gamma_{n-k}^2.$$

On choisit K_n tel que $(n - K_n) \to \infty$ et $\sum_{j=1}^{K_n} \gamma_{n-j} \to \infty$ quand n tend vers l'infini. Comme $1-x \leq e^{-x}$ pour $x \geq 0$, on remarque $b_{nK_n} \leq \exp\left(-2\eta \sum_{j=1}^{K_n} \gamma_{n-j}\right) \to 0$. En posant $\Gamma = \sum_{n=1}^{\infty} \gamma_n^2$, on déduit

$$\sum_{k=1}^{n-1} b_{nk-1}\gamma_{n-k}^2 \leq \Gamma - \sum_{k=1}^{K_n} \gamma_k^2 + b_{nK_n}\Gamma \underset{n\to\infty}{\longrightarrow} 0.$$

ce qui conclut la preuve. □

Il est difficile de calibrer le taux γ_n à partir de la majoration utilisée dans la preuve. En pratique, un choix classique qui vérifie les hypothèses du théorème est $\gamma_n \sim 1/n$ qui permet par un bon choix de paramètres d'obtenir une vitesse de convergence de l'ordre de $1/n$.

Corollaire 4.4 *Sous les hypothèses du théorème, si de plus J est convexe et $\nabla J(.)$ est Lipschitzienne:*

$$\forall x, x' \in \mathcal{X}\ ,\ \|\nabla J(x) - \nabla J(x')\| \leq L\|x - x'\|,$$

alors $\mathbb{E}\big(J(X_n) - J(x^*)\big) \leq L\ \mathbb{E}\|X_n - x^*\|^2$.

Preuve. Par la convexité de J, on sait que $J(x^*) \geq J(x) + \langle x^* - x, \nabla J(x)\rangle$ pour tout $x \in \mathcal{X}$. En particulier, pour $x = X_n$,

$$J(X_n) \leq J(x^*) + \langle X_n - x^*, \nabla J(X_n)\rangle \leq J(x^*) + \|X_n - x^*\|.\|\nabla J(X_n)\|,$$

par l'inégalité de Cauchy-Schwarz. Or $\|\nabla J(X_n)\| = \|\nabla J(X_n) - \nabla J(x^*)\| \leq L\|X_n - x^*\|$. On conclut en prenant l'espérance. □

Remarque 4.5 *Du fait que $\|X_n - x^*\|$ est au carré dans le membre de droite de l'inégalité, la convergence de $J(X_n)$ vers le minimum $J(x^*)$ est plus rapide que celle de $X_n - x^*$.*

La moyenne $\overline{X}_n := \frac{1}{n}\sum_{i=1}^n X_i$ converge aussi vers le minimiseur x^* et donne dans certains cas de meilleurs résultats. Le pas d'apprentissage γ_n doit alors être calibré différemment.

Proposition 4.6 *On se place sous les hypothèses du théorème. Pour tout $c > 0$, en prenant $\gamma_i = c/\sqrt{n}$ (qui ne dépend pas de i mais seulement du nombre d'itérations fixé à l'avance), on a*

$$\mathbb{E}\|\overline{X}_n - x^*\|^2 \leq \frac{1}{2\eta\sqrt{n}}\big(c^{-1}\mathbb{E}\|X_1 - x^*\|^2 + cM^2\big)$$

Preuve. On rappelle la version discrète de l'inégalité de Jensen. Soit $\lambda_1, ..., \lambda_n$ des poids positifs de somme 1 et ϕ une fonction convexe, alors pour tout $x_1, ..., x_n \in \mathcal{X}$

$$\phi\Big(\sum_{i=1}^n \lambda_i x_i\Big) \leq \sum_{i=1}^n \lambda_i \phi(x_i).$$

On applique cette inégalité à la suite X_i avec $\lambda_i = 1/n$ et $\phi = \| \,.\, - x^*\|^2$. Après passage à l'espérance, on obtient

$$\mathbb{E}\|\overline{X}_n - x^*\|^2 \leq \frac{1}{n}\sum_{i=1}^n \mathbb{E}\|X_i - x^*\|^2.$$

D'après l'équation (2), pour $\gamma_i = c/\sqrt{n}$,

$$\frac{2\eta c}{\sqrt{n}}\mathbb{E}\|X_i - x^*\|^2 \leq \mathbb{E}\|X_i - x^*\|^2 - \mathbb{E}\|X_{i+1} - x^*\|^2 + \frac{c^2M^2}{n},$$

qui donne en sommant sur i,

$$\frac{2\eta c}{\sqrt{n}}\sum_{i=1}^n \mathbb{E}\|X_i - x^*\|^2 \leq \mathbb{E}\|X_1 - x^*\|^2 - \mathbb{E}\|X_{n+1} - x^*\|^2 + c^2M^2 \leq \mathbb{E}\|X_1 - x^*\|^2 + c^2M^2,$$

ou encore

$$\frac{1}{n}\sum_{i=1}^n \mathbb{E}\|X_i - x^*\|^2 \leq \frac{\mathbb{E}\|X_1 - x^*\|^2}{2\eta c\sqrt{n}} + \frac{cM^2}{2\eta\sqrt{n}},$$

ce qui conclut la preuve. □

Le choix optimal du taux d'apprentissage diffère donc selon si on considère X_n comme solution ou la moyenne $\overline{X}_n$. La valeur de c qui minimise le membre de droite est $c = \sqrt{\mathbb{E}\|X_1 - x^*\|^2}/M$ qui est inconnue en pratique.

4.2 Application à la régression

L'algorithme du gradient stochastique est particulièrement utilisé en apprentissage statistique où on cherche à expliquer des relations entre variables à partir d'un échantillon. Lorsque l'échantillon est de très grande taille, une optimisation déterministe exacte est trop coûteuse. On se contente alors d'approximation par des méthodes de type Monte-Carlo.

La régression linéaire suppose une relation linéaire de type $T = x(S)$ entre une variable T et un vecteur de variables explicatives S. Dans le cadre statistique classique, on observe un échantillon $(s_i, t_i), i = 1, ..., N$ traité comme des réalisations indépendantes d'un même vecteur aléatoire. En posant $S = (s_1, ..., s_N)^\top$ et $T = (t_1, ..., t_n)^\top$, la relation s'écrit

$$T = x(S) + \epsilon,$$

où ϵ modélise l'erreur d'ajustement qui "regroupe" toute l'information non disponible. En dimension finie, l'opérateur x peut être identifié à un vecteur de $\mathbb{R}^d$, $d \geq 1$. Le modèle de régression simple correspond au cas d'une relation de type linéaire $t_i = xs_i + \epsilon_i$ entre des variables réelles t_i et s_i, avec ici $x \in \mathbb{R}$. Pour la perte quadratique, le problème d'optimisation consiste à minimiser

$$J(x) = \frac{1}{N}\|T - x.S\|^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - xs_i)^2 \ , \ x \in \mathbb{R}.$$

Pour une relation affine, il suffit d'ajouter la constante comme nouvelle variable explicative. Le modèle de régression multiple est une généralisation à plusieurs variables explicatives $s_i = (s_i^{(1)}, ..., s_i^{(d)})$. Dans ce cas l'opérateur x est une forme linéaire et s'identifie à un vecteur de $\mathbb{R}^d$

$$x(S) = \begin{pmatrix} \langle x, s_1 \rangle \\ \vdots \\ \langle x, s_d \rangle \end{pmatrix}.$$

On a donc ici $J(x) = \|T - x(S)\|^2 = \sum_{i=1}^{N} J_i(x)$ pour $J_i(x) := (t_i - \langle x, s_i \rangle)^2$. Enfin, il arrive d'utiliser une fonction de coût autre que la perte quadratique. Soit $c : \mathbb{R}^2 \to \mathbb{R}_+$ une fonction différentiable où $c(a, b)$ représente le coût d'une estimation de a par b, on définit

$$J(x) = \frac{1}{N}\sum_{i=1}^{N} c\big(t_i, \langle x, s_i \rangle\big) := \frac{1}{N}\sum_{i=1}^{N} J_i(x) \ , \ x \in \mathbb{R}^d.$$

Exemples

1. Dans le cas plus simple du problème de régression linéaire, on cherche à évaluer la meilleure approximation de $T = (t_1, ..., t_N)^\top$ par un vecteur constant $x.\mathbf{1}$ pour $x \in \mathbb{R}$ et $\mathbf{1} =$

$(1,...,1)^\top$. Si N est très grand, calculer la moyenne empirique $\bar{t} = \sum_{i=1}^N t_i$ peut être coûteux (?) mais on peut l'approcher par un algorithme de descente de gradient. On écrit alors $\bar{t}$ comme le minimiseur de

$$J : x \mapsto \frac{1}{N}\sum_{i=1}^N \|x - t_i\|^2 := \frac{1}{N}\sum_{i=1}^N J_i(x) \ , \ x \in \mathbb{R}^d.$$

On a ici $\nabla J_i(x) = 2(x - t_i)$. En prenant $\gamma_n = 1/2n$, on déduit les premières itérations (et pour X_0 quelconque)

$$X_1 = t_{i_1} \ , \ X_2 = t_{i_1} - \frac{t_{i_1} - t_{i_2}}{2} = \frac{t_{i_1} + t_{i_2}}{2} \ , \ X_3 = \frac{t_{i_1} + t_{i_2}}{2} - \frac{1}{3}\Big(\frac{t_{i_1} + t_{i_2}}{2} - t_{i_3}\Big) = \frac{t_{i_1} + t_{i_2} + t_{i_3}}{3}$$

et finalement le terme général $X_n = (t_{i_1} + ... + t_{i_n})/n$. L'algorithme de gradient stochastique (avec le pas $\gamma_n = 1/2n$) revient donc ici à évaluer la moyenne empirique sur un sous-échantillon tiré aléatoirement avec remise. On remarque que le choix de la constante (ici 2) qui apparaît dans le taux d'apprentissage γ_n a une forte influence sur le comportement de la solution. La valeur optimale 2 correspond à la plus petite valeur de η vérifiant l'hypothèse *iii*) du théorème.

2. Dans le cas général avec $s_i \in \mathbb{R}^d$, le calcul est très vite plus compliqué. Pour la perte quadratique, on pose $J_i(x) = \frac{1}{2}(t_i - \langle x, s_i\rangle)^2$, ce qui donne

$$\nabla J_i(x) = -(t_i - \langle x, s_i\rangle)s_i = -t_i s_i + ({s_i}^\top s_i)x.$$

On obtient alors la suite définie récursivement par

$$X_n = \gamma_n t_{i_n} s_{i_n} + \big(\mathrm{I} - \gamma_n {s_{i_n}}^\top s_{i_n}\big) X_{n-1} \ , \ n = 1, 2, ...$$

3. Pour une fonction de perte quelconque $c(.,.)$ différentiable en son deuxième argument (de dérivée notée $\partial_2 c$), on a $J_i(x) = c(t_i, \langle s_i, x\rangle)$ et la suite s'écrit

$$X_n = X_{n-1} - \gamma_n s_{i_n} \partial_2 c\big(t_{i_n}, \langle s_{i_n}, X_{n-1}\rangle\big) \ , \ n = 1, 2, ...$$

4. De manière générale, une relation du type

$$T = f(S) + \epsilon$$

entre des variables existe toujours mais n'est pas forcément linéaire. Les versions simples des réseux de neurones artificiels permettent d'estimer la fonction de régression f à partir de transformations de fonctions linéaires. Typiquement, on cherchera à approcher f par une fonction du type $g \circ x(S)$ où g est une fonction simple fixée par le statisticien (appelée

fonction d'activation) et x est un opérateur linaire. C'est le principe du réseau de neurone monocouche. La version à deux couches s'écrit

$$f(S) \approx g_0 \circ x_0\big(g_1 \circ x_1(S), ..., g_k \circ x_k(S)\big)\big),$$

qui permet (beaucoup) plus de flexibilité. Ici encore, $x_0, x_1, ..., x_k$ sont des opérateurs linéaires et $g_0, g_1, ..., g_1$ des fonctions simples fixées. Ces méthodes sont extrêmement puissantes mais difficiles à calibrer du fait du grand nombre de paramètres et de la nature non-linéaire du modèle. En l'absence d'expression explicite, on utilise des méthodes numériques d'optimisation dont le gradient stochastique, particulièrement adapté au problème.

5 Algorithme EM

L'algorithme EM (pour Espérance-Maximisation) est une méthode d'optimisation introduite pour le calcul du maximum de vraisemblance, rendu difficile par la présence de variables latentes (variables non observées) ou de données manquantes. Contrairement aux autres méthodes d'optimisation vues dans ce cours, cet algorithme n'est pas basé sur une exploration aléatoire de l'espace et n'est limité qu'au cadre restreint du calcul de maximum de vraisemblance. Son appartenance à la catégorie des algorithmes stochastiques est surtout due au fait qu'il est utilisé en Statistique et qu'il fait intervenir des variables aléatoires et calculs d'espérance.

Soit $X = (X_1, ..., X_n)$ des observations issues de variables aléatoires iid dont la loi appartient à un modèle paramétrique $\mathcal{M} = \{\mu_\theta : \theta \in \Theta\}$. On supposera pour simplifier que pour tout $\theta \in \Theta$, μ_θ admet une densité f_θ sur $\mathbb{R}^d$ par rapport à la mesure de Lebesgue.

Définition 5.1 *La vraisemblance du modèle $\mathcal{M}$ est la fonction qui à $\theta \in \Theta$ associe la densité de X sous θ évaluée en X. Dans le cas iid, la vraisemblance est donnée par*

$$V(\theta, X) = \prod_{i=1}^{n} f_\theta(X_i), \theta \in \Theta.$$

La vraisemblance $\theta \mapsto L_X(\theta)$ est une fonction aléatoire (puisque dépendante de X) qui est d'autant plus élevée que θ rend probable l'observation de X. On appelle estimateur du maximum de vraisemblance toute variable aléatoire $\hat{\theta} = \hat{\theta}(X)$ qui maximise $L(., X)$.

Dans un modèle régulier, déterminer un estimateur du maximum de vraisemblance est rarement difficile. Même dans les cas où une solution analytique n'est pas connue, il est souvent possible de s'approcher numérique de la solution. L'algorithme EM est utilisé pour des modèles dont la vraisemblance est une fonction compliquée, mais qui peut être rendue régulière par l'introduction de variables non observées.

Un exemple simple: Dans le modèle Gaussien $X_i \sim \mathcal{N}(m, \sigma^2)$, la vraisemblance est donnée par

$$V(m, \sigma^2, X) = \frac{1}{(\sqrt{2\pi}\sigma)^n} \exp\Big(- \frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - m)^2 \Big), \ m \in \mathbb{R}, \ \sigma > 0.$$

L'objectif étant de maximiser cette fonction, on peut tout aussi bien considérer la log-vraisemblance $v(m, \sigma, X) := \log V(m, \sigma, X)$, qui vaut ici

$$v(m, \sigma^2, X) = -\frac{n}{2} \log\big(2\pi\sigma^2\big) - \frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - m)^2.$$

On montre alors facilement que le maximum de vraisemblance est atteint en

$$\hat{m} = \frac{1}{n} \sum_{i=1}^n X_i \quad \text{et} \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \hat{m})^2.$$

Si on suppose maintenant l'existence de deux types d'individus (A ou B) dans la population, chacun issu d'un modèle Gaussien. La loi de la variable aléatoire X_i en fonction du type est donc

$$X_i \sim \begin{cases} \mathcal{N}(m_A, \sigma_A^2) & \text{si } X_i \text{ est de type } A, \\ \mathcal{N}(m_B, \sigma_B^2) & \text{si } X_i \text{ est de type } B. \end{cases}$$

Si le type de chaque individu n'est pas observé, on considère alors le modèle dit de mélange Gaussien, de densité

$$x \mapsto \frac{p}{\sqrt{2\pi}\sigma_A} \exp\Big(- \frac{(x - m_A)^2}{2\sigma_A^2} \Big) + \frac{1-p}{\sqrt{2\pi}\sigma_B} \exp\Big(- \frac{(x - m_B)^2}{2\sigma_B^2} \Big),$$

où p représente la probabilité qu'un individu donné soit de type A. Soit $\theta = (m_A, m_B, \sigma_A, \sigma_B, p)$, la vraisemblance du modèle de mélange, donnée par

$$V(\theta, X) = \prod_{i=1}^n \left[\frac{p}{\sqrt{2\pi}\sigma_A} \exp\Big(- \frac{(X_i - m_A)^2}{2\sigma_A^2} \Big) + \frac{1-p}{\sqrt{2\pi}\sigma_B} \exp\Big(- \frac{(X_i - m_B)^2}{2\sigma_B^2} \Big) \right],$$

ne permet pas d'obtenir une expression analytique simple de ses maximiseurs.

Si les types des variables étaient connus, la vraisemblance auraient une forme plus simple. Posons $Z_i = 1$ si X_i est de type A et 0 sinon, on suppose les Z_i iid de loi de Bernoulli de paramètre p. La log-vraisemblance des observations (X_i, Z_i) vaut à une constante additive près

$$v(\theta, X, Z) = \sum_{i=1}^n \left[\log\Big(\frac{p}{\sigma_A}\Big) - \frac{(X_i - m_A)^2}{2\sigma_A^2} \right] Z_i + \left[\log\Big(\frac{1-p}{\sigma_B}\Big) + \frac{(X_i - m_B)^2}{2\sigma_B^2} \right] (1 - Z_i).$$

On calcule facilement les estimateurs du maximum de vraisemblance dans ce cas, en considérant les échantillons $(X_i)_{i:Z_i=1}$ et $(X_i)_{i:Z_i=0}$ séparemment. L'idée de l'algorithme EM consiste à remplacer la fonction $v(., X, Z)$ qui est inconnue par son espérance sachant X, sous une valeur fixée θ_0 du paramètre. L'estimateur à l'étape n est calculé comme le maximiseur de cette espérance. Puis, l'espérance est actualisée en prenant l'estimateur comme nouvelle valeur, et ainsi de suite jusqu'à ce que la suite converge.

Algorithme EM: Soit $\theta \in \Theta$ le paramètre du modèle dont on choisit une valeur initiale θ_0 arbitrairement. On note Z la variable non-observée qui rend simple le calcul du maximum de vraisemblance. A chaque itération k, on alterne les deux étapes suivantes:

- *(Espérance):* On évalue la quantité $Q(\theta_k, \theta) := \mathbb{E}_{\theta_k}\big(v(\theta, X, Z)\big|X\big)$ qui est l'espérance sous θ_k de la log-vraisemblance de (X, Z) conditionnellement à X. Autrement dit, l'aléa dans l'espérance ne provient que de Z, dont la loi est induite par θ_k.
- *(Maximisation):* On construit θ_{k+1} qui maximise $\theta \mapsto Q(\theta_k, \theta)$.

Proposition 5.2 *La suite* $\big(V(\theta_k, X)\big)_{n\in\mathbb{N}}$ *est croissante en* k.

Preuve. On aura besoin des deux lemmes suivants.

Lemme 5.3 *Soit* Y *une variable aléatoire de densité* f *et* g *une densité définie sur le même espace telles que* $\mathbb{E}\big|\log g(Y)\big|$ *et* $\mathbb{E}\big|\log f(Y)\big|$ *soient finis. Alors,*

$$\mathbb{E}\big(\log g(Y)\big) \leq \mathbb{E}\big(\log f(Y)\big).$$

Preuve du lemme. Par l'inégalité de Jensen appliquée à la fonction logarithme (concave),

$$\mathbb{E}\big(\log g(Y)\big) = \mathbb{E}\left(\log \frac{g(Y)}{f(Y)}\right) + \mathbb{E}\big(\log f(Y)\big) \leq \log \mathbb{E}\left(\frac{g(Y)}{f(Y)}\right) + \mathbb{E}\big(\log f(Y)\big).$$

Or, $\mathbb{E}\big(g(Y)/f(Y)\big) = \int \big(g(y)/f(y)\big) f(y)dy = \int g(y)dy = 1.$ □

Lemme 5.4 *Pour tout* θ_k*, la fonction* $\theta \mapsto v(\theta, X) - Q(\theta_k, \theta)$ *est maximale en* $\theta = \theta_k$.

Preuve du lemme. Soit $g_\theta(z|X)$ la densité de Z sachant $X = x$ sous θ, on remarque que

$$v(\theta, X) - v(\theta, X, Z) = -\log g_\theta(Z|X).$$

En intégrant contre $g_{\theta_k}(.|X)$ (ce qui revient à évaluer sous l'espérance $\mathbb{E}_{\theta_k}(.|X)$), on trouve

$$v(\theta, X) - \mathbb{E}_{\theta_k}\big(v(\theta, X, Z)\big|X\big) = -\mathbb{E}_{\theta_k}\big(\log g_\theta(Z|X)\big|X\big).$$

Or, d'après le lemme précédent, $\mathbb{E}_{\theta_k}\big(\log g_\theta(Z|X)\big|X\big) \leq \mathbb{E}_{\theta_k}\big(\log g_{\theta_k}(Z|X)\big|X\big)$. $\square$

On en arrive à la propriété principale de l'algorithme: l'augmentation du passage de θ_k à θ_{k+1} sur la fonction $Q(\theta_k, .) = \mathbb{E}_{\theta_k}\big(v(., X, Z)\big|X\big)$ induit une augmentation plus importante encore sur la log-vraisemblance. En effet, on obtient directement par le lemme précédent,

$$0 \leq Q(\theta_k, \theta_{k+1}) - Q(\theta_k, \theta_k) \leq v(\theta_{k+1}, X) - v(\theta_k, X),$$

ce qui implique en particulier que la suite $v(\theta_k, X)$ (et donc $V(\theta_k, X)$) est croissante. $\square$.

En général, on stoppe l'algorithme dès que la différence $Q(\theta_k, \theta_{k+1}) - Q(\theta_k, \theta_k)$ passe sous un seuil $\epsilon > 0$ (petit) fixé à l'avance.

On a ici montré que l'algorithme EM permettait d'augmenter la vraisemblance à chaque itération, ce qui n'est pas suffisant pour avoir la convergence vers un maximum global (ou même local). Ce type de résultat est difficile à obtenir théoriquement. Pour optimiser les chances de converger vers le maximum global, il est recommandé de lancer l'algorithme plusieurs fois, avec des points de départ différents.

Exemple 1. Reprenons le modèle de mélange Gaussien, de densité

$$f_\theta(x) = \frac{p}{\sqrt{2\pi}\sigma_A} \exp\Big(-\frac{(x-m_A)^2}{2\sigma_A^2}\Big) + \frac{1-p}{\sqrt{2\pi}\sigma_B} \exp\Big(-\frac{(x-m_B)^2}{2\sigma_B^2}\Big).$$

où on pose $\theta = (m_A, \sigma_A, m_B, \sigma_B, p)$. Soit $Z_i = \mathbb{1}\{X_i \text{ de type } A\}$, on rappelle que la log-vraisemblance de (X, Z) s'écrit à une constante additive près

$$v(\theta, X, Z) = \sum_{i=1}^n \Big[\log\Big(\frac{p}{\sigma_A}\Big) - \frac{(X_i - m_A)^2}{2\sigma_A^2}\Big] Z_i + \Big[\log\Big(\frac{1-p}{\sigma_B}\Big) + \frac{(X_i - m_B)^2}{2\sigma_B^2}\Big](1 - Z_i).$$

Bien sûr, les Z_i ne sont pas observés, mais on peut calculer l'espérance de la vraisemblance sous une valeur quelconque $\theta_k = \big(m_{A,k}, \sigma_{A,k}, m_{B,n}, \sigma_{B,n}, p_k\big)$. En partant d'une valeur initiale θ_0 arbitraire, on effectue donc à chaque itération de l'algorithme l'étape d'espérance, conditionnellement à X. Comme seuls les Z_i sont traités comme aléatoires, il suffit ici de calculer les quantités $\tau_i = \tau_i(\theta_k) = \mathbb{E}_{\theta_k}(Z_i|X)$. De plus, Z_i n'est dépendant de X que par X_i, on a donc $\tau_i = \mathbb{E}_{\theta_k}(Z_i|X_i) = \mathbb{P}_{\theta_k}(Z_i = 1|X_i)$. Sous θ_k et conditionnellement à l'événement $Z_i = 1$, X_i suit une loi nomale de paramètres $m_{A,k}, \sigma_{A,k}$. On déduit par la formule de Bayes

$$\tau_i = \mathbb{P}_{\theta_k}(Z_i = 1|X_i) = f_{\theta_k}(X_i|Z_i = 1)\frac{\mathbb{P}_{\theta_k}(Z_i = 1)}{f_{\theta_k}(X_i)} = \frac{p_k \exp\big(-(X_i - m_{A,k})^2/2\sigma_{A,k}^2\big)}{\sqrt{2\pi}\sigma_{A,k} f_{\theta_k}(X_i)},$$

que l'on peut calculer explicitement. On déduit

$$\mathbb{E}_{\theta_k}\big(v(\theta,X,Z)\big|X\big)=\sum_{i=1}^n\Big[\log\Big(\frac{p}{\sigma_A}\Big)-\frac{(X_i-m_A)^2}{2\sigma_A^2}\Big]\tau_i+\Big[\log\Big(\frac{1-p}{\sigma_B}\Big)+\frac{(X_i-m_B)^2}{2\sigma_B^2}\Big](1-\tau_i).$$

Maximiser $\theta\mapsto\mathbb{E}_{\theta_k}\big(\ell(\theta,X,Z)\big|X\big)$ peut alors se faire analytiquement. En posant $\overline{\tau}=\sum_{i=1}^n\tau_i$, on obtient $p_{k+1}=\overline{\tau}/n$ et

$$m_{A,k+1}=\frac{1}{\overline{\tau}}\sum_{i=1}^n X_i\tau_i\quad,\quad m_{B,n+1}=\frac{1}{n-\overline{\tau}}\sum_{i=1}^n X_i(1-\tau_i)$$
$$\sigma^2_{A,k+1}=\frac{1}{\overline{\tau}}\sum_{i=1}^n\tau_i(X_i-m_{A,k+1})^2\quad,\quad\sigma^2_{A,k+1}=\frac{1}{n-\overline{\tau}}\sum_{i=1}^n(1-\tau_i)(X_i-m_{B,n+1})^2$$

Dans le cas du modèle de mélange, chaque itération de l'algorithme EM se fait donc explicitement.

Exemple 2. On teste la durée de vie de n ampoules sur une certaine période. On suppose que la durée de vie T_i de la i-ème ampoule suit une loi exponentielle de paramètre $\theta>0$, de densité $t\mapsto\theta e^{-\theta t}\mathbb{1}\{t>0\}$. La log-vraisemblance de $T=(T_1,...,T_n)$ est donc simplement

$$v(\theta,T)=n\log(\theta)-\theta\sum_{i=1}^n T_i.$$

On suppose maintenant qu'on n'observe ces durées de vie que sur un intervalle de temps fini $[0,t]$. Les observations sont alors les valeurs $X_i=\min\{T_i,t\}$. Soit θ_k une valeur arbitraire du paramètre, en remarquant que $T_i=X_i+(T_i-X_i)\mathbb{1}\{T_i>t\}$, on calcule l'espérance sachant X, sous θ_k:

$$\mathbb{E}_{\theta_k}\big(\ell(\theta,T)\big|X\big)=n\log(\theta)-\theta\sum_{i=1}^n X_i-\theta\sum_{i=1}^n\mathbb{E}_{\theta_k}\big((T_i-t)\mathbb{1}\{T_i>t\}\ \big|\ X_i\big).$$

Clairement $\mathbb{E}_{\theta_k}\big((T_i-t)\mathbb{1}\{T_i>t\}\ \big|\ X_i\big)=0$ si $X_i<t$. De plus, comme la loi exponentielle est sans mémoire, on a pour $X_i=t$, $\mathbb{E}_{\theta_k}\big((T_i-t)\mathbb{1}\{T_i>t\}\big|X_i\big)=\mathbb{E}_{\theta_k}\big((T_i-t)\big|T_i>t\big)=1/\theta_k$. Soit $\overline{X}_n=\frac{1}{n}\sum_{i=1}^n X_i$ et $p_n=\frac{1}{n}\sum_{i=1}^n\mathbb{1}\{X_i=t\}$, on obtient

$$\mathbb{E}_{\theta_k}\big(\ell(\theta,T)\big|X\big)=n\log(\theta)-n\theta\Big(\overline{X}_n+\frac{p_n}{\theta_k}\Big).$$

L'étape de maximisation de l'algorithme EM conduit à

$$\theta_{k+1}=\frac{1}{\overline{X}_n+p_n/\theta_k}.$$

On montre facilement que $\lim_{k\to\infty}\theta_k=(1-p_n)/\overline{X}_n$, le point fixe de $\theta\mapsto 1/(\overline{X}_n+p_n/\theta)$.