Rapport Projet de Synthèse

Une utilisation de la méthode dite de *Contrastive Divergence* pour l'apprentissage statistique

Axel Flatrès--Berthelot & Baptiste Doyen

(Binôme 97 / PAG 2)

Juin 2016

CentraleSupélec

Sommaire

Introduction

En termes de méthodologie statistique, il est d'usage de recourir à l'estimateur de maximum de vraisemblance pour estimer les paramètres liés à une distribution aléatoire. En effet, celui-ci permet de disposer d'un estimateur sans biais de manière simple et utile:

- Simple : il suffit de minimiser la vraisemblance $\boldsymbol{L(X,\theta)}$ et $\boldsymbol{\theta_{MV} = argmax_{\theta} L(X,\theta)}$
- Utile: il est asymptotiquement consistant, permet de disposer de l'information de Fisher et est optimal (dans la classe des estimateurs sans biais)

Néanmoins, outre ses qualités théoriques, l'estimateur de maximum de vraisemblance ne présente pas les mêmes avantages en ce qui concerne son calcul pratique.

En effet, si l'on considère une distribution de loi de probabilité de la forme suivante :

$$\boldsymbol{P(X,\theta) = \frac{1}{Z(\theta)} exp(X^T \theta X + B^T X)}$$

Le calcul de $\boldsymbol{\theta_{MV}}$ implique celui de $\boldsymbol{Z(\theta)}$.

Or, il se peut que $\boldsymbol{Z(\theta)}$ ne soit pas calculable en pratique (par exemple : s'il s'agit d'une somme contenant un nombre exponentiel de termes).

D'où la nécessité d'une autre méthode pour estimer en pratique le paramètre $\boldsymbol{\theta}$

L'objet de notre projet de synthèse dont ce texte en est la présentation est d'étudier cette méthode afin de la mettre en place algorithmiquement sur des cas simples (le cas gaussien par exemple).

Partie I : Théorie

1) Le modèle énergétique

Afin de réaliser une inférence statistique, c'est-à-dire le passage entre des observations empiriques à la modélisation de ces observations à travers une loi de probabilité, nous allons choisir de toujours utiliser une loi de probabilité de même forme.

Cette forme est la suivante et est inspirée des résultats de physique statistique (statistique de Maxwell-Boltzmann) :

$$\mathbf{P(X, W)} = \frac{\mathbf{1}}{\mathbf{Z(\theta)}} \mathbf{exp(X^T WX)}$$

Nous appellerons ce modèle, le modèle énergétique. L'énergie associée a pour expression :

$$\boldsymbol{E(X, W)} = \boldsymbol{-X^T WX}$$

Ainsi, l'énergie associée à X est élevée si la probabilité correspondante $\boldsymbol{P(X, W)}$ est faible et vice-versa.

Ce qui est cohérent si l'on interprète ce résultat avec une approche plus « physique » : les configurations les plus probables sont celles qui requièrent le moins d'énergie. Et elles le sont d'autant plus, que l'énergie associée est minimale.

L'un des points forts de ce modèle est son caractère universel : il obéit à un principe selon lequel le plus probable correspond à une minimisation de l' « énergie » nécessaire à sa réalisation.

2) L'apprentissage par Contrastive Divergence

Hypothèse : on suppose que la distribution aléatoire observée obéit à la règle du modèle énergétique et possède donc la densité associée.

Dans ce cas, on pose

$$\text{a) } \boldsymbol{p_0} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\delta(x - x_n)}$$

La distribution expérimentale, c'est-à-dire celle à partir de laquelle l'algorithme « apprend ».

$$\text{b) } \boldsymbol{p_\infty} = \boldsymbol{p(W)}$$

(La distribution finale, c'est-à-dire celle à partir de laquelle découle l'observation. La théorie qui existerait derrière l'expérimental.)

Au lieu de minimiser la vraisemblance, la méthode dite de Contrastive Divergence se propose de minimiser la différence entre deux divergences, appelées des divergences de Kullback-Leibler :

$$\boldsymbol{CD_n(W) = KL(p_0 \parallel p_\infty) - KL(p_n \parallel p_\infty)}$$

Où $$\boldsymbol{KL(p_0 \parallel p_\infty) = \sum_x p_0(x)\, log\, \frac{p_0(x)}{p(x,W)}}$$

La divergence de Kullback-Keiber entre deux distributions de probabilité mesure la distance entre ces deux distributions (attention : elle est non-symétrique).

Afin de déterminer $\boldsymbol{\widehat{W} = argmin_W\, CD_n(W)}$, on utilise une méthode dite de descente de gradient.

On définit la suite $\boldsymbol{W_n}$ de la sorte :

$$\boldsymbol{W_{n+1} = W_n + \frac{\partial CD_n(W)}{\partial W}_{W_n}}$$

Cette suite converge vers la valeur souhaitée, à savoir $\boldsymbol{\widehat{W}.}$

Reste à déterminer $\boldsymbol{\frac{\partial CD_n(X,W)}{\partial W}_{W_n}}$: pour cela nous allons utiliser l'approximation de Hinton.

3) L'approximation de *Geoffrey E. Hinton*

D'après ce qui précède :

(Par souci de simplification on prend $\frac{\partial}{\partial W} = \frac{\partial}{\partial W_n}$)

$$\frac{\partial CD_n(X,W)}{\partial W} = \frac{\partial KL(p_0 \parallel p_\infty)}{\partial W} - \frac{\partial KL(p_n \parallel p_\infty)}{\partial W}$$

Or $\frac{\partial KL(p_0 \| p_\infty)}{\partial W} = \langle \frac{\partial \log p_0}{\partial W} \rangle_{p_0} - \langle \frac{\partial \log p_\infty}{\partial W} \rangle_{p_0}$ (p_0 indépendant de W_n)

De même : $\frac{\partial KL(p_n \| p_\infty)}{\partial W} = \langle \frac{\partial \log p_n}{\partial W} \rangle_{p_n} - \langle \frac{\partial \log p_\infty}{\partial W} \rangle_{p_n}$

Et : $- \langle \frac{\partial \log p_\infty}{\partial W} \rangle_{p_0} + \langle \frac{\partial \log p_\infty}{\partial W} \rangle_{p_n}$ est considéré comme négligeable.(Hinton)

D'où : $\frac{\partial CD_n(X,W)}{\partial W} \approx \langle \frac{\partial \log p_0}{\partial W} \rangle_{p_0} - \langle \frac{\partial \log p_n}{\partial W} \rangle_{p_n} \approx \langle \mathbf{X}^\mathbf{T}\mathbf{X} \rangle_{p_0} - \langle \mathbf{X}^\mathbf{T}\mathbf{X} \rangle_{p_n}$

p_n n'est pas indépendant de W_n mais par effet de différence le calcul de la fonction de partition est ainsi évité par différence des deux quantités ci-dessus.

Afin de déterminer $\langle \mathbf{X}^\mathbf{T}\mathbf{X} \rangle_{p_n}$, on utilise l'échantillonnage de Gibbs.

4) L'échantillonnage de Gibbs

Une forme ***sans variables cachées***, à partir des lois jointes :
on la nomme Echantillonnage de Gibbs cyclique et elle repose sur la relation suivante.

Soit **(X^j)** une famille de vecteurs de dimension n telle que **$X^j = (x_1^j, \dots, x_n^j)$.**

$$x_i^j \sim P\left(x_i^j \middle| x_{-i}^j\right) = \frac{1}{1 + e^{-\sum_{j \neq i} \theta_{i,j} x_j}}$$

avec x_{-i}^j le vecteur X^j ôté de sa composante x_i^j.

Etant donné un vecteur, suite à l'échantillonnage de ses composantes grâce à la loi jointe ci-dessus, on peut réaliser une descente de gradient pour obtenir une nouvelle matrice des paramètres.

Partie II : implémentation de la solution

1) Gaussienne unique

Objectif : déterminer les paramètres d'une gausienne à savoir sa moyenne mu et sa variance sigma.

Entrées : m valeurs générées aléatoirement à partir de $\mathcal{N}(\mu, \sigma^2)$.

Algorithme :

```
%taux d'apprentissage
epsilon = 10^(-3);

n = 10;
m = 1000;

%génération des training examples

mu = 45;
sigma2 =10;

Y = normrnd(mu,sigma2,[1 m]);

%conversion en binaire des données
Y_int = floor(Y);
Y_binary = de2bi(Y_int,n);

X1 = zeros(n,m);
X2 = zeros(n,m);

%Initialisation des paramètres
W0 = zeros(n,n);
```

```
%Début entraînement du modèle sur les training examples


for i =1:m

        X1(:,i) = Y_binary(i,:);

        for d = 1:n

            s1 = 0;

            for c = 1:n
                if c ~= d
                s1 = s1 + W0(d,c)*X1(c,i);
                end;
            end;

            p1 = (1/(1+exp(-s1)));
            p0 = 1-p1;
            X2(d,i) = randsample(0:1,1,true,[p0 p1]);

        end;

%Début descente de Gradient

        for a=1:n

               for b=1:n

                    W0(a,b) = W0(a,b) +
                    epsilon*0.5*(X1(a,i)*X1(b,i)-
                    X2(a,i)*X2(b,i));

            end;

        end;

%Fin descente de Gradient

end;

%Fin entraînement du modèle sur les training examples
```

```
%Début de la comparaison entre les résultats obtenus et le modèle
théorique générateur.

k = 100;
X = zeros(1,k);
result = zeros(1,k);


for i = 1:k
    X(i)=i;
end;


X_binary = de2bi(X,n);


for i = 1:k
     result(i) =
     exp(0.5*X_binary(i,:)*W0*transpose(X_binary(i,:)));
end;


%Détermination de la constante de normalisation
Q = trapz(X,result);


%Densité théorique
for i = 1:k
     Xnormal(i) = (1/sqrt(sigma2*2*pi))*exp(-0.5*(X(i)-mu)*(X(i)-
mu)/sigma2);
end;


%Affichage des résultats
figure
plot(X,result/Q,'*');
hold on
plot(X,Xnormal,'*');
```

2) Deuxième cas : reconnaître un mélange de gausienne (*Gaussian Mixture)*

Objectif : déterminer les paramètres d'un mélange de gausiennes à savoir les moyennes mu1,mu2,…,mu(n), les variance sigma^{2}1, sigma^{2}2,…, sigma(n) et les poids associées à chaque gaussienne 1,2,…,n

Entrées : m valeurs générées aléatoirement à partir de $\sum_i p_i \mathcal{N}(\mu_i, \sigma^2{}_i)$.

Algorithme :

L'algorithme est le même que celui qui précède à la différence de :

```
%génération des training examples

d = 1;
mu = [20;50];

k = [5 5];

sigma = zeros(1,d,length(k));

for i = 1:length(k)
        sigma(:,:,i)=k(i);
end;

%poids des gaussiennes
p = [1 2];

%génération de l'objet gmdistribution
obj = gmdistribution(mu,sigma,p);


Y = random(obj,m);



%Densité théorique

Xnormal = pdf(obj,transpose(X)) ;
```

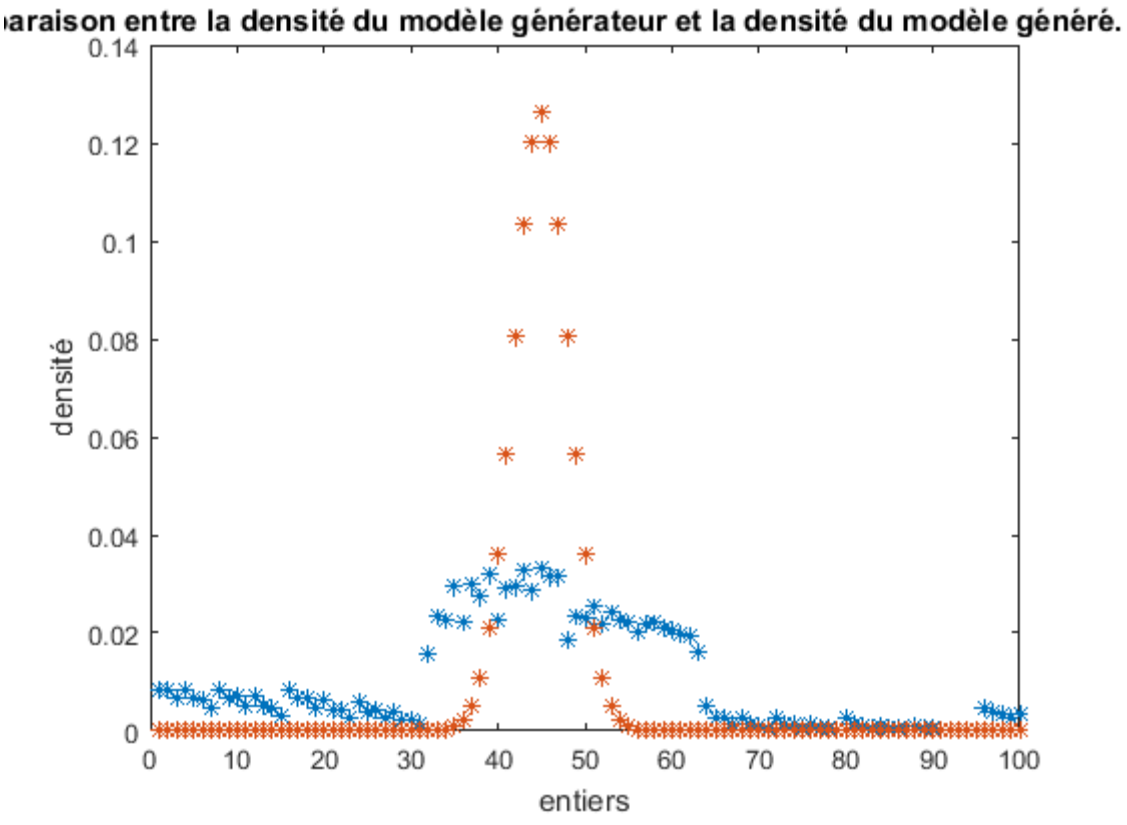
Partie III : Résultats

1) Gaussienne unique

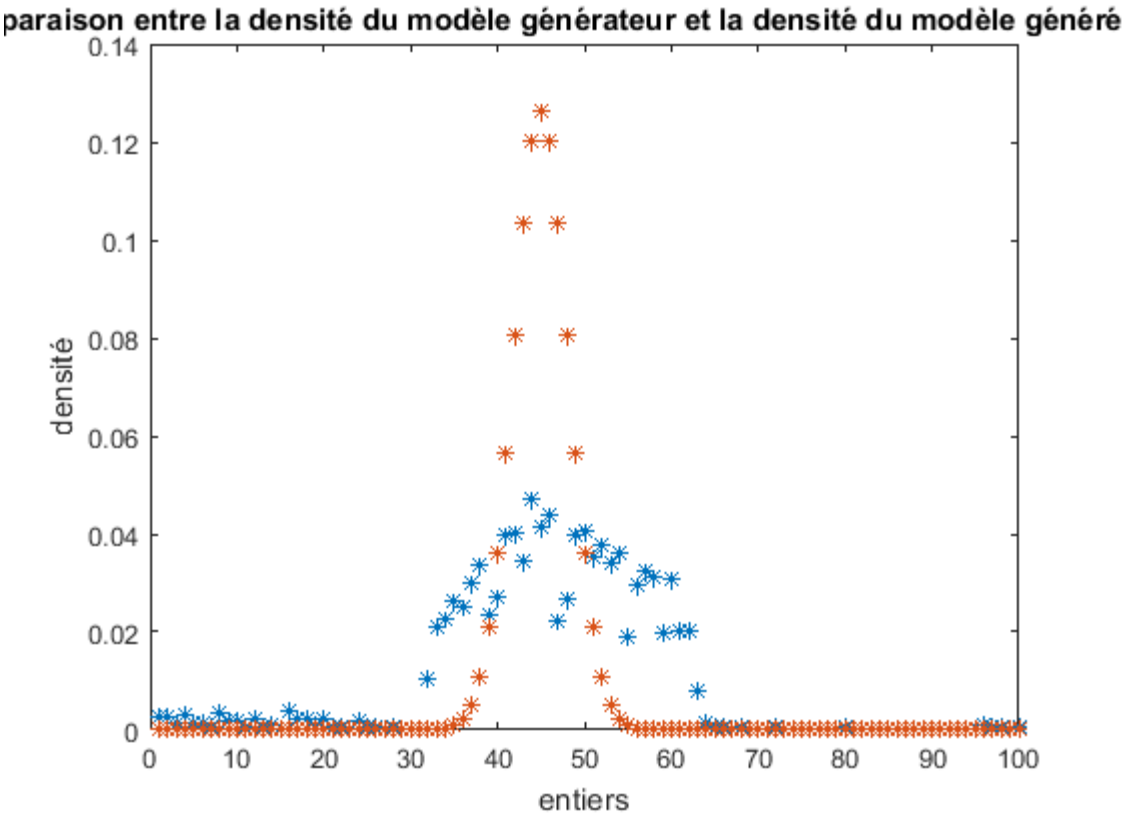
Comparaison entre la densité du modèle générateur des valeurs initiales et la densité du modèle généré par descente du gradient.

Paramètres : (mu,sigma2) = (45,5)

m = 10 000



m = 50 000



Interprétation :

- Les valeurs sont symétriques par rapport à la valeur moyenne
- On distingue un pic central autour de la valeur moyenne.
- Plus m augmente, plus la figure est satisfaisante

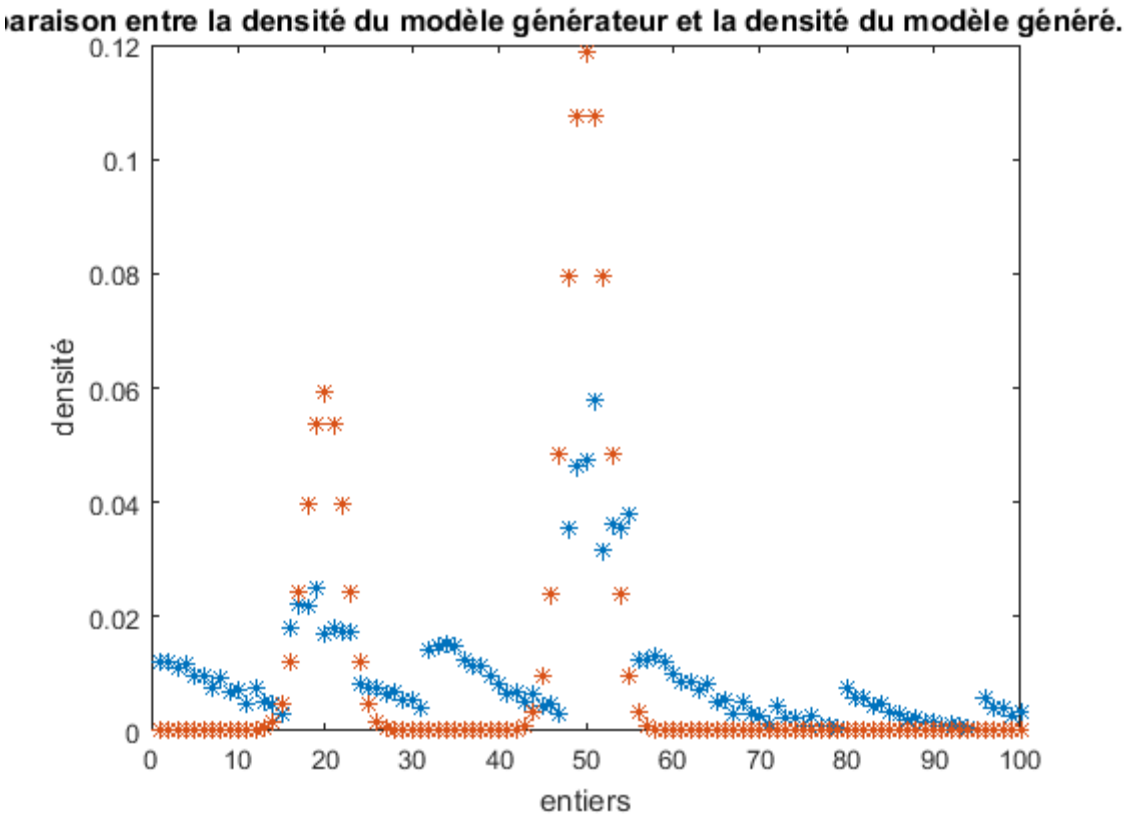
2) Mélange de gausiennes

Comparaison entre la densité du modèle générateur des valeurs initiales et la densité du modèle généré par descente du gradient.

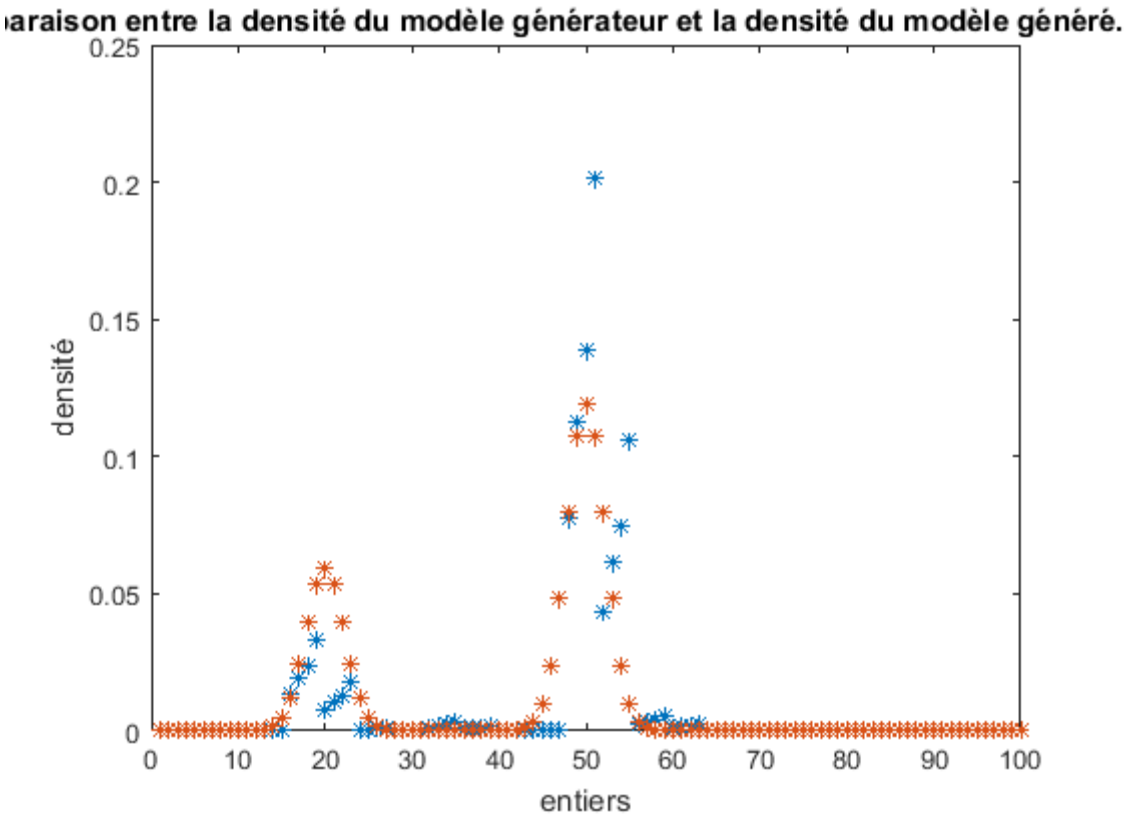
Paramètres : (mu1,sigma²1) = (20,5) et (mu2,sigma²2) = (50,5)

Poids = (1,2)

m = 1000



m = 5000



Interprétation :

- Les valeurs sont symétriques par rapport à la valeur moyenne
- On distingue un pic central autour de la valeur moyenne.
- Contrairement au cas précédent, les valeurs sont moins recentrées autour de la valeur moyenne (surtout pour le deuxième pic)
- Plus m augmente, plus la figure est satisfaisante

Conclusion

Dans le cadre de ce projet, nous nous sommes familiarisés avec Matlab. Cela a été très enrichissant.

De plus, les résultats obtenus semblent plutôt satisfaisants au regard des comparaison graphiques présentées ci-dessus.

Les valeurs prises ont de plus été réduites à des entiers, ce qui a rajouté de l'imprécision aux résultats obtenus.

Pour conclure, cette méthode peut être qualifiée d'efficace car avec peu de calculs et de lignes de code, on peut retrouver des valeurs satisfaisantes du modèle théorique.

Nous tenons également à remercier chaleureusement notre tuteur M. Pablo Piantanida pour son tutorat et son aide au cours de ce projet.