

# RAPPORT TP2 INF8225

---

Elliot Sisteron (1807165)

## Partie 1

L'objectif de cette partie du TP est de se familiariser l'apprentissage automatique via la régression logistique. Pour ce faire, nous allons utiliser l'approche de descente du gradient sur un problème de classification de documents.

Nous allons ici comparer deux méthodes : l'approche par descente du gradient régulière ainsi qu'une variante stochastique par mini-batch. Nous écrirons ici un logiciel en Matlab pour étudier les performances de ces deux algorithmes.

On utilise le jeu de données *20news\_w100* qui correspond à une BDD d'articles. On est intéressé par 100 mots clés pour classer les articles.

### Descente du gradient

Voici le code matlab :

```
1 load 20news_w100.mat
2 n = 4;
3 m = size(newsgroups,2);
4 o = ones(1,m);
5 i = 1:m;
6 j = newsgroups;
7 Y = sparse(i,j,o,m,n);
8
9 Theta = rand(4,101)-.5;
10 X = documents;
11 X = [X ; ones(1,16242)];
12 taux_dapprentissage = 0.0005;
13
14 maxIterations = 50;
15
16 per = randperm(m);
17
18 % Creation exemplaire apprentissage
19 tailleApprentissage = round(70*m/100);
20 XApprentissage = zeros(101,tailleApprentissage);
21 YApprentissage = zeros(tailleApprentissage, n);
22 for i=1:tailleApprentissage
23     XApprentissage(:,i) = X(:,per(i));
24     YApprentissage(i,:) = Y(per(i),:);
25 end
26
```

```

27 A = transpose(XApprentissage*YApprentissage);
28
29 % Creation exemplaire training
30 tailleTraining = round(15*m/100);
31 XTraining = zeros(101,tailleTraining);
32 YTraining = zeros(tailleTraining, n);
33 for i=tailleApprentissage+1:tailleApprentissage+tailleTraining
34     XTraining(:,i-tailleApprentissage) = X(:,per(i));
35     YTraining(i-tailleApprentissage,:) = Y(per(i),:);
36 end
37
38 normes = [];
39 logVraisemblances = [];
40 precisions = [];
41
42 for k=1:maxIterations
43     B = zeros(4,101);
44
45     for i=1:tailleApprentissage
46         probas = zeros(1,n);
47         Z = 0;
48         for j=1:n
49             y = zeros(1,n);
50             y(j) = 1;
51             expo = exp(y * Theta * XApprentissage(:,i));
52             probas(j) = expo;
53             Z = Z + expo;
54         end
55         probas = probas / Z;
56         B = B + transpose(XApprentissage(:,i)*probas);
57     end
58
59 % Mise a jour parametres
60 G = A - B;
61 Theta = Theta + taux_dapprentissage * G;
62 normes = [normes norm(Theta)];
63
64 % Log vraisemblance conditionnelle
65 logVraisemblance = 0;
66 for i=1:tailleTraining
67     logVraisemblance = logVraisemblance + ...
68         YTraining(i,:)*Theta*XTraining(:,i);
69     Z = 0;
70     for j=1:n
71         y = zeros(1,n);
72         y(j) = 1;
73         expo = exp(y * Theta * XTraining(:,i));
74         Z = Z + expo;
75     end
76     logVraisemblance = logVraisemblance - log(Z);
77 end

```

```

77
78     logVraisemblances = [logVraisemblances logVraisemblance];
79
80     % Precision ensemble d'apprentissage et ensemble validation
81     ok = 0;
82     for i=1:tailleTraining
83         maxProba = -1;
84         maxj = 0;
85         for j=1:n
86             y = zeros(1,n);
87             y(j) = 1;
88             proba = exp(y*Theta*XTraining(:,i));
89             if proba > maxProba
90                 maxProba = proba;
91                 maxj = j;
92             end
93         end
94         if YTraining(i,maxj) == 1
95             ok = ok + 1;
96         end
97     end
98     precision = ok / tailleTraining;
99
100     precisions = [precisions precision];
101 end

```

Les résultats sont les suivants :

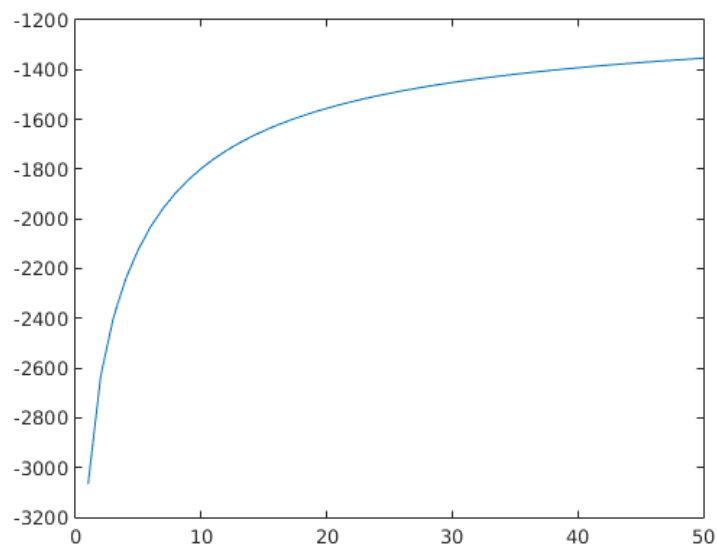


FIGURE 1 – Log vraisemblances

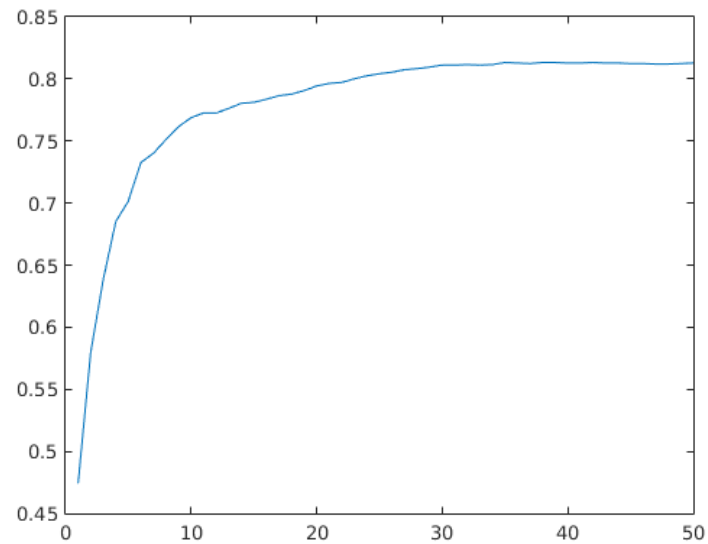


FIGURE 2 – Précision

## Descente du gradient par minibatch

Voici le code matlab :

```

1  load 20news_w100.mat
2  n = 4;
3  m = size(newsgroups,2);
4  o = ones(1,m);
5  i = 1:m;
6  j = newsgroups;
7  Y = sparse(i,j,o,m,n);
8
9  Theta = rand(4,101)-.5;
10 X = documents;
11 X = [X ; ones(1,16242)];
12 taux_dapprentissage = 0.0005;
13
14 maxIterations = 150;
15
16 normes = [];
17 logVraisemblances = [];
18 precisions = [];
19
20 % mini batch init
21 nBatch = 20;
22 tailleBatch = floor(m/nBatch);
23 batch=zeros(nBatch,tailleBatch);
24
25
26 for k=1:maxIterations
27     per = randperm(tailleBatch*20);

```

```

28     for i=1:nBatch
29         batch(i,:) = per(1+tailleBatch*(i-1):tailleBatch*i);
30     end
31
32     for l=1:nBatch
33         B = zeros(4,101);
34         for i=1:tailleBatch
35             ii = batch(l,i);
36             probas = zeros(1,n);
37             B = B + transpose(X(:,ii)*Y(ii,:));
38             Z = 0;
39             for j=1:n
40                 y = zeros(1,n);
41                 y(j) = 1;
42                 expo = exp(y * Theta * X(:,ii));
43                 probas = probas + expo*y;
44                 Z = Z + expo;
45             end
46             probas = probas / Z;
47             B = B + transpose(X(:,ii)*probas);
48         end
49         % Mise a jour parametres
50         G = B/tailleBatch;
51         Theta = Theta + taux_dapprentissage * G;
52     end
53
54     normes = [normes norm(Theta)];
55
56     % Log vraisemblance conditionnelle
57     logVraisemblance = 0;
58     for i=1:tailleBatch*5
59         logVraisemblance = logVraisemblance + Y(i,:)*Theta*X(:,i);
60         Z = 0;
61         for j=1:n
62             y = zeros(1,n);
63             y(j) = 1;
64             expo = exp(y * Theta * X(:,i));
65             Z = Z + expo;
66         end
67         logVraisemblance = logVraisemblance - log(Z);
68     end
69
70     logVraisemblances = [logVraisemblances logVraisemblance];
71
72     % Precision ensemble d'apprentissage et ensemble validation
73     ok = 0;
74     for i=1:tailleBatch*5
75         maxProba = -1;
76         maxj = 0;
77         for j=1:n
78             y = zeros(1,n);

```

```

79         y(j) = 1;
80         proba = exp(y*Theta*X(:,i));
81         if proba > maxProba
82             maxProba = proba;
83             maxj = j;
84         end
85     end
86     if Y(i,maxj) == 1
87         ok = ok + 1;
88     end
89 end
90 precision = ok / (tailleBatch*5);
91
92 precisions = [precisions precision];
93 end

```

Ici, l'algorithme est moins rapide; il met plus de temps à converger. Toutefois, il est plus performant : la précision finale est bien meilleure.

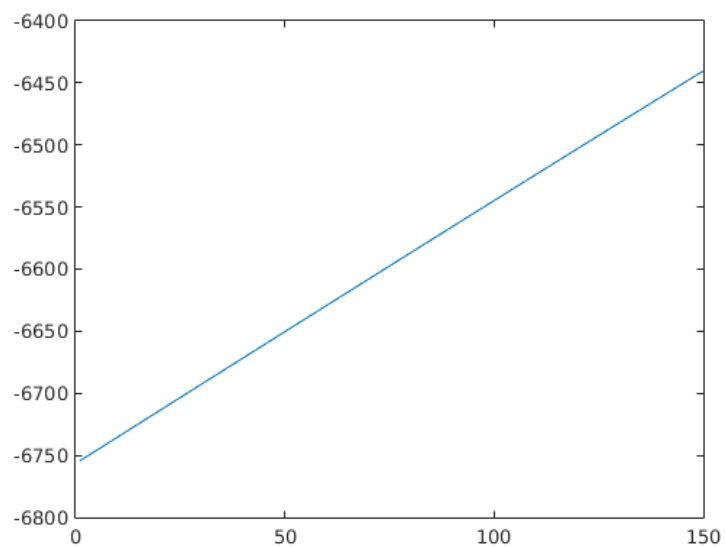


FIGURE 3 – Log vraisemblances

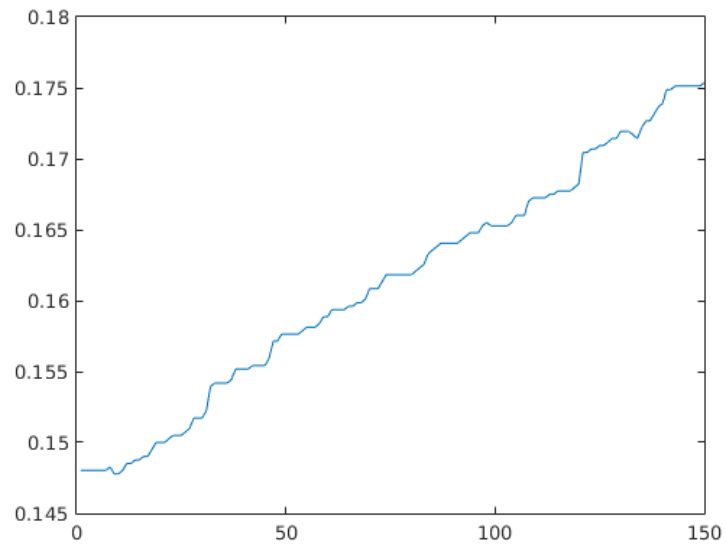


FIGURE 4 – Précision

## Partie 2

On va ajouter 100 caractéristiques binaires à chaque exemple. On les échantillonne selon l'énoncé.

```

1 load 20news_w100.mat
2 n = 4;
3 m = size(newsgroups,2);
4 o = ones(1,m);
5 i = 1:m;
6 j = newsgroups;
7 Y = sparse(i,j,o,m,n);
8
9 Theta = rand(4,101)-.5;
10 X = documents;
11 X = [X ; ones(1,16242)];
12 taux_dapprentissage = 0.0005;
13
14 maxIterations = 150;
15
16 normes = [];
17 logVraisemblances = [];
18 precisions = [];
19
20 % mini batch init
21 nBatch = 20;
22 tailleBatch = floor(m/nBatch);
23 batch=zeros(nBatch,tailleBatch);
24
25 % lambdas

```

```

26 lambda1 = 0.7;
27 lambda2 = 1.8;
28
29 for k=1:maxIterations
30     per = randperm(tailleBatch*20);
31     for i=1:nBatch
32         batch(i,:) = per(1+tailleBatch*(i-1):tailleBatch*i);
33     end
34
35     for l=1:nBatch
36         B = zeros(4,101);
37         for i=1:tailleBatch
38             ii = batch(l,i);
39             probas = zeros(1,n);
40             B = B + transpose(X(:,ii)*Y(ii,:));
41             Z = 0;
42             for j=1:n
43                 y = zeros(1,n);
44                 y(j) = 1;
45                 expo = exp(y * Theta * X(:,ii));
46                 probas = probas + expo*y;
47                 Z = Z + expo;
48             end
49             probas = probas / Z;
50             B = B + transpose(X(:,ii)*probas);
51         end
52         % Mise a jour parametres
53         RL1 = (1/20)*(lambda1*2*Theta);
54         RL2 = (Theta < 0) + (Theta > 0);
55         RL2 = RL2*lambda2/20;
56         G = RL1 + RL2;
57         Theta = Theta + taux_dapprentissage * G;
58     end
59
60     normes = [normes norm(Theta)];
61
62     % Log vraisemblance conditionnelle
63     logVraisemblance = 0;
64     for i=1:tailleBatch*5
65         logVraisemblance = logVraisemblance + Y(i,:)*Theta*X(:,i);
66         Z = 0;
67         for j=1:n
68             y = zeros(1,n);
69             y(j) = 1;
70             expo = exp(y * Theta * X(:,i));
71             Z = Z + expo;
72         end
73         logVraisemblance = logVraisemblance - log(Z);
74     end
75
76     logVraisemblances = [logVraisemblances logVraisemblance];

```



```

77
78     % Precision ensemble d'apprentissage et ensemble validation
79     ok = 0;
80     for i=1:tailleBatch*5
81         maxProba = -1;
82         maxj = 0;
83         for j=1:n
84             y = zeros(1,n);
85             y(j) = 1;
86             proba = exp(y*Theta*X(:,i));
87             if proba > maxProba
88                 maxProba = proba;
89                 maxj = j;
90             end
91         end
92         if Y(i,maxj) == 1
93             ok = ok + 1;
94         end
95     end
96     precision = ok / (tailleBatch*5);
97
98     precisions = [precisions precision];
99 end

```

Voici les résultats :

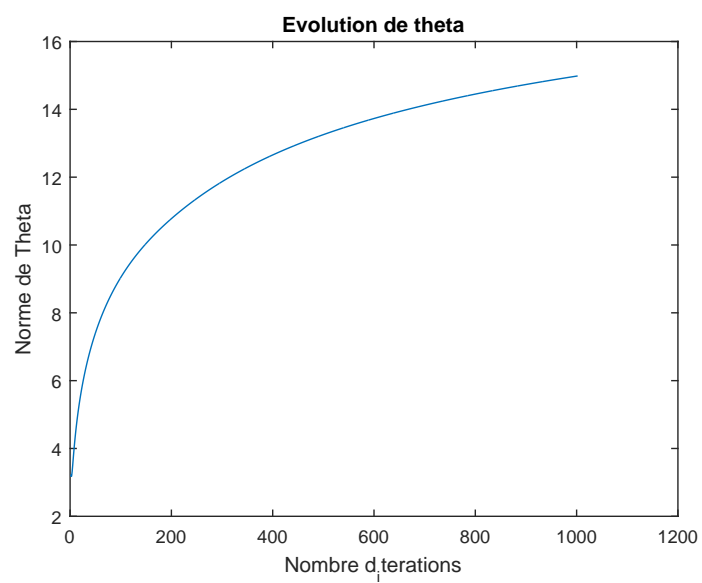


FIGURE 5 – Norme  $W$  sans mini batch

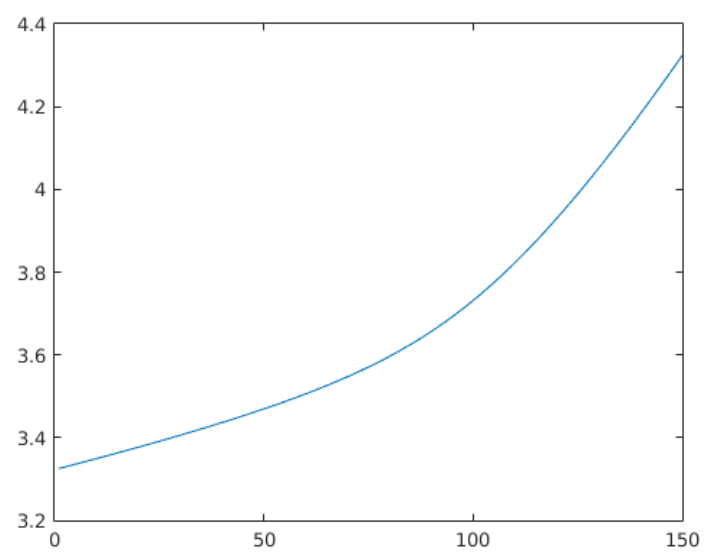


FIGURE 6 – Norme W avec mini batch