Modélisation et reconnaissance des formes
Gradient stochastique

1 Présentation de la méthode

L'algorithme du gradient stochastique est une méthode de descente de gradient utilisée pour la minimisation d'une fonction objectif qui est écrite comme une somme de fonctions différentiables.

C'est un cas très fréquent dans l'estimation aux moindres carrés par exemple ($min_{a,b} \sum (y(i) - ax_i - b)^2$ ainsi que dans le domaine de la classification supervisée où on minimise la somme des distances entre valeur attendue et valeur prédite par le système sur l'ensemble des données disponibles.

La plupart du temps on cherche à trouver la valeur de w minimisant le risque empirique :

$$Q(w) = \frac{1}{n} \sum_{1}^{n} Q_i(w)$$

Une méthode de descente de gradient est le plus souvent employée. Il s'agit d'une méthode itérative construisant une suite w_n définie par

$$w_{n+1} = w_n - \mu \nabla Q(w) = w_n - \mu \frac{1}{n} \sum_{1}^{n} \nabla Q_i(w)$$

η est le pas d'itération. On l'appelle fréquemment *Taux d'apprentissage*.

En apprentissage, le nombre n de données utilisées peut être très grand (des millions de données...). Dans ce cas le calcul de la somme des gradients peut prendre un temps prohibitif.

Dans la méthode de descente du gradient stochastique (SGD), on procède à une simplification drastique. Au lieu de calculer le gradient du risque empirique ∇Q exactement, chaque itération estime ce gradient sur la base d'une seule mesure. L'algorithme réalise ainsi une mise à jour après chaque exemple.

— choix de w_0 et de η
  — for i=1 :n do
    choisir aléatoirement un exemple noté $l(i)$
    $w_{n+1} = w_n - \mu \nabla Q_{l(i)}(w)$
  — fin do

2 Mise en oeuvre sur un exemple jouet

Nous allons tester cette méthode sur un exemple très simple du calcul du minimum d'une fonction constituée de deux termes $f = \frac{1}{2}(f_1 + f_2)$. Au lieu de calculer le gradient de f et d'appliquer directement une descente de gradient, nous allons utiliser SGD et ainsi choisir aléatoirement une des deux fonctions et appliquer la descente de gradient soit sur f_1 soit sur f_2 en fonction du tirage effectué. La structure grossière de l'algo est écrite dans Algorithme 1).

Appliquer cet algorithme avec $f_1(x) = (x+1)^2$ et $f_2(x) = (x-1)^2$. Visualiser sur un graphique le tableau des x_i générés.

Répéter cette étape avec 5 initialisations différentes et afficher les résultats sur un même graphique.

Ajouter sur ce graphique le résultat obtenu par une descente de gradient *classique*.

Algorithm 1 Minimisation par gradient stochastique

$niter = 1000$
choisir aléatoirement une valeur initiale x_0
for $i = 1 : niter$ **do**
  choisir aléatoirement un nombre u entre 0 et 1
  $numfonc = (u > .5)$
  appliquer une itération de descente de gradient à partir de x_i avec $f1$ ou f_2 en utilisant $numfonc * f_1 + (1 - numfonc) * f_2$. Choisir $\mu = 1/(20 + i)$.
end for