Étudiants ingénieurs en aérospatial

Mémoire de 3e année

Théorie et Benchmark des méthodes de descente en vue d'une application au machine learning

Auteurs :
M. Audet Yoann
M. Chandon Clément
M. De Claverie Chris
M. Huynh Julien

Encadrant :
Pr. Peschard Cédric

Version 1.0 du
1er mai 2019

Remerciements

Table des matières

Chapitre 1

Introduction

Chapitre 2

Développement théorique des méthodes de descente

Le but de ce rapport est d'étudier les différentes possibilités afin de minimiser une fonction. Le problème se formule de la manière suivante :

$$\underset{x}{\operatorname{argmin}} f(x) \tag{2.1}$$

On cherche donc l'argument x tel que f(x) soit minimal.

2.1 Retour sur les méthodes de descente

2.1.1 Motivation de l'étude et de l'intérêt

2.1.2 Les méthodes de gradients

Les méthodes de gradients sont des méthodes de descentes.

2.2 Introduction aux espaces de Krylov

Dans cette partie, nous allons introduire un outil mathématique important qui permet la justification de la méthode du gradient conjugué : les espaces de Krylov.

2.2.1 Définition des espaces de Krylov

2.2.2 Quelques propriétés

2.2.3 Un premier algorithme : GMRES

2.3 L'approche par les espaces de Krylov du gradient conjugué

2.3.1 Définition du gradient conjugué

Grâce aux espaces de krylov, nous sommes capable d'améliorer les algorithmes de gradient qui sont présentés ci-dessus. Nous partons donc d'un algorithme du gradient basique :

$$\begin{cases} x_0 \in \mathbb{R} \text{ le choix initial} \\ x_{k+1} = x_k + \alpha_k(b - Ax_k) \end{cases} \tag{2.2}$$

Nous définissons alors le vecteur résidu $r_k = b - Ax_k$ appartenant à l'espace de Krylov d'ordre k et définit par le résidu à l'origine : r_0. De ce fait, il vient que x_{k+1} appartient à l'espace affine composé par le point x_0 et le k$^{\text{ieme}}$ espace de krylov : $\mathscr{K}_k$. Afin de simplifier notre exposé, nous supposons que nous avons affaire à des matrices symétriques définies positives.

Dans la méthode du gradient conjugué, nous n'allons pas choisir la définition comme dans les autres méthodes de gradient mais nous avons d'autres critères qui sont plus intéressant :

— Le premier repose sur le principe d'orthogonalisation :

$$\exists x_{k+1} \in [x_0 + \mathscr{K}_k], r_{k+1} \perp \mathscr{K}_k \tag{2.3}$$

— Le principe de minimisation :

$$\exists x_{k+1} \in [x_0 + \mathscr{K}_k], \frac{1}{2} < Ax, x > - < b, x > \text{ est minimal} \tag{2.4}$$

Dans le cas d'une matrice symétrique définie positive, les deux choix ci-dessus donnent la même solution x_{k+1} d'où le choix de A symétrique définie positive.
Définissons ensuite le vecteur direction :

$$d_k = x_{k+1} - x_k \tag{2.5}$$

De part la construction de d, il est possible de déduire quelques propriétés :

— L'espace de de Krylov est une combinaison linéaire des vecteurs r_k et d_k

$$\mathscr{K}_k = vectr_0, ..., r_k = vectd_0, ..., d_k \tag{2.6}$$

— Tous les vecteurs de la suite sont orthogonaux :

$$\forall 0 \leq l < k \leq n-1, < r_k, r_l >= 0 \tag{2.7}$$

— Les vecteurs de la suite (d_k) sont conjugué selon le produit vectoriel défini par A :

$$\forall 0 \leq l < k \leq n-1, < Ad_k, d_l >= 0 \tag{2.8}$$

C'est grâce à cette dernière propriété que le gradient conjugué est appelé ainsi.

2.3.2 Propriétés du gradient conjugué

2.3.3 Algorithme du gradient conjugué

2.3.4 Quelques mots sur le gradient conjugué non-linéaire

Le but est ici de voir si l'on peut améliorer l'algorithme ci-dessus pour l'appliquer à des fonctions quadratiques non convexes voir non linéaires. Nous n'allons présenter ici que la méthode de Fletcher-Reeves. Cette méthode consiste en deux petits changement dans l'algorithme du CG.

La première modification consiste à changer le calcul du pas de descente. En effet, nous

2.4 D'autres algorithmes de minimisation

Dans cette partie, nous nous intéressons aux algorithmes de Newton qui sont d'autres algorithmes permettant une nouvelle approche des problèmes de minimisation.

2.4.1 Les algorithmes de Newton

Le but d'un algorithme de Newton est d'approximer notre fonction par le développement de Taylor de celle-ci à l'ordre 2. On peut donc dès lors remarquer que nous aurons besoin de la seconde dérivée de cette fonction (ou de la Hessienne en dimension supérieure). Le développement de Taylor est :

$$f(x+h) = f(x) + < h, \nabla f(x) > + \frac{1}{2} < h, H_f(x)h > + o(||h||^2) \tag{2.9}$$

Le principe est assez simple : on se trouve au point x et nous allons choisir le point x+h comme prochain point tel que ce nouveau point minimise le développement de Taylor ci-dessus. Ce développement de Taylor s'apparente à une fonction quadratique.

Dans ce type de méthode, nous choisissons notre direction de descente de la manière suivante :

$$d_k = -\nabla^2 f_k^{-1} \nabla f_k \tag{2.10}$$

Bien entendu, dans la réalité, nous ne calculons pas l'inverse de la hessienne mais nous résolvons un système linéaire. Cette méthode peut donc paraître un peu lourde.

2.4.2 Les algorithmes de Quasi-Newton

Les algorithmes de Quasi-Newton se basent sur les algorithmes de Newton. Cependant, l'amélioration vient du fait qu'ils ne nécessitent pas la dérivée seconde de la fonction que l'on cherche à minimiser. En effet, nous allons choisir une direction de descente de la manière suivante :

$$d_k = -B^{-1} \nabla f_k \tag{2.11}$$

Avec B une matrice définie positive qui est recalculée à chaque itération afin d'approximer la valeur de la hessienne.

Chapitre 3

Comparaison logiciel des méthodes

chris je sais pas comment tu veux organiser ta partie

Chapitre 4

Applications des méthodes de descente au Machine Learning

4.1 Introduction et Motivation

4.2 Application des algorithmes

Chapitre 5

Conclusion

Table des figures