

Linear discriminant analysis con CUDA C

Francesco Polvere *

October 30, 2018

Abstract

In questo documento viene presentata una implementazione dell'algoritmo di linear discriminant analysis tramite computazioni l'utilizzo della GPU sulla piattaforma CUDA. L'implementazione parallela viene confrontata con l'implementazione classica in C, misurando le caratteristiche principali.

1 Introduction

Da una analisi attenta del metodo di LDA, si può notare la presenza di numerose operazioni che presentano un alto grado di parallelizzabilità

2 Analisi dello stato dell'arte

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec

varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3 Modello teorico

La LDA è un metodo utilizzato per trovare una combinazione lineare di features che caratterizzino o separano due o più classi di oggetti o eventi. Il combinatore risultante può essere utilizzato come classificatore lineare o ancor prima della classificazione, come operazione di riduzione di dimensionalità.

3.1 LDA Multiclasse

Supponiamo di avere n classi. La Within-scatter matrix calcolata come:

$$S_w = \sum_{i=1}^C \sum_{x \in C_i} (x - x_i)(x - x_i)'$$

Mentre la between-scatter matrix calcolata come:

$$S_b = \sum_{i=1}^n m_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})'$$

*F. Polvere, Corso di GPU computing, A/A 2017-2018, Università degli studi di Milano, via Celoria 28, Milano, Italia
E-mail: francesco.polvere@studenti.unimi.it

m_i	Calcolo media delle classi
M	Calcolo media totale
$(x-x_i)$	Differenza tra vettori
$(x-x_i)'$	Calcolo vettore trasposto

Dove \bar{x} é la media totale di tutte le classi, m_i é il numero di dati di training per ogni classe.

Dopo aver ottenuto S_b e S_w , vogliamo trovare l'equazione lineare che massimizza l'equazione in figura.

$$J(W) = \frac{|W^T S_b W|}{|W^T S_w W|}$$

Si pu dimostrare che la trasformazione W pu essere ottenuta risolvendo un problema sugli autovalori:

$$S_b W = \lambda S_w W$$

3.2 Parallelizzazione di LDA utilizzando CUDA

L'algoritmo LDA consiste in varie step mostrati in tabella 1. Operazioni su GPU.

4 Simulazione ed esperimenti

Sono stati sviluppati due algoritmi. Quello sequenziale e quello parallelo.

4.1 Dataset

Per il benchmark é stato utilizzato il dataset Iris, introdotto da Ronald Fisher. Il dataset consiste in tre classi, ognuna delle quali rappresenta una specie di pianta, con un numero di istanze di 50 per ogni classe, per un totale di 150 istanze. Le variabili considerate sono quattro e consistono nella lunghezza e larghezza del sepal e del petalo.

5 Risultati ottenuti

Il codice stato eseguito su un computer con le seguenti caratteristiche:

- GPU: NVIDIA GeForce 610M, compute capability 2.1, architettura FERMI (2.1) con 48 Cores/SM.
- CPU: Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz, 2401 Mhz, 4 core, 8 processori logici

6 Conclusioni

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.