DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

TP2 - Reconocimiento de Dígitos

May 18, 2022 Métodos Numéricos

Grupo 8

Integrante	LU	Correo electrónico
Cappella Lewi, F. Galileo	653/20	galileocapp@gmail.com
Anachure, Juan Pablo	99/16	janachure@gmail.com
La Tessa, Octavio	477/16	octalate@hotmail.com

En este trabajo utilizamos técnicas matriciales para reconocer dígitos en imágenes

Palabras clave:

Reconocimiento de Dígitos	C++
Primary Component Analysis (PCA)	K-Nearest Neighbors

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (++54 +11) 4576-3300

https://exactas.uba.ar

Secciones

1 Introducción

2 Desarrollo

Para el análisis armamos un módulo de `python`.

2.1 Modelo

2.2 Auto α

2.3 Implementación

El módulo fue implementado en `c++` usando la librería Eigen y con pybind11 lo compilamos para usarlo desde `python`.
Aprovechamos la librería OpenMP para paralelizar las operaciones de Eigen.

2.3.1 Optimización KNN

Originalmente calculábamos los k vecinos más cercanos calculando la distancia entre , lo que es $\Theta(TODO)$.
Esto fue mejorado para en cambio calcular una matriz $D \in \mathbb{R}^{n \times m}$ que en la posición $D_{i,j}$ tiene la distancia al cuadrado entre el vector de entrenamiento i y el vector de prueba j, lo que reduce las operaciones necesarias para encontrar las k distancias más chicas y es más simple de paralelizar. En total (sin paralelizar) toma $\Theta(TODO)$ esto implica una mejora sustancial en el tiempo calcular lo buscado.

2.4 Datos

Los datos de dígitos los obtuvimos de la competencia "Digit Recognizer" de Kaggle

3 Experimentación

3.1 Maximización

3.1.1 Parámetros

3.1.2 Tamaño de entrenamiento

3.2 Limitación PCA

4 Conclusión

I Justificación Optimización KNN

Siendo $A \in \mathbb{R}^{n \times d}, B \in \mathbb{R}^{m \times d}$, quiero armar una matriz $D \in \mathbb{R}^{n \times m} : D_{i,j} = ||row_i(A) - row_j(B)||_2^2$.
Para ello, tomo $S_A \in \mathbb{R}^{n \times m} : S_{Ai,j} = ||row_i(A)||_2^2, S_B \in \mathbb{R}^{n \times m} : S_{Bi,j} = ||row_i(B)||_2^2$.
Y demuestro que $D = S_A - 2AB^t + S_B$:
$D = S_A - 2AB^t + 2_B \iff$
$D_{i,j} = (S_A - 2AB^t + 2_B)_{i,j} \quad \leftarrow X = Y \iff X_{i,j} = Y_{i,j}$
$= S_{Ai,j} - (2AB^t)_{i,j} + S_{Bi,j} \quad \leftarrow (X - Y)_{i,j} = X_{i,j} - Y_{i,j}$
$= S_{Ai,j} - 2(AB^t)_{i,j} + S_{Bi,j} \quad \leftarrow \alpha X_{i,j} = \alpha X_{i,j}$
$= S_{Ai,j} - 2row_i(A)col_j(B^t) + S_{Bi,j} \quad \leftarrow (XY)_{i,j} = row_i(X)col_j(Y)$
$= S_{Ai,j} - 2row_i(A)row_j^t(B) + S_{Bi,j} \quad \leftarrow row_i^t(X) = col_i(X^t)$
$= ||row_i(A)||_2^2 - 2row_i(A)row_j^t(B) + ||row_j(B)||_2^2 \quad \leftarrow$ Definición previa
$= row_i^t(A)row_i(A) - 2row_i(A)row_j^t(B) + row_j^t(B)row_j(B) \quad \leftarrow ||v||_2^2 = v^t v$
$= (row_i(A) - row_j(B))^t(row_i(A) - row_j(B)) \quad \leftarrow (v - w)^t(v - w) = v^t v - 2v^t w + w^t w$
$= ||row_i(A) - row_j(B)||_2^2 \ \square \quad \leftarrow ||v||_2^2 = v^t v$

II Demostración SVD

Figuras

Bibliografía