
Grid Search

Cabrera Coyotzi Kevin Merlin
kcabrerac2200@alumno.ipn.mx

Badillo Perez Jose Antonio
jbadillop2200@alumno.ipn.mx

Barbosa Hernández Jayme Said
jbarbosah2200@alumno.ipn.mx
Unidad Profesional Interdisciplinaria en Ingeniería Campus Tlaxcala, IPN

1 Aspectos generales

Esta práctica se realizó en un ordenador portátil Huawei D15 que tiene un procesador AMD Ryzen 7 5700U y una memoria RAM de 16 GB. Se empleó el sistema operativo Ubuntu, versión 24.04.3 LTS.

También se utilizó el conjunto de datos Fashion MNIST, que incluye 60000 ejemplos para entrenamiento y 10000 para prueba. El conjunto con el que trabajaremos es un archivo CSV que posee valores de 0 a 255 debido a la escala de grises de las imágenes, cuyas dimensiones son de 28x28 píxeles.

2 Experimentación

Para la experimentación de los datos con los que íbamos a trabajar de manera que del CSV con los datos de entrenamiento se dividió con una proporción 80-20 y el archivo CSV con los datos de prueba se quedó sin modificación alguna. Para tratar los valores de una manera más precisa se recurrió a una normalización estándar la cual pone el valor de los píxeles en un rango $[0,1]$ para poder mejorar el desempeño de los modelos.

En esta práctica se evaluaron tres modelos de aprendizaje supervisado:

1. Random Forest (RF).
2. Support Vector Machine (SVM).
3. K- Nearest Neighbors (KNN).

Para cada uno de los modelos se realizó una búsqueda manual de los hiperparámetros mediante combinaciones posibles, con un objetivo poder determinar los parámetros que maximicen el desempeño. Las métricas de evaluación para considerar la precisión del modelo se utilizaron las siguientes métricas:

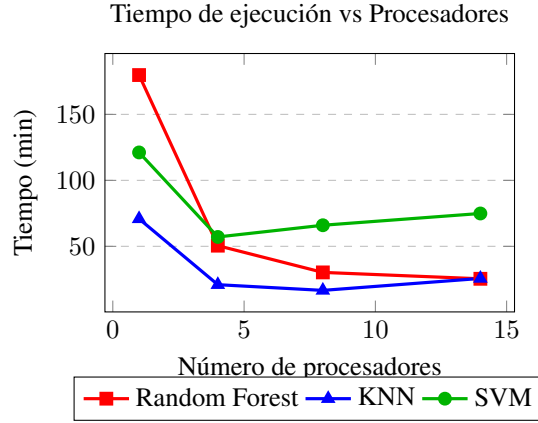
- Accuracy: porcentaje de predicciones correctas.
- F1-score: balance entre la precisión y recall considerado de manera global entre clases.
- Precision: proporción de verdaderos positivos entre predicciones positivas.
- Recall: Proporción de verdaderos positivos sobre positivos reales.

Estas evaluaciones se realizaron con los 3 modelos usando distintos número de procesadores disponibles a pesar de que el dispositivo empleado poseía la capacidad de 16 procesadores fue limitado a usar un máximo de 14 para el proceso de paralelización, debido a que se presentaron problemas de una sobrecarga y procedía a reiniciarse el dispositivo.

Este problema únicamente fue resuelto limitando el número máximo de procesadores para que fuera 14; investigado se encontró que el problema se debía por que se usaban otros procesos para la

división de las tareas y combinaciones y aparte se reservaba el número de procesadores a utilizar en la paralelización.

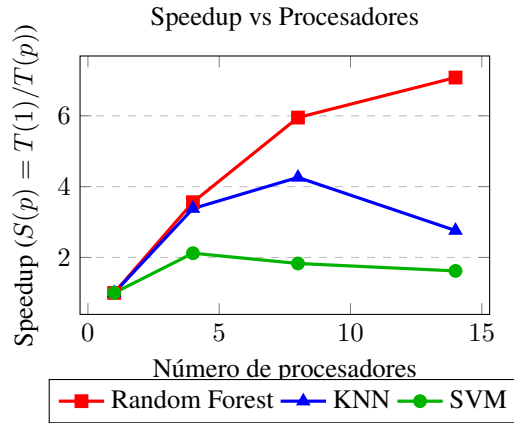
3 Reporte de los resultados de la búsqueda de hiperparámetros



| Modelo | 1 | 4 | 8 | 14 |
|--------|--------|-------|-------|-------|
| RF | 179.80 | 50.50 | 30.24 | 25.39 |
| KNN | 70.83 | 20.93 | 16.61 | 25.66 |
| SVM | 121.15 | 57.10 | 65.97 | 74.89 |

Table 1: Tiempos de ejecución (min) según número de procesadores.

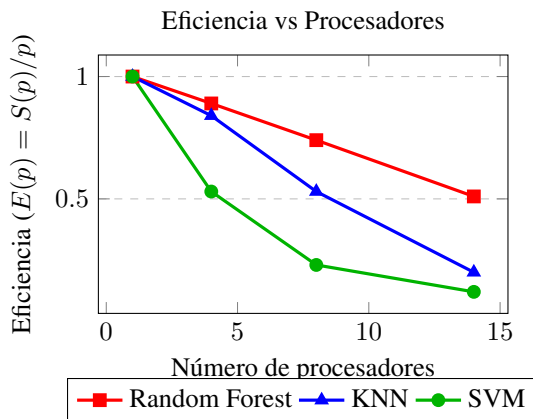
Figure 1: Comparación del tiempo de ejecución y datos numéricos.



| Modelo | 1 | 4 | 8 | 14 |
|--------|------|------|------|------|
| RF | 1.00 | 3.56 | 5.95 | 7.08 |
| KNN | 1.00 | 3.38 | 4.26 | 2.76 |
| SVM | 1.00 | 2.12 | 1.83 | 1.62 |

Table 2: Speedup obtenido por cada modelo.

Figure 2: Comparación del Speedup con distintos números de procesadores.



| Modelo | 1 | 4 | 8 | 14 |
|--------|------|------|------|------|
| RF | 1.00 | 0.89 | 0.74 | 0.51 |
| KNN | 1.00 | 0.84 | 0.53 | 0.20 |
| SVM | 1.00 | 0.53 | 0.23 | 0.12 |

Table 3: Eficiencia de los modelos en función de los procesadores.

Figure 3: Comparación de la eficiencia paralela de cada modelo.

| Modelo | Accuracy | F1-score | Precisión | Recall | Mejores hiperparámetros |
|---------------|----------|----------|-----------|--------|---|
| Random Forest | 0.8820 | 0.8815 | 0.8820 | 0.8837 | {n_estimators: 250, max_depth: None, min_samples_split: 2, min_samples_leaf: 1} |
| KNN | 0.8632 | 0.8641 | 0.8657 | 0.8646 | {n_neighbors: 5, weights: 'distance', metric: 'manhattan'} |
| SVM | 0.9004 | 0.9010 | 0.9009 | 0.9017 | {kernel: 'rbf', C: 10, gamma: 'scale'} |

Table 4: Métricas de desempeño y mejores combinaciones de hiperparámetros obtenidos mediante *Grid Search*.

4 Reporte de los resultados de las 10 iteraciones

Table 5: Promedio y desviación estándar del Accuracy en 10 iteraciones.

| Modelo | Accuracy ($\mu \pm \sigma$) | Tiempo medio ($\mu \pm \sigma$) |
|---------------|-------------------------------|-----------------------------------|
| KNN | 0.861 ± 0.0023 | 913.49 ± 351.91 |
| SVM | 0.9018 ± 0.0016 | 464.33 ± 64.81 |
| Random Forest | 0.8829 ± 0.0018 | 223.63 ± 10.22 |

5 Conclusiones de los experimentos

En la parte de búsqueda de hiperparámetros, se puede observar en Figure 1, el tiempo de ejecución disminuye conforme se incrementa el número de procesadores, evidenciando el impacto positivo de la paralelización en la búsqueda de hiperparámetros. Sin embargo, no pasa lo mismo con todos los modelos, en el caso de SVM, sí disminuye al momento de paralelizarlo pero va aumentando poco conforme lo paralelizamos más e igualmente con KNN parece disminuir pero a los 14 procesadores aumenta su tiempo.

En Figure 2, se aprecia que el speedup obtenido presenta un comportamiento sublineal en la mayoría de los casos, lo que indica que los algoritmos no escalan de forma perfecta con el número de procesadores disponibles. Aun así, el incremento de rendimiento al pasar de 1 a 8 procesadores resulta significativo (en excepción de SVM que disminuye su speedup), demostrando la efectividad del cómputo paralelo para acelerar tareas de búsqueda exhaustiva como el Grid Search.

Por otro lado, la Figure 3 muestra que la eficiencia decrece a medida que se agregan más procesadores, un comportamiento esperado en entornos paralelos debido a la sobrecarga de comunicación y sincronización. Este fenómeno refleja la existencia de una cantidad óptima de procesadores a partir de la cual los beneficios marginales se reducen.

Finalmente, como se presenta en Table 4, los tres modelos alcanzaron métricas de desempeño muy cercanas, con valores altos de accuracy y F1-score. El mejor de todos para clasificar el dataset es Support Vector Machine, contando con un accuracy del 90.

Al momento de realizar las 10 repeticiones con el conjunto de prueba, el SVM obtuvo el Accuracy promedio más alto (0.9018), y además con la menor desviación estándar, lo que lo hace el modelo más preciso y estable.

El Random Forest tuvo un desempeño ligeramente menor (0.8829), pero con un tiempo de ejecución mucho menor, siendo más eficiente computacionalmente.

El KNN fue el más lento (más de 900 s) y con menor Accuracy, lo que sugiere que no escala bien en tiempo ni supera a los otros modelos en precisión.