

Evaluación de Support Vector Machine en reconocimiento de letras manuscritas usando el conjunto de datos EMNIST Letters

John Patricio Serrano Carrasco
Departamento de Ingeniería Informática
Universidad de Santiago de Chile
11 de Mayo de 2024

Resumen— Se explora la eficacia de Support Vector Machine (SVM) en el reconocimiento de letras manuscritas del dataset EMNIST Letters mediante tres experimentos en los que se evalúa el valor de accuracy total y por cada clase. El primero implementa SVM usando imágenes como vectores directos, obteniendo una accuracy competente. El segundo incorpora una reducción de dimensionalidad mediante Principal Component Analysis (PCA), ajustando las características a 128 componentes principales sin impactar negativamente la accuracy. El tercer experimento mejora marginalmente la accuracy al utilizar vectores de características derivados de una Red Convolutiva pre-entrenada. Los resultados confirman la robustez de SVM para problemas de clasificación y destacan la importancia del procesamiento previo de datos.

Palabras claves—Support Vector Machine, Principal Component Analysis, Redes Convolucionales y Reconocimiento de letras.

I. INTRODUCCIÓN

En la era actual, la capacidad de reconocer y procesar texto manuscrito automáticamente se ha vuelto cada vez más importante. Dentro de este contexto, el dataset EMNIST Letters ofrece una base para desarrollar y probar algoritmos de reconocimiento. Este estudio se centró en evaluar la efectividad de Support Vector Machine (SVM), un método de Aprendizaje Supervisado robusto y eficiente, para clasificar imágenes de letras manuscritas.

El enfoque de la experiencia fue triple. En el primer experimento, se utilizó SVM con los datos de imagen directamente como vectores de entrada. En el segundo experimento, se aplicó Principal Component Analysis (PCA) para reducir la dimensionalidad de los datos a 128 componentes principales antes de entrenar el modelo SVM, buscando comprender el impacto de la reducción de dimensionalidad en el desempeño del modelo. Finalmente, el tercer experimento exploró el uso de características extraídas mediante una Red Convolutiva simple, con el objetivo de comparar la efectividad de vectores de características más sofisticados en la precisión de clasificación.

Cada experimento mantuvo una metodología consistente en la selección de muestras, utilizando 1000 muestras por clase para entrenamiento y 100 muestras por clase para pruebas, con el fin de garantizar que no hubiera superposición entre los conjuntos de entrenamiento y prueba. Estos experimentos proporcionan perspectivas valiosas sobre cómo diferentes técnicas

de preprocesamiento y modelos de Aprendizaje Automático pueden influir en la tarea de reconocimiento de letras manuscritas, cubriendo tanto el ámbito teórico como implicaciones prácticas para desarrollos futuros.

En los tres experimentos se evalúa el valor de accuracy total y de cada clase, métrica de desempeño que mide la frecuencia con la que un modelo de Aprendizaje Automático predice correctamente el resultado [1].

II. DESARROLLO

El **primer experimento** se centró en evaluar la capacidad del clasificador SVM para reconocer letras manuscritas utilizando las imágenes provenientes del dataset directamente como vectores de características. Este experimento se realizó utilizando la biblioteca **scikit-learn**, aprovechando su implementación del clasificador SVM con un kernel RBF (Radial Basis Function) [2]. Las imágenes del conjunto de datos EMNIST, que constan de 28x28 píxeles, se utilizan **como vectores unidimensionales de 784 características**. Se configuró un **Pipeline** que incorpora el modelo SVM, permitiendo un flujo de trabajo automatizado y reproducible. Respecto a la configuración de la SVM, se utilizó el **Kernel RBF**, el cual es eficaz en la gestión de la no linealidad en los espacios de características complejos [2]. El valor de **C (Parámetro de regularización)** se mantuvo en su valor predeterminado de 1.0, equilibrando la clasificación correcta de los ejemplos de entrenamiento contra la maximización del margen de decisión [2]. Respecto al hiperparámetro **Gamma**, el valor predeterminado de “scale” se usó, el cual sirve para definir el alcance de influencia de las muestras individuales [2].

El **segundo experimento** incorporó PCA antes del entrenamiento del SVM para reducir la dimensionalidad de los datos a 128 componentes principales, con el objetivo de mejorar la eficiencia y desempeño del modelo. Se configuró un Pipeline que incluye PCA seguido por el SVM con el Kernel RBF. Este experimento fue diseñado para evaluar si una reducción significativa en la dimensionalidad podría preservar la calidad del reconocimiento. Respecto a hiperparámetros, **n_components** de PCA se fijó en 128, seleccionado para capturar la mayoría de la varianza informativa mientras se reducen los datos. Kernel, C y Gamma de SVM fueron los mismos que en el experimento 1 para mantener consistencia en las comparaciones.

El **tercer experimento** exploró el impacto de utilizar vectores de características extraídos de una Red Convolutiva en

la precisión de la SVM. Estos vectores fueron preentrenados y proporcionados en archivos de formatos .CSV y al igual que los experimentos 1 y 2, un Pipeline se configuró, donde el SVM con Kernel RBF procesa directamente estos vectores de entrada. Respecto a la configuración de la SVM, esta fue consistente con los experimentos anteriores, utilizando las mismas configuraciones de Kernel, C y Gamma para una evaluación equitativa.

Cada uno de los experimentos aseguró una metodología rigurosa en la selección de muestras, utilizando 1000 muestras por clase para el entrenamiento y 100 muestras por clase para las pruebas. Se emplearon técnicas de selección aleatoria para garantizar que no hubiera superposición entre los conjuntos de entrenamiento y prueba. La evaluación de cada modelo se realizó utilizando la función **accuracy_score** de scikit-learn, complementada con Matrices de Confusión y reportes de clasificación detallados obtenidos de la función **classification_report**, que permitieron un análisis profundo del rendimiento por clase.

Se desarrollaron funciones para evitar la repetición de código y obtener gráficos y tablas que permitan comparar los resultados obtenidos.

- **modelProcess:** Recibe un Pipeline y los conjuntos de entrenamiento y prueba con sus respectivos datos y etiquetas. Las salidas son la accuracy total del modelo y los resultados tras la prueba del modelo. La función utiliza el Pipeline y los conjuntos para entrenar el modelo y luego evaluarlo. Tras esto, se obtiene la accuracy total de la SVM y también el reporte de clasificación de cada clase, lo cual entrega distintas métricas.
- **createConfusionMatrix:** Recibe las etiquetas del conjunto de prueba y los resultados tras evaluar el modelo SVM. Retorna un arreglo de Numpy con la accuracy por cada clase. La función utiliza las predicciones realizadas por el modelo para crear la Matriz de Confusión. La matriz se imprime y gráfica, para luego obtener la accuracy asociada a cada clase, la cual se muestra como un arreglo de Numpy.
- **createClassAccuracyPlot:** Recibe el arreglo de Numpy que contiene la accuracy por cada clase y muestra un gráfico de barras. Para ello, inicialmente se crea un dataframe para registrar los valores de cada clase y poder graficar la información.
- **createTotalAccuracyPlot:** Gráfica el valor de la accuracy total obtenido por el modelo SVM.
- **experimentsAccuracyPlot:** Recibe el accuracy total de cada experimento y realiza un gráfico de barras para poder comparar los valores obtenidos.

Además de scikit-learn, se utiliza Pandas para la manipulación de datos, Numpy para operaciones matemáticas y Matplotlib junto con Seaborn para visualizar resultados mediante gráficos y realizar análisis comparativos.

III. RESULTADOS EXPERIMENTALES

Respecto al experimento 1, utilizando las imágenes directamente como vectores de entrada, este enfoque básico alcanzó una accuracy global respetable, cuyo valor aproximado es

0,88. Esto subraya la capacidad del kernel RBF de SVM para manejar efectivamente las altas dimensiones de los datos de entrada. Hay una notable variabilidad en la accuracy entre diferentes clases como se puede apreciar en la Figura 2, con la accuracy más baja siendo del 0,71 para la clase 9 y la más alta del 0,96 para la clase 23. Esta variabilidad puede ser indicativa de la diversidad en la complejidad visual de los caracteres o de desbalances en la representatividad de ciertas clases dentro del conjunto de datos, probablemente debido a similitudes en las formas de las letras.

Recall y F1-score también varían significativamente entre las clases. Por ejemplo, la clase 26 tiene un recall extremadamente alto de 0,97 y un F1-score del mismo valor mientras que la clase 7 muestra un recall más bajo de 0,68 y un F1-score de 0,71. Esto nuevamente puede reflejar una dificultad a la hora de predecir algunas letras.

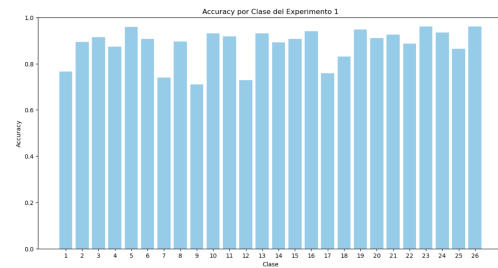


Figura 1: Accuracy por clase del experimento 1.

Clase	Accuracy	Recall	F1-Score	Support
1	0.77	0.88	0.82	100
2	0.89	0.92	0.91	100
3	0.91	0.95	0.93	100
4	0.87	0.89	0.88	100
5	0.96	0.92	0.94	100
6	0.91	0.88	0.89	100
7	0.74	0.68	0.71	100
8	0.89	0.85	0.87	100
9	0.71	0.76	0.73	100
10	0.93	0.93	0.93	100
11	0.92	0.89	0.90	100
12	0.73	0.75	0.74	100
13	0.93	0.95	0.94	100
14	0.89	0.83	0.86	100
15	0.91	0.98	0.94	100
16	0.94	0.95	0.95	100
17	0.76	0.72	0.74	100
18	0.83	0.88	0.85	100
19	0.95	0.92	0.93	100
20	0.91	0.91	0.91	100
21	0.93	0.87	0.90	100
22	0.89	0.85	0.87	100
23	0.96	0.97	0.97	100
24	0.93	0.86	0.90	100
25	0.86	0.89	0.88	100
26	0.96	0.97	0.97	100

Figura 2: Métricas por cada clase del experimento 1.

La variabilidad en la accuracy, recall y F1-score entre diferentes clases sugiere que mientras SVM es generalmente robusto, su rendimiento puede fluctuar significativamente dependiendo de las características específicas de cada clase. Algunas clases son clasificadas con alta accuracy y recall, mientras que otras muestran resultados notablemente inferiores.

El uso de un Kernel RBF, fue fundamental para lograr una accuracy total alta. Sin embargo, la variabilidad entre las clases indica que la configuración por defecto del Kernel puede no ser óptima para todas las clases, sugiriendo la necesidad de ajustes específicos, como el cambio de los valores C y Gamma. Al cambiar el valor de C de 1,0 a 10, la accuracy total aumenta 0,1, con algunas clases teniendo una pequeña variación de accuracy.

En el experimento 2, la reducción de dimensionalidad a 128 componentes mantuvo una precisión similar a la del primer experimento, obteniendo una accuracy total aproximada de 0,89, lo que sugiere que la mayoría de la información relevante para la clasificación se preservó. Este resultado es indicativo de que PCA puede ser una herramienta eficaz para mejorar la eficiencia computacional sin sacrificar significativamente el rendimiento.

Varias clases muestran un rendimiento excepcional, con accuracy y f1-score que alcanzan un valor de 0,9. Por ejemplo, como se puede observar en la 4, las clases 3, 5, 13, 15, 19, 23 y 26 exhiben altas métricas en todos los aspectos, destacando la efectividad del modelo en clasificar estos caracteres correctamente. Esto sugiere que las características esenciales para distinguir estas clases han sido eficazmente capturadas y mantenidas después de la reducción de dimensionalidad. Sin embargo, algunas clases como la 12 y 17 presentan una accuracy y f1-score notablemente más bajos. Esto sugiere que ciertas características distintivas de estas clases pueden no ser capturadas adecuadamente incluso después de la reducción de dimensionalidad, o bien, características cruciales para distinguir estas clases pueden haberse perdido durante el proceso de PCA. Comparado con el experimento 1, la accuracy total y las métricas por clase son relativamente similares, lo que indica que PCA ha sido efectivo sin sacrificar demasiado el desempeño.

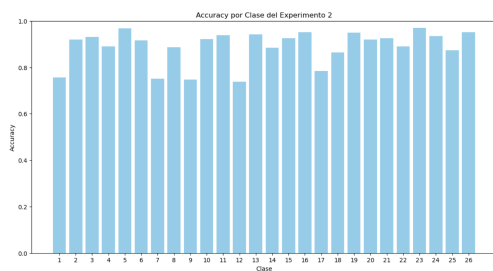


Figura 3: Accuracy por clase del experimento 2.

Podría ser útil experimentar con el número de componentes principales seleccionados en PCA para ver si un número ligeramente mayor de componentes podría mejorar la clasificación de clases que tienen un desempeño bajo. PCA, cuando se utiliza adecuadamente, puede ser una herramienta poderosa para la reducción de dimensionalidad en tareas de clasificación complejas sin una pérdida significativa de información crucial, manteniendo una alta eficiencia computacional y resultados de clasificación robustos. Al igual que con el experimento anterior, al aumentar el valor de C aumenta la accuracy total en 0,1.

El experimento 3 que utilizó vectores de características pre-

Clase	Accuracy	Recall	F1-Score	Support
1	0.76	0.87	0.81	100
2	0.92	0.92	0.92	100
3	0.93	0.95	0.94	100
4	0.89	0.89	0.89	100
5	0.97	0.92	0.94	100
6	0.92	0.88	0.90	100
7	0.75	0.72	0.73	100
8	0.89	0.85	0.87	100
9	0.75	0.77	0.76	100
10	0.92	0.94	0.93	100
11	0.94	0.91	0.92	100
12	0.74	0.76	0.75	100
13	0.94	0.96	0.95	100
14	0.88	0.84	0.86	100
15	0.92	0.98	0.95	100
16	0.95	0.96	0.96	100
17	0.78	0.76	0.77	100
18	0.86	0.89	0.88	100
19	0.95	0.95	0.95	100
20	0.92	0.92	0.92	100
21	0.93	0.87	0.90	100
22	0.89	0.89	0.89	100
23	0.97	0.98	0.98	100
24	0.93	0.86	0.90	100
25	0.87	0.89	0.88	100
26	0.95	0.97	0.96	100

Figura 4: Métricas por cada clase del experimento 2.

extraídos, logró una accuracy total ligeramente superior a los otros dos con un valor aproximado de 0,94, lo que sugiere que un método de extracción de características más sofisticado puede proveer una base más robusta para la clasificación.

La mayoría de las clases tienen una accuracy, recall y f1-score extremadamente altos, lo que sugiere que los vectores de características utilizados en este experimento son altamente efectivos para representar diferencias significativas entre clases. Por ejemplo, las clases 14 y 15 alcanzan un valor de accuracy y recall de 0,99 y 1,00 respectivamente, demostrando un reconocimiento casi perfecto. Aunque la mayoría de las clases muestran un rendimiento excepcional, hay algunas excepciones, como por ejemplo, la clase 8, la cual muestra una accuracy más baja de 0,69 y un f1-score de 0,69. Además, la clase 11 también muestra una accuracy y f1-score relativamente bajos de 0,73 y 0,72 respectivamente. Estos resultados indican que ciertas clases aún presentan desafíos en términos de diferenciación, posiblemente debido a similitudes visuales con otras clases.

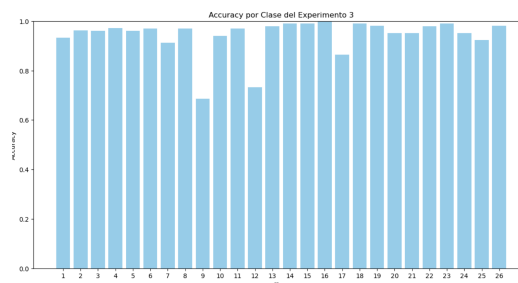


Figura 5: Accuracy por clase del experimento 3.

Sorprendentemente y a diferencia de los experimentos anteriores, al aumentar el valor de C la accuracy total disminuye en 0,1. Esto puede significar que existe un límite en el que la accuracy aumenta.

La consistencia en la accuracy entre los experimentos sugiere que SVM, particularmente con el kernel RBF, es robusto frente a variaciones en la naturaleza y calidad de las características de entrada. La eficacia marginalmente superior del tercer experimento puede atribuirse a la calidad de las

Clase	Accuracy	Recall	F1-Score	Support
0	0.93	0.96	0.95	100
1	0.96	1.00	0.98	100
2	0.96	0.96	0.96	100
3	0.97	1.00	0.99	100
4	0.96	0.98	0.97	100
5	0.97	0.98	0.98	100
6	0.91	0.84	0.87	100
7	0.97	0.97	0.97	100
8	0.69	0.70	0.69	100
9	0.94	0.93	0.93	100
10	0.97	0.96	0.96	100
11	0.73	0.71	0.72	100
12	0.98	0.97	0.97	100
13	0.99	0.95	0.97	100
14	0.99	0.96	0.97	100
15	1.00	0.98	0.99	100
16	0.86	0.89	0.88	100
17	0.99	0.96	0.97	100
18	0.98	0.99	0.99	100
19	0.95	0.96	0.96	100
20	0.95	0.96	0.96	100
21	0.98	0.94	0.96	100
22	0.99	0.99	0.99	100
23	0.95	0.98	0.97	100
24	0.92	0.97	0.95	100
25	0.98	0.99	0.99	100

Figura 6: Métricas por cada clase del experimento 3.

características utilizadas, que probablemente capturan mejor las diferencias entre las clases.

El experimento 3 muestra consistentemente la mayor accuracy en casi todas las clases. Esto sugiere que las características extraídas de manera avanzada son más efectivas para la clasificación en este contexto. El uso de PCA en el experimento 2 generalmente resulta en una accuracy total ligeramente superior al experimento 1, lo que indica que la reducción de dimensionalidad mediante PCA conserva las características esenciales necesarias para la clasificación efectiva en la mayoría de las clases. A pesar de la reducción drástica en la dimensionalidad en el experimento 2, este logra mantener un desempeño muy similar al experimento 1, lo que demuestra la eficacia de PCA en la preservación de información relevante.

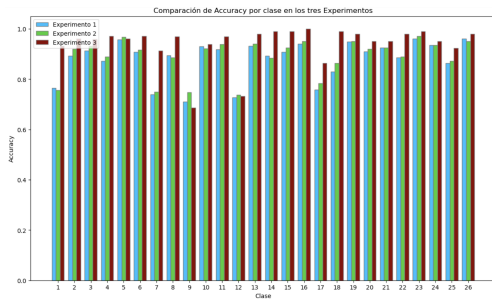


Figura 7: Accuracy por clase de cada experimento

De la Figura 7 se desprende que algunas clases muestran diferencias notables de rendimiento entre los experimentos. Por ejemplo, clases que tienen un desempeño moderado o bajo en los experimentos 1 y 2 ven mejoras significativas en el experimento 3, destacando el impacto de una mejor calidad de las características. Sin embargo, hay variaciones incluso dentro del experimento 3, lo que puede reflejar la dificultad inherente de algunas clases o la eficacia variable de las técnicas de extracción de características para ciertos tipos de caracteres. Algunas clases tienen un desempeño consistentemente bajo a través de los tres experimentos. Esto podría indicar la necesidad de estrategias específicas adicionales para esas clases.

Durante la experiencia se realizaron dos pruebas más. La primera fue incluyendo StandardScaler en el Pipeline para la normalización de los datos, pero esto provocó una baja de 0,2 en la accuracy total de los tres experimentos. También se probó cambiando el valor de Gamma, pero esto aumentó significativamente el tiempo de ejecución, con valores como 0,9 tomando más de 10 minutos de ejecución al menos con el experimento 1.

IV. CONCLUSIONES

La experiencia realizada a través de estos tres experimentos ilustra profundamente la capacidad y versatilidad de SVM en la clasificación de letras manuscritas. Además, esta serie de pruebas no solo validó la efectividad inherente de SVM en manejar grandes volúmenes de datos y su complejidad asociada, sino que también resaltó la importancia crítica de la preparación adecuada de los datos y el ajuste fino de los hiperparámetros del modelo. La consistencia de los valores de accuracy a través de los experimentos reafirma la robustez de SVM, mientras que las variaciones específicas de accuracy entre las clases en los diferentes experimentos subrayan la necesidad de una consideración cuidadosa del tipo y la calidad de los vectores de características utilizados.

El segundo experimento demostró que es posible manejar eficientemente la maldición de la dimensionalidad sin comprometer la accuracy. Sin embargo, el tercer experimento, destacó el potencial de explorar aún más el espacio de características para mejorar aún más la precisión del modelo.

El primer experimento destaca por su simplicidad y rapidez de implementación, aunque tiene un mayor tiempo a la hora de ejecutar el Pipeline en comparación a los otros experimentos. Esta aproximación directa facilita la configuración y comprensión del modelo, aunque enfrenta desafíos como la alta dimensionalidad, potencialmente reduciendo el rendimiento del modelo. En los experimentos subsiguientes, la introducción de PCA y el uso de vectores de características preentrenados ofrecen mejoras significativas en la accuracy y la gestión de la dimensionalidad. El uso de PCA ayuda a mejorar la eficiencia computacional, aunque a costa de perder información potencialmente importante. Por otro lado, el tercer experimento aprovecha características avanzadas que capturan mejor las variaciones específicas de cada clase, mejorando la precisión pero incrementando la complejidad y el costo computacional del procesamiento de datos.

Mientras que los resultados obtenidos son prometedores y demuestran la potencia de SVM, también plantea la pregunta importante sobre cómo se pueden mejorar aún más los resultados obtenidos con la búsqueda de los hiperparámetros óptimos para cada experimento.

REFERENCIAS

- [1] Evidently AI Team, "Accuracy vs. precision vs. recall in machine learning: what's the difference?" Recuperado el 10 de mayo de 2024, de: <https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall>, s.f.
- [2] Scikit-Learn, "Implementación svm de scikit-learn," Recuperado el 10 de mayo de 2024, de: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>, 2024.