

Tarea 1 - Aprendizaje Automático - SVM

Diego Alejandro Valdés Fernández
 Departamento de Ingeniería Informática
 Universidad de Santiago de Chile, Santiago, Chile
 diego.valdes.f@usach.cl

I. RESUMEN

Se evalúa el desempeño del Support Vector Machine (SVM) en la clasificación de imágenes del conjunto de datos QuickDraw 10 (10 clases, 3390 entrenamiento, 1166 prueba). Se comparan seis modelos: tres con kernel RBF y tres con kernel lineal. Cada grupo utiliza tres representaciones de las imágenes: vector completo, PCA y UMAP. El objetivo es determinar cómo el kernel y la representación de las imágenes influyen en la precisión de la clasificación de dibujos a mano.

II. INTRODUCCIÓN

Las Máquinas de Vectores de Soporte (SVM) son fundamentales en el aprendizaje automático por su capacidad para clasificar datos con alta precisión. De forma sencilla, en un espacio de dos dimensiones donde una SVM busca la mejor línea para separar distintas categorías. Sin embargo, para abordar datos más complejos y de mayor dimensión, las SVM utilizan los kernels. Estos permiten que la SVM opere de manera efectiva en espacios de dimensionalidad superior, encontrando separaciones complejas que serían imposibles con una simple línea recta en el espacio original. Gracias a esto, las SVM son aplicables en diversos campos, incluyendo el reconocimiento de imágenes, que es el aspecto principal de nuestro estudio para la clasificación de dibujos a mano.

Para este laboratorio se tenían inicialmente 10 000 imágenes para entrenamiento y 1 000 para prueba. Sin embargo, debido a limitaciones de hardware al iniciar los experimentos, la cantidad de datos de entrenamiento se redujo al 40 % del conjunto original. Finalmente, se utilizaron 3 390 imágenes para entrenamiento y 1 190 para prueba en los experimentos presentados. Estas limitaciones se debieron principalmente a la capacidad del equipo utilizado, cuyas características técnicas se aprecian en el siguiente listado.

- **Memoria RAM:** 8,00 GB (7,37 GB utilizables)
- **Procesador:** AMD Ryzen 5 4500U with Radeon Graphics
- **Sistema Operativo:** Windows de 64 bits
- **Arquitectura del Procesador:** x64

En esta investigación se realizaron seis experimentos con modelos SVM. Los tres primeros utilizaron una función *kernel* de base radial (RBF) y los tres restantes, un *kernel* lineal. Para cada tipo de *kernel*, se compararon tres estrategias de generación de vectores de características:

1. Vectores obtenidos directamente al convertir las imágenes en un formato unidimensional de tamaño 256x256.
2. Vectores de 256 dimensiones generados mediante Análisis de Componentes Principales (PCA).
3. Vectores de 256 dimensiones generados mediante la técnica de reducción de dimensionalidad no lineal UMAP.

Para evaluar el rendimiento de cada uno de los seis modelos SVM, se calculará la métrica de accuracy para la clasificación de las diez clases del conjunto de datos. Adicionalmente, se determinará el accuracy específico para cada modelo individualmente. Finalmente, se realizará una comparación del desempeño entre los modelos que utilizan el kernel RBF y aquellos que emplean el kernel lineal, con el objetivo de discernir la influencia de la elección del kernel y la estrategia de representación en la tarea de clasificación de dibujos a mano.

III. CONCEPTOS

A continuación, se presentan los principales conceptos utilizados en esta tarea, junto con una breve descripción de cada uno:

- **Kernel:** Función que transforma los datos a un espacio de mayor dimensión para permitir que un modelo lineal pueda separar clases no lineales.
- **RBF (Radial Basis Function):** Tipo de kernel que mide la similitud entre puntos según su distancia.
- **Linear:** Tipo de kernel que no transforma los datos, sino que aplica directamente un clasificador lineal en el espacio original.
- **Hiperparámetros (C y gamma):**
 - **C:** Controla el equilibrio entre un margen amplio y pocos errores de clasificación. Valores altos priorizan menor error en entrenamiento.
 - **Gamma:** Controla la influencia de cada punto en el kernel RBF. Valores bajos consideran puntos lejanos, y valores altos, solo cercanos.
- **GridSearch:** Método para encontrar la mejor combinación de hiperparámetros en un modelo, probando exhaustivamente distintas combinaciones posibles dentro de un conjunto definido de valores.
- **Accuracy por clase:** Porcentaje de muestras correctamente clasificadas dentro de una clase específica.

- **Accuracy del modelo:** Porcentaje global de muestras correctamente clasificadas sobre el total del conjunto de prueba.
- **Aplanar imágenes:** Proceso de convertir una imagen bidimensional (como 256×256) en un vector unidimensional para que pueda ser procesado por algoritmos de aprendizaje automático.
- **PCA (Principal Component Analysis):** Técnica de reducción de dimensionalidad que proyecta los datos en una nueva base ortogonal, preservando la mayor varianza posible.
- **UMAP (Uniform Manifold Approximation and Projection):** Técnica de reducción de dimensionalidad no lineal que conserva la estructura global y local de los datos, útil para visualizar o simplificar conjuntos complejos.

IV. EXPERIMENTOS REALIZADOS

La búsqueda de hiperparámetros se realizó utilizando la técnica de *GridSearch* aplicada sobre un subconjunto reducido del conjunto de entrenamiento, considerando únicamente 100 imágenes por clase. Los valores óptimos encontrados para **C** y, cuando correspondía, **gamma**, se mantuvieron fijos y se utilizaron posteriormente para entrenar los modelos con el 40 % del conjunto de datos total.

IV-A. Modelo 1: RBF + Vector unidimensional

Este modelo utiliza un kernel RBF junto con imágenes aplanadas de tamaño 256×256 convertidas directamente en vectores unidimensionales. Se realizó una búsqueda de hiperparámetros mediante validación cruzada para encontrar los valores óptimos de **C** y **gamma**, obteniéndose los siguientes resultados: **C = 10.0** y **gamma = auto**. El accuracy total alcanzado por la SVM fue de **0.666**.

Tabla I: Métricas por clase - Modelo 1

| Clase | Precision | Recall | F1-Score | Support |
|-----------------|-------------|-------------|-------------|-------------|
| Clase 0 | 0.68 | 0.69 | 0.68 | 100 |
| Clase 1 | 0.62 | 0.68 | 0.65 | 100 |
| Clase 2 | 0.71 | 0.56 | 0.63 | 100 |
| Clase 3 | 0.76 | 0.77 | 0.77 | 100 |
| Clase 4 | 0.58 | 0.67 | 0.62 | 100 |
| Clase 5 | 0.66 | 0.64 | 0.65 | 100 |
| Clase 6 | 0.71 | 0.66 | 0.68 | 100 |
| Clase 7 | 0.73 | 0.78 | 0.75 | 100 |
| Clase 8 | 0.71 | 0.73 | 0.72 | 100 |
| Clase 9 | 0.57 | 0.52 | 0.54 | 100 |
| Promedio | 0.67 | 0.67 | 0.67 | 1000 |

Es el modelo con mejor desempeño general (**accuracy 0.666**). El kernel RBF logra capturar no linealidades relevantes directamente desde los vectores aplanados. La matriz de confusión muestra una buena clasificación equilibrada entre la mayoría de las clases.

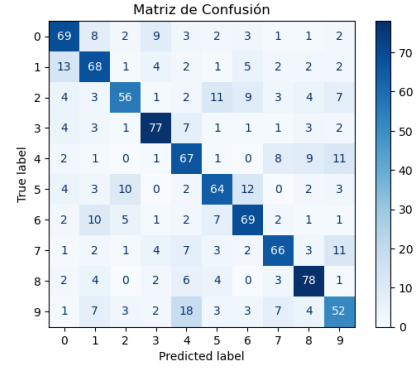


Figura 1: Matriz de confusión - Modelo 1

IV-B. Modelo 2: RBF + PCA

Este modelo utiliza un kernel RBF y vectores de características generados mediante Análisis de Componentes Principales (PCA), reduciendo la dimensionalidad de las imágenes a 256 componentes principales. Se realizó una búsqueda de hiperparámetros mediante validación cruzada para encontrar los valores óptimos de **C** y **gamma**, obteniéndose los siguientes resultados: **C = 10.0**, **gamma = scale**. El accuracy total alcanzado por la SVM fue de **0.617**.

Tabla II: Métricas por clase - Modelo 2

| Clase | Precision | Recall | F1-Score | Support |
|-----------------|-------------|-------------|-------------|-------------|
| Clase 0 | 0.71 | 0.60 | 0.65 | 100 |
| Clase 1 | 0.57 | 0.69 | 0.63 | 100 |
| Clase 2 | 0.55 | 0.36 | 0.44 | 100 |
| Clase 3 | 0.61 | 0.76 | 0.68 | 100 |
| Clase 4 | 0.52 | 0.83 | 0.64 | 100 |
| Clase 5 | 0.62 | 0.45 | 0.52 | 100 |
| Clase 6 | 0.82 | 0.53 | 0.64 | 100 |
| Clase 7 | 0.61 | 0.70 | 0.65 | 100 |
| Clase 8 | 0.64 | 0.78 | 0.71 | 100 |
| Clase 9 | 0.64 | 0.47 | 0.54 | 100 |
| Promedio | 0.63 | 0.62 | 0.61 | 1000 |

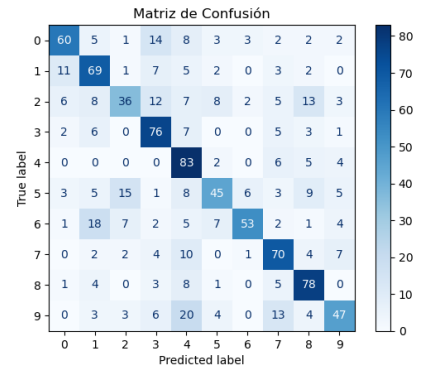


Figura 2: Matriz de confusión - Modelo 2

Muestra un rendimiento ligeramente inferior al Modelo 1, con un **accuracy de 0.617**. Aunque PCA reduce la dimensionalidad de manera eficiente, parece perder parte de la estructura no lineal que el kernel RBF podría aprovechar.

IV-C. Modelo 3: RBF + UMAP

Este modelo utiliza un kernel RBF y vectores de características generados mediante la técnica de reducción de dimensionalidad no lineal UMAP, transformando las imágenes a 256 dimensiones. Se realizó una búsqueda de hiperparámetros mediante validación cruzada para encontrar los valores óptimos de **C** y **gamma**, obteniéndose los siguientes resultados: **C = 10.0**, **gamma = scale**. El accuracy total alcanzado por la SVM fue de **0.312**.

Tabla III: Métricas por clase - Modelo 3

| Clase | Precision | Recall | F1-Score | Support |
|-----------------|-------------|-------------|-------------|-------------|
| Clase 0 | 0.50 | 0.01 | 0.02 | 100 |
| Clase 1 | 0.14 | 0.44 | 0.21 | 100 |
| Clase 2 | 0.00 | 0.00 | 0.00 | 100 |
| Clase 3 | 0.27 | 0.48 | 0.35 | 100 |
| Clase 4 | 0.47 | 0.09 | 0.15 | 100 |
| Clase 5 | 0.88 | 0.28 | 0.42 | 100 |
| Clase 6 | 0.23 | 0.32 | 0.27 | 100 |
| Clase 7 | 0.49 | 0.55 | 0.52 | 100 |
| Clase 8 | 0.78 | 0.58 | 0.61 | 100 |
| Clase 9 | 0.31 | 0.37 | 0.34 | 100 |
| Promedio | 0.39 | 0.31 | 0.29 | 1000 |

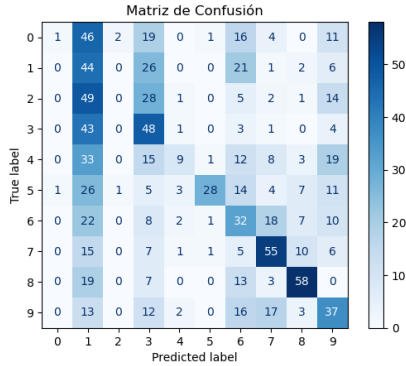


Figura 3: Matriz de confusión - Modelo 3

Este modelo presenta el peor rendimiento de los modelos RBF (**accuracy 0.312**). UMAP, aunque útil para visualización, parece introducir demasiada compresión o ruido en el espacio reducido, dificultando la separación clara de clases por parte del SVM.

IV-D. Modelo 4: Lineal + Vector unidimensional

Este modelo emplea un kernel lineal, utilizando imágenes aplanadas de tamaño 256×256 convertidas directamente en vectores unidimensionales. Se realizó una búsqueda de hiperparámetros mediante validación cruzada para encontrar el valor óptimo de **C**, obteniéndose el siguiente resultado: **C = 10.0**. El valor de **gamma** no aplica para el kernel lineal. El accuracy total alcanzado por la SVM fue de **0.592**.

Su desempeño (**accuracy 0.592**) es competitivo considerando que utiliza un kernel lineal. La separación entre clases más sencillas se mantiene aceptable, aunque la falta de capacidad para modelar relaciones no lineales limita su efectividad global.

Tabla IV: Métricas por clase - Modelo 4

| Clase | Precision | Recall | F1-Score | Support |
|-----------------|-------------|-------------|-------------|-------------|
| Clase 0 | 0.58 | 0.66 | 0.62 | 100 |
| Clase 1 | 0.65 | 0.60 | 0.62 | 100 |
| Clase 2 | 0.60 | 0.44 | 0.51 | 100 |
| Clase 3 | 0.69 | 0.70 | 0.70 | 100 |
| Clase 4 | 0.48 | 0.58 | 0.52 | 100 |
| Clase 5 | 0.63 | 0.55 | 0.59 | 100 |
| Clase 6 | 0.62 | 0.60 | 0.61 | 100 |
| Clase 7 | 0.60 | 0.60 | 0.60 | 100 |
| Clase 8 | 0.67 | 0.72 | 0.70 | 100 |
| Clase 9 | 0.52 | 0.48 | 0.50 | 100 |
| Promedio | 0.60 | 0.59 | 0.59 | 1000 |

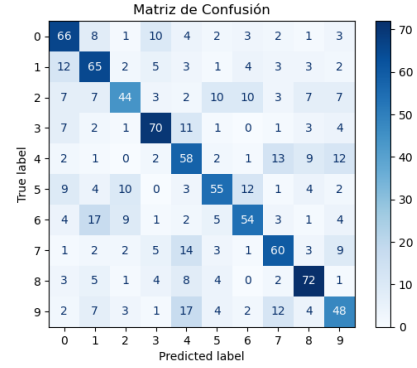


Figura 4: Matriz de confusión - Modelo 4

IV-E. Modelo 5: Lineal + PCA

Este modelo emplea un kernel lineal y vectores de características generados mediante Análisis de Componentes Principales (PCA), reduciendo las imágenes a 256 componentes. Se realizó una búsqueda de hiperparámetros mediante validación cruzada para encontrar el valor óptimo de **C**, obteniéndose el siguiente resultado: **C = 10.0**. El valor de **gamma** no aplica para el kernel lineal. El accuracy total alcanzado por la SVM fue de **0.595**.

Tabla V: Métricas por clase - Modelo 5

| Clase | Precision | Recall | F1-Score | Support |
|-----------------|-------------|-------------|-------------|-------------|
| Clase 0 | 0.61 | 0.64 | 0.62 | 100 |
| Clase 1 | 0.53 | 0.59 | 0.56 | 100 |
| Clase 2 | 0.64 | 0.59 | 0.61 | 100 |
| Clase 3 | 0.63 | 0.66 | 0.64 | 100 |
| Clase 4 | 0.46 | 0.64 | 0.54 | 100 |
| Clase 5 | 0.65 | 0.55 | 0.59 | 100 |
| Clase 6 | 0.71 | 0.60 | 0.65 | 100 |
| Clase 7 | 0.62 | 0.63 | 0.62 | 100 |
| Clase 8 | 0.70 | 0.67 | 0.68 | 100 |
| Clase 9 | 0.46 | 0.39 | 0.42 | 100 |
| Promedio | 0.60 | 0.59 | 0.60 | 1000 |

Muy similar en rendimiento al Modelo 4 (**accuracy 0.595**). PCA ayuda a eliminar ruido y reducir la dimensión, lo cual favorece ligeramente al clasificador lineal, pero no genera una mejora significativa.

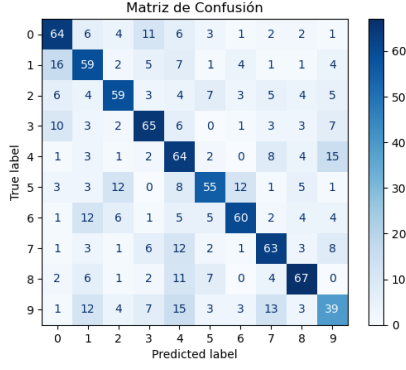


Figura 5: Matriz de confusión - Modelo 5

IV-F. Modelo 6: Lineal + UMAP

Este modelo emplea un kernel lineal y vectores de características generados mediante la técnica de reducción de dimensionalidad UMAP, transformando las imágenes a un espacio de 256 dimensiones. Se realizó una búsqueda de hiperparámetros mediante validación cruzada para encontrar el valor óptimo de C , obteniéndose el siguiente resultado: $C = 10.0$. El valor de γ no aplica para el kernel lineal. El accuracy total alcanzado por la SVM fue de **0.319**.

Tabla VI: Métricas por clase - Modelo 6

| Clase | Precision | Recall | F1-Score | Support |
|-----------------|-------------|-------------|-------------|-------------|
| Clase 0 | 0.56 | 0.05 | 0.09 | 100 |
| Clase 1 | 0.16 | 0.88 | 0.28 | 100 |
| Clase 2 | 0.16 | 0.04 | 0.06 | 100 |
| Clase 3 | 0.31 | 0.17 | 0.22 | 100 |
| Clase 4 | 0.33 | 0.07 | 0.12 | 100 |
| Clase 5 | 0.75 | 0.36 | 0.48 | 100 |
| Clase 6 | 0.30 | 0.20 | 0.24 | 100 |
| Clase 7 | 0.63 | 0.45 | 0.53 | 100 |
| Clase 8 | 0.41 | 0.53 | 0.46 | 100 |
| Clase 9 | 0.44 | 0.42 | 0.43 | 100 |
| Promedio | 0.44 | 0.32 | 0.31 | 1000 |

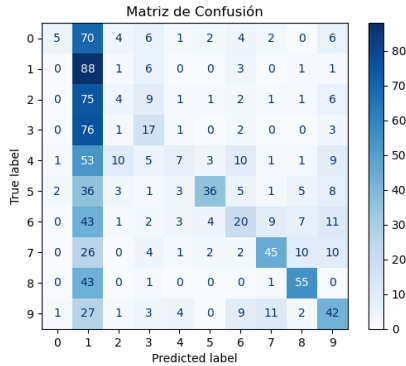


Figura 6: Matriz de confusión - Modelo 6

Al igual que con el Modelo 3, la combinación con UMAP produce resultados deficientes (**accuracy 0.319**). El espacio proyectado por UMAP no parece ser adecuado para una separación lineal clara entre clases.

Con el objetivo de visualizar comparativamente el desempeño de los modelos, se elaboró un gráfico que muestra el *accuracy* obtenido por cada configuración experimental, contrastando directamente cada modelo con **kernel RBF** frente a su contraparte equivalente con **kernel lineal**. De este modo, se puede observar el efecto que tiene el tipo de *kernel* sobre el rendimiento en cada técnica de generación de características: vector unidimensional, PCA y UMAP.

Tabla VII: Accuracy total por técnica de vector y tipo de kernel

| Técnica de vectores | Kernel RBF | Kernel Lineal |
|-----------------------|------------|---------------|
| Vector unidimensional | 0.67 | 0.59 |
| PCA | 0.62 | 0.60 |
| UMAP | 0.31 | 0.32 |

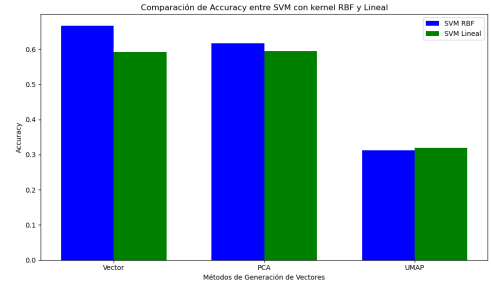


Figura 7: Comparación RBF VS Lineal

La diferencia entre los modelos es más notable en el caso del vector unidimensional, donde RBF supera al kernel lineal. Para PCA, ambos kernels se comportan de manera similar, aunque RBF mantiene una ligera ventaja. En contraste, en UMAP, los resultados son bajos para ambos kernels, pero el modelo lineal logra una leve mejora, indicando que UMAP no genera una representación óptima para separación no lineal en este contexto.

V. CONCLUSIONES

En conclusión, los experimentos realizados demostraron que el rendimiento de un modelo SVM está fuertemente influenciado tanto por la elección del *kernel* como por la técnica utilizada para representar los datos. En particular, el modelo que utilizó el **kernel RBF** junto a vectores unidimensionales mostró el mejor desempeño general, con un balance sólido entre *precision*, *recall* y *F1-score* en casi todas las clases. El uso de **PCA** ofreció resultados consistentes, especialmente en combinación con el **kernel lineal**, mientras que **UMAP** no logró generar representaciones efectivas para la clasificación supervisada en este contexto. Además, se evidenció la importancia de evaluar no solo el *accuracy* total, sino también el comportamiento por clase, ya que ciertos modelos mostraron un rendimiento desigual que podría comprometer su aplicabilidad práctica. La correcta selección de hiperparámetros, realizada mediante *GridSearch*, también fue clave para optimizar cada configuración y obtener resultados confiables utilizando un subconjunto reducido del conjunto de entrenamiento.