



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# SVMs

**Grau en Enginyeria Informàtica**  
**Aprendizaje automático**

Iñaki Díez Lambies

Andrea Gascó Pau

Grupo: L1-4CO21

**Contenido**

1. Introducción.....	3
2. Experimentos .....	3
2.1. Kernel gaussiano.....	4
2.2. Kernel sigmoide .....	5
2.3. Kernel polinómico.....	5
3. Conclusión.....	6

## 1. Introducción

En esta práctica se nos propone la tarea de generar diferentes modelos a través de la librería `sklearn`, basándonos en el kernel lineal para los kernels gaussiano, polinómico y sigmoide.

Para realizar esto, se nos ha provisto del conjunto de datos de MNIST, tanto sus datos como las etiquetas de estos.

El objetivo es conseguir un modelo que no supere el tamaño de 4Mbytes, con el error mínimo después de haber hecho el experimento con los diferentes kernels y haber escogido el que mejor resultado ofrecía.

## 2. Experimentos

Para realizar los experimentos con los diferentes kernels, hemos realizado unos scripts para cada uno de ellos, basándonos como hemos dicho anteriormente, en el kernel lineal.

Los parámetros que hemos ido variando en los diferentes experimentos han sido:

Parámetro de penalización (**C**), que determina la importancia que se le da a minimizar la función de error del modelo en comparación con la complejidad de este. Por ejemplo, si se aumenta el valor del parámetro de penalización, se le dará más importancia a la simplicidad del modelo, lo que puede llevar a un modelo menos preciso, pero más generalizado. Por el contrario, si se disminuye el valor del parámetro de penalización, se le dará más importancia a la precisión del modelo, lo que puede llevar a un modelo más complejo y preciso, pero menos generalizado. Es decir, se usa para controlar el trade-off entre precisión y generalización.

Gamma ( **$\gamma$** , en nuestros scripts **G**), que determina la influencia de cada punto de entrenamiento en la función de decisión. Por ejemplo, si se aumenta el valor de gamma, cada punto de entrenamiento tendrá una mayor influencia en la función de decisión, lo que puede llevar a un modelo más preciso, pero menos generalizado. Por el contrario, si se disminuye el valor de gamma, cada punto de entrenamiento tendrá una menor influencia en la función de decisión, lo que puede llevar a un modelo menos preciso, pero más generalizado.

Degree (**D**), que determina el grado del kernel polinomial utilizado en la función de decisión del modelo. Por ejemplo, Si se aumenta el valor de degree, se utilizará un kernel polinomial de mayor grado en la función de decisión, lo que puede llevar a un modelo más complejo y preciso, pero menos generalizado. Por el contrario, si se disminuye el valor de degree, se utilizará un kernel polinomial de menor grado en la función de decisión, lo que puede llevar a un modelo menos preciso, pero más generalizado.

Coefo (**R**), que determina el valor de la constante que se utiliza en el kernel polinomial y sigmoidal utilizado en la función de decisión. Por ejemplo, Si se aumenta el valor de

coefo, se utilizará un valor más alto de la constante en el kernel polinomial o sigmoidal, lo que puede llevar a un modelo más complejo y preciso, pero menos generalizado. Por el contrario, si se disminuye el valor de coefo, se utilizará un valor más bajo de la constante en el kernel polinomial o sigmoidal, lo que puede llevar a un modelo menos preciso, pero más generalizado.

## 2.1. Kernel gaussiano

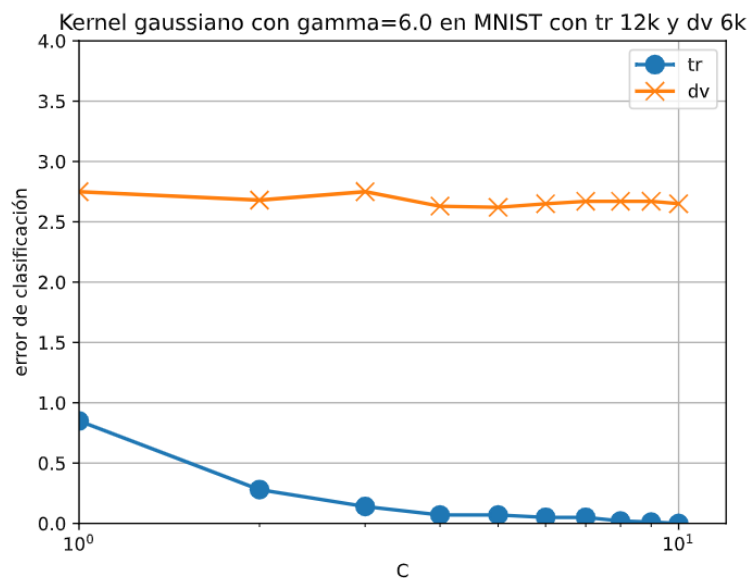
En este kernel, los principales valores a explorar son **C** y **G**.

Los valores que se han usado para el experimento son para C (0.01, 0.1, 1, 10, 100, 1000, 10000) y para G (1e-15, 1e-10, 1e-05, 1e-02, 1e-01, 1, 10, 100).

Hemos observado que para valores cercanos a 0 de G, se observa una mejoría conforme aumentamos su valor en todo el rango de C. A partir de G igual a 1, los resultados tanto en el conjunto de training como en el conjunto de test comienzan a ser mucho más positivos.

Como por definición los valores de C y G son contrarios el uno del otro, un equilibrio de valores entre ambos pueda dar un buen resultado. Tras observar que los valores cercanos a 10 tanto en C como en G resultan prometedores, hemos realizado otro experimento con los valores C (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12) y G (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12).

Tras este experimento, podemos observar que el modelo con el error más bajo y que cumple la restricción de 4Mbytes es el que tiene como parámetros C de 5 y una G de 6 con un error de 2.62.



## 2.2. Kernel sigmoid

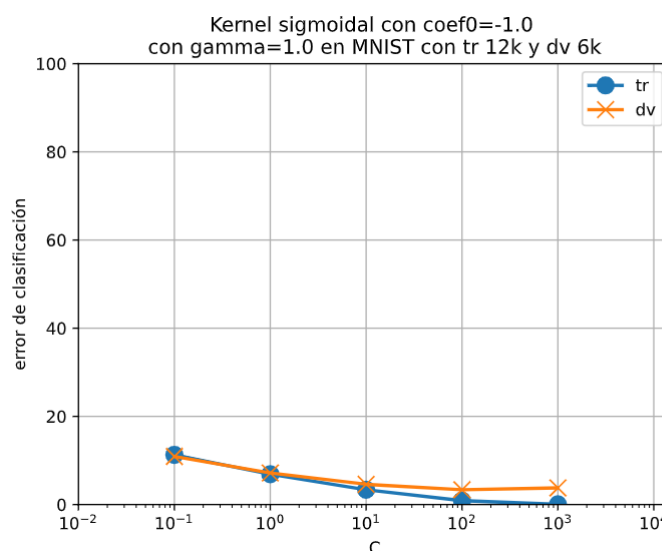
En este kernel, los principales valores a explorar son **C**, **G** y **R**.

Los valores que se han usado para el experimento son para C (0.01, 0.1, 1, 10, 100, 1000, 10000), para G (1e-01, 1, 10, 100) y para R (0, 1, 5, 10). Para valores elevados de R hemos observado que una G elevada, a partir de 10, empieza a dar peores resultados.

Hemos realizado un segundo experimento con R también negativas (-10, -5, -1, 0, 1, 5) y hemos obtenido buenos resultados en valores cercanos a -1.

A raíz de esto, hemos realizado otro experimento probando valores de R cercanos a -1, (-2, -1.5, -1, -0.5).

Tras este experimento, podemos observar que el modelo con el error más bajo y que cumple la restricción de 4MB es el que tiene como parámetros C de 100, una G de 1 y una R de -1 con un error de 3.35.



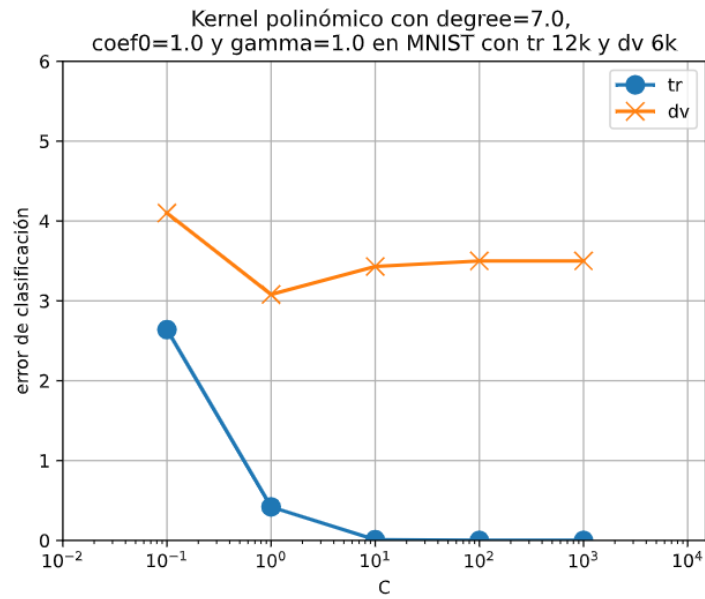
## 2.3. Kernel polinómico

En este kernel, los principales valores a explorar son **C**, **G**, **D** y **R**.

Los valores que se han usado para el experimento son para C (0.01, 0.1, 1, 10, 100, 1000), para G (1e-01, 1, 10, 100), para R (0, 1, 5) y D (0, 1, 2, 5). Este experimento ha resultado en conclusiones como que valores cercanos a 1 tanto de R como de G aportan una tasa de error más baja por norma general. Además, se observa una tendencia descendente del error ante mayores valores de D.

A partir de esto se ha realizado un segundo experimento con, además de los valores anteriormente probados de C, G y R; valores aún mayores de D (6, 7, 8, 9, 10) en búsqueda de un menor valor de error.

Así pues se ha encontrado un modelo compuesto por C de 1, R de 1, G de 1 y D de 7 que contiene el error más bajo en evaluación de 3.08 que, además, cumple la restricción.



### 3. Conclusión

A partir de todo lo anterior, podemos concluir con firmeza que el mejor kernel para nuestro problema se trata del Gaussiano con valor es de C igual a 5 y G igual a 6 con un error de 2.62 ante el conjunto de tests.