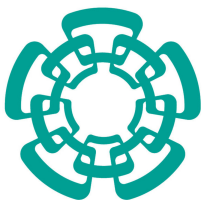


# LTI Cinvestav



## Máquinas de soporte vectorial con *kernels*

### Reconocimiento de patrones

Profesor: Dr. Wilfrido Gómez Flores

Estudiante: Rafael Pérez Torres

## 1. Introducción

Las máquinas de soporte vectorial son clasificadores basados en la idea de máximo margen entre dos líneas (hiperplanos) que separan a instancias de dos clases. Dichos hiperplanos son conocidos como vectores de soporte, que mientras mayor margen describan disminuirán el riesgo de que un patrón desconocido sea clasificado de forma incorrecta, alcanzando así la generalización del clasificador.

Las máquinas de soporte vectorial, como cualquier clasificador lineal, pueden intentar la clasificación de datos linealmente no separables, a través de una función kernel. Dicha función realiza una transformación de los datos en un espacio de baja dimensionalidad a otro de alta dimensionalidad en el que sea posible alcanzar esta separabilidad.

El presente documento muestra los resultados de la implementación de el clasificador de máquinas de soporte vectorial utilizando distintas funciones kernel con diferentes parámetros para las etapa de entrenamiento.

## 2. Marco teórico

Las máquinas de soporte vectorial atienden a un problema específico que otro tipo de clasificadores lineales, como el perceptrón y sus variantes dejan de lado: dado un conjunto

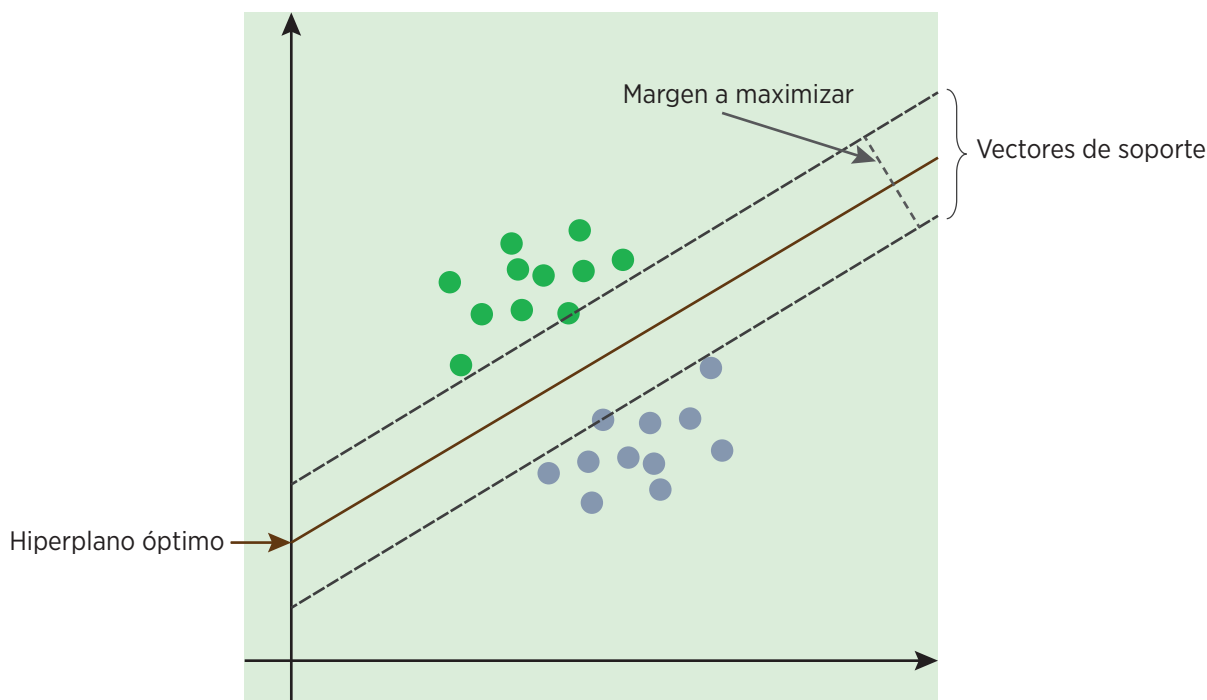


Figura 1: Identificación de vectores de soporte e hiperplano óptimo.

de clases (linealmente o no separables) existe más de un hiperplano que puede definir la separación entre las mismas; sin embargo, no todos los hiperplanos ofrecen la misma calidad o margen de separación entre las clases.

En ese sentido, la idea general en las máquinas de soporte vectorial es identificar aquel hiperplano que ofrece el mayor margen de separación entre las instancias de las diferentes clases.

Para identificar el mejor hiperplano, este clasificador busca aquellos datos (vectores) que se encuentren más hacia el margen *interno* de las clases. Si se consideran dos clases, se tendrían dos vectores delimitando a cada una de las dos clases. Dichos vectores son llamados vectores de soporte, ya que en ellos la máquina se basa para obtener el hiperplano que esté justo a la mitad, y que a la postre será el hiperplano que ofrezca la mayor separabilidad entre las clases, como se muestra en la Figura 1.

### 3. Metodología

La actividad consistió en realizar el conjunto de pasos descrito en el Algoritmo 1 para realizar el entrenamiento y la clasificación utilizando las máquinas de soporte vectorial.

Se realizó la división de cada dataset en 70 % para entrenamiento y 30 % para prue-

---

**Algoritmo 1** Algoritmo para la clasificación con SVM

---

- 1: Crear matriz Hessiana  $H = Ytr_i, Ytr_j, K(Xtr_i, Xtr_j)$
  - 2: Encontrar  $\alpha$ , tal que  $\max \left[ \sum_{i=1}^N \alpha_i - \frac{1}{2} \alpha^T H \alpha \right]$  sujeto a  $0 \leq \alpha_i \leq C \forall i$  y  $\sum_{i=1}^N \alpha_i y_i = 0$
  - 3: Determinar los vectores de soporte  $\{x_s, y_s\}$  que satisfacen la condición  $0 < \alpha_i < C$ .
  - 4: Calcular el bias:  $w_0 = \frac{1}{N_s} \sum_{s \in S} \{y_s - \sum_{m \in S} \alpha_m y_m K(x_m, x_s)\}$
  - 5: Dado un patrón  $x'$ , se clasificará según:  $y' = \text{sign} \{K(x_s, x')^T \alpha_s y_s + w_0\}$
- 

ba. Asimismo, se preparó la secuencia de parámetros mostrada en la Tabla 1 para la implementación de las funciones kernel lineal ( $K(x_i, x_j) = x_i^T x_j$ ), polinomial ( $K(x_i, x_j) = (x_i^T x_j)^2$ ) y *rbf* ( $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|_2^2)$ ) para cada uno de los cuatro datasets considerados en la asignación: *BreastMG*, *BreastUS*, *Diabetes* y *Heart*.

Función kernel	Parámetro	Valores
Lineal	C	1, 10, 100, 1000
	Orden	2, 5, 10, 100
Polinomial	C	1, 10, 100, 1000
	Gamma	0.01, 0.1, 0.5, 1.0
RBF	C	1, 10, 100, 1000

Tabla 1: Configuración de parámetros para la ejecución de las SVM

Respecto a la utilización de estos parámetros, se realizaron 31 experimentos con cada combinación de los parámetros. Por ejemplo, para la función kernel polinomial se ejecutaron 31 repeticiones con la tupla (orden, C) = (2,1), 31 repeticiones con la tupla (2,10), etcétera.

## 4. Resultados obtenidos

La ejecución del algoritmo para clasificación mediante SVM se realizó, como ha sido mencionado, 31 veces por cada configuración de parámetros para la función kernel sobre los cuatro datasets proporcionados. A continuación se describen los resultados obtenidos por cada tipo de función kernel aplicada.

La Tabla 2 muestra los porcentajes de error obtenidos por la SVM utilizando una función kernel lineal. Puede observarse que el parámetro  $C = 10$  obtuvo los porcentajes de

error más bajos para todos los datasets, a excepción del *BreastMG*. En general, a través de esta función se obtuvieron porcentajes de error no mayores al 25 %.

La Tabla 3 muestra los tiempos de ejecución del kernel lineal por las 31 ejecuciones para cada uno de los datasets. No puede inferirse información relevante, salvo que el dataset de *Diabetes* es el que mayor tiempo toma para su procesamiento (contiene 768 muestras de 8 dimensiones), mientras que el dataset *Heart* es el más rápido en procesarse (contiene 270 muestras de 13 dimensiones).

Dataset / Parameter (C)	BreastMG	BreastUS	Diabetes	Heart
1	2.3719	10.396	24.165	17.483
10	3.3397	9.3794	24.109	17.324
100	7.4194	10.396	24.727	18.041
1000	8.425	15.954	25.512	19.833

Tabla 2: Porcentajes de error del kernel lineal.

Dataset/ Parameter(C)	BreastMG	BreastUS	Diabetes	Heart
1	119.79	139.26	215.88	24.785
10	121.53	145.6	219.14	25.098
100	117.92	155.8	222.78	24.979
1000	118.78	156.31	211.97	23.287

Tabla 3: Tiempos de ejecución (segundos) del kernel lineal.

En adelante, por cuestiones de espacio, solamente se muestran las gráficas de los resultados para las funciones kernel polinomial y *rbf*. La Figura 2 muestra los porcentajes de errores obtenidos por la función kernel polinomial con distintas configuraciones. De dicha figura puede concluirse que para estos datasets, la *SVM* con una función kernel polinomial con dimensiones bajas tiende a obtener porcentajes de error menores. En general, sus porcentajes de error son no mayores a 64.77 %. De la Figura 3, que muestra los tiempos de ejecución para esta misma función kernel, puede observarse que la duración de la ejecución es casi duplicada a comparación de la función kernel lineal.

En la Figura 4 se muestran los porcentajes de error de la función kernel *rbf* con las distintas configuraciones implementadas. Puede notarse que ofrece los mejores resultados para el dataset *BreastMG*. En general, sus porcentajes de error son no mayores a 33.28 %. Asimismo, la Figura 5 muestra los tiempos de ejecución para este tipo de función. Puede observarse que la clasificación a través de esta función kernel es la que mayor tiempo requiere.

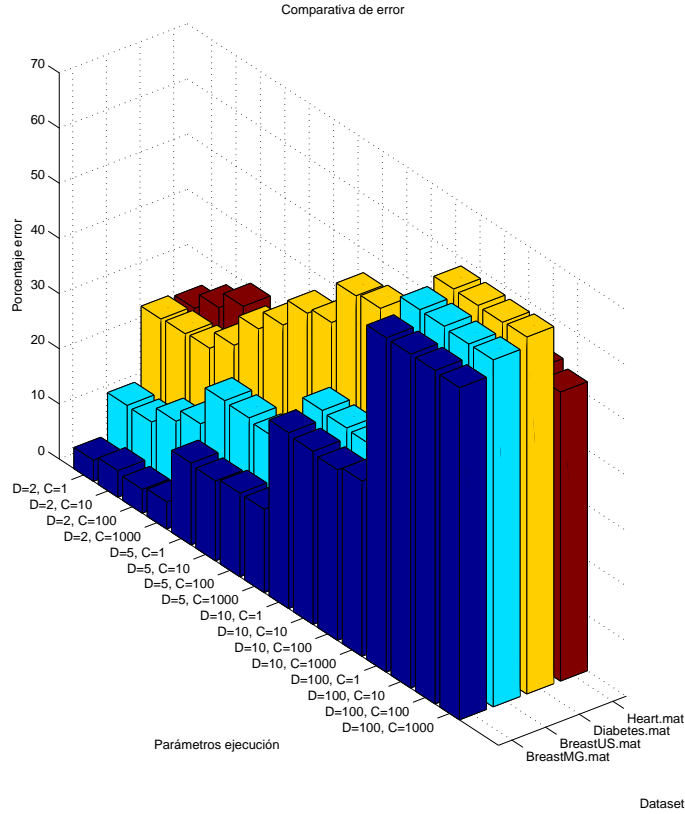


Figura 2: Porcentajes de error obtenidos por la función kernel polinomial.

La ejecución de todo el experimento alcanzó los 30'458 segundos  $\approx$  8 horas. Atendiendo a los mejores resultados de la ejecución de cada función kernel, tal como se muestra en la Tabla 4, la función kernel *rbf* obtiene los porcentajes de error más bajos.

	BreastMG		BreastUS		Diabetes		Heart	
	% error	Parámetros	% error	Parámetros	% error	Parámetros	% error	Parámetros
Lineal	2.3719	C=1	9.3794	C=10	24.109	C=10	17.324	C=10
Polinomial	3.814	O=2, C=1	11.29	O=2, C=10	24.755	O=2, C=1	23.337	O=5, C=100
rbf	2.2011	$\gamma=0.1$ , C=1	9.993	$\gamma=0.01$ , C=1	23.394	$\gamma=0.5$ , C=1	15.93	$\gamma=0.01$ , C=10

Tabla 4: Mejores resultados por dataset.

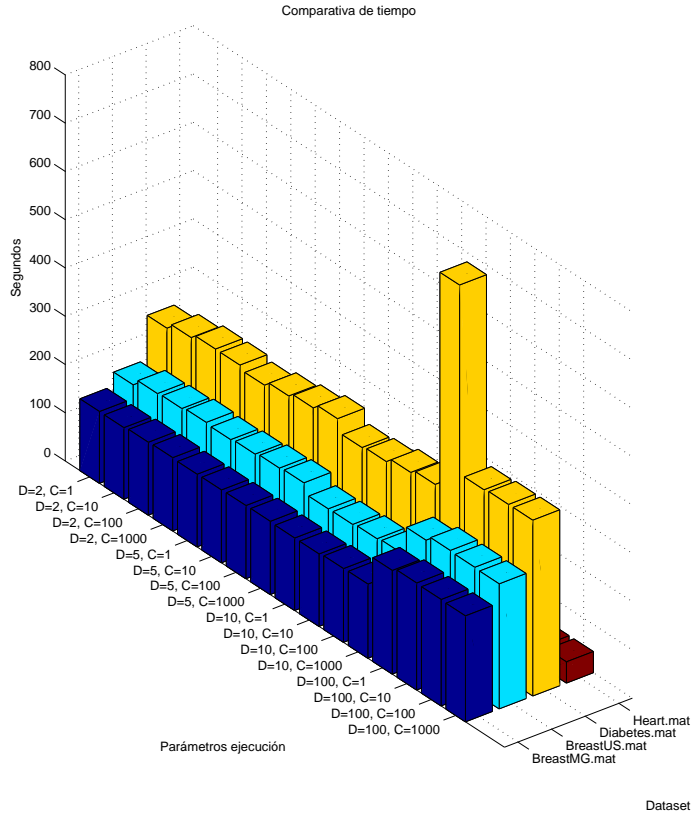


Figura 3: Tiempos de ejecución obtenidos por la función kernel polinomial.

## 5. Conclusiones

El presente documento ha presentado los resultados obtenidos de la implementación de un clasificador *SVM* utilizando las funciones kernel lineal, polinomial y *rbf* utilizando diferentes valores para su cálculo. Gracias al uso del *truco* del kernel, los clasificadores lineales obtienen la posibilidad de trabajar con datos que en un inicio no son linealmente separables, pero que al ver aumentada su dimensionalidad podrían evidenciar algún grado de separación.

Tomando como base los datasets proporcionados y atendiendo a los mejores resultados en cada dataset por cada una de las funciones kernel implementadas, la función kernel *rbf* obtiene los mejores resultados, con la excepción del caso del dataset *BreastUS* en el que se ve superado por la función kernel lineal.

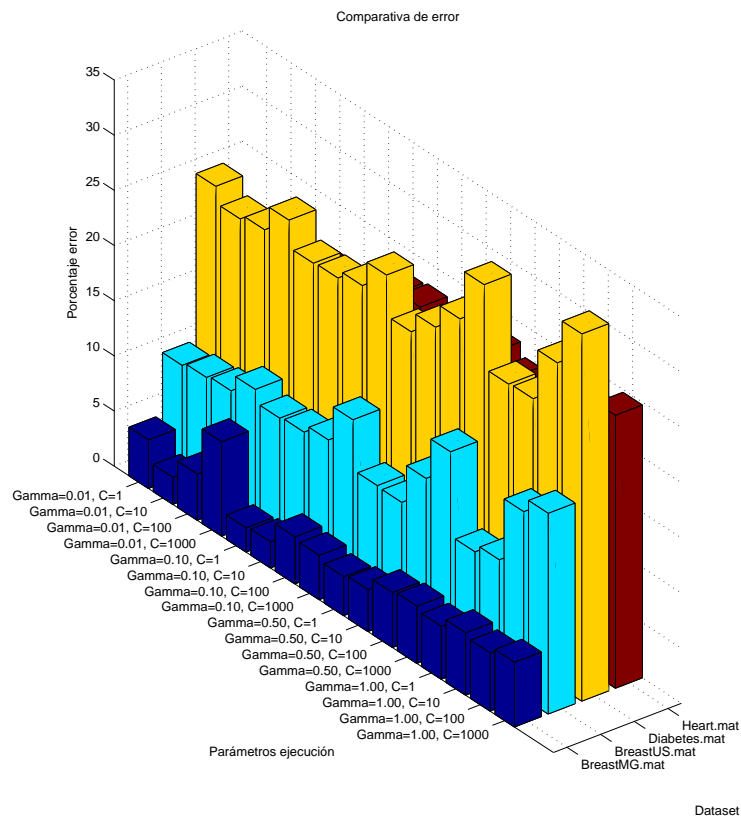


Figura 4: Porcentajes de error obtenidos por la función kernel *rbf*.

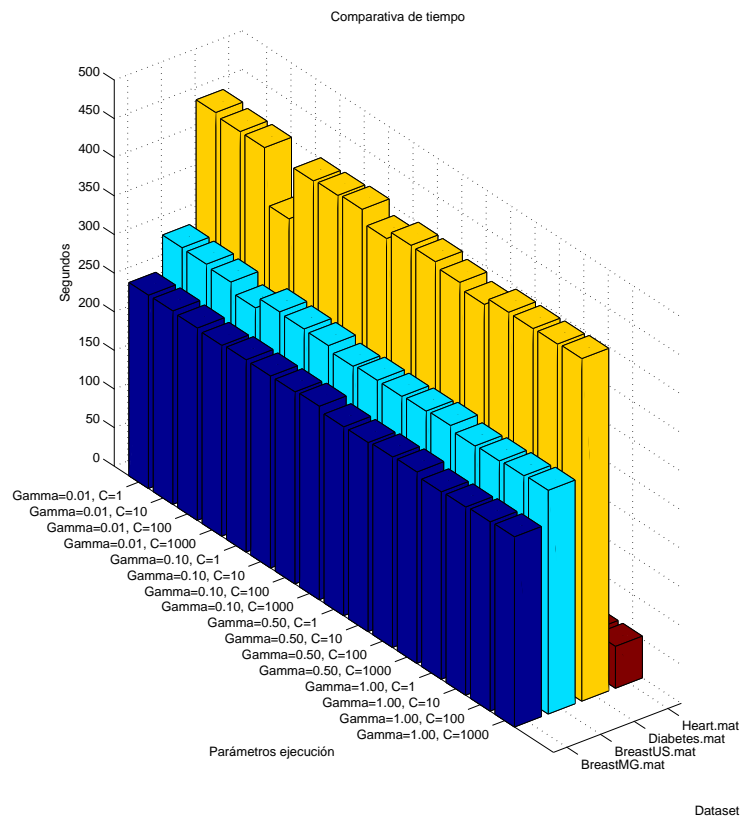


Figura 5: Tiempos de ejecución obtenidos por la función kernel *rbf*.