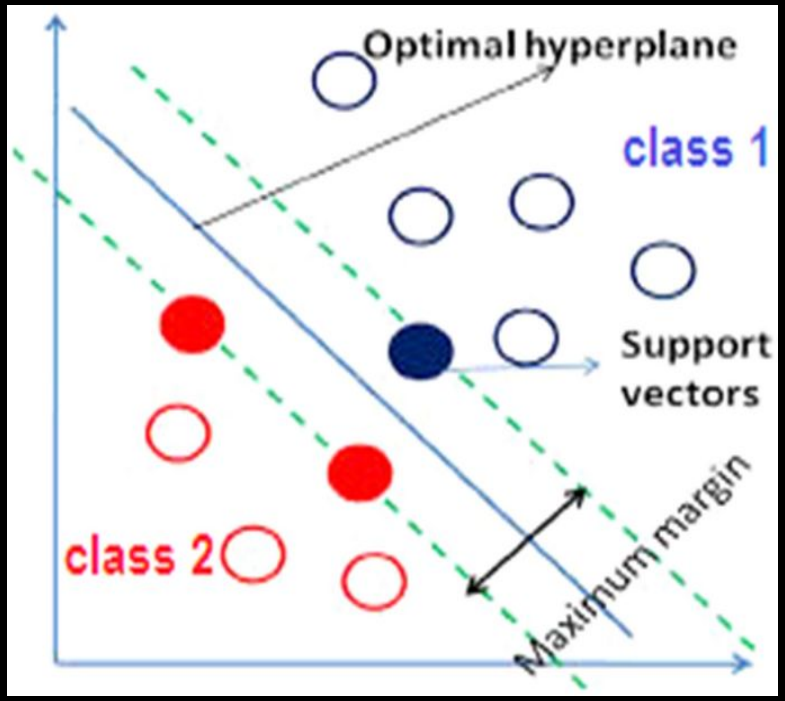
PUCP

INGENIERÍA INFORMÁTICA

Aprendizaje Automático



Unidad 6: SVM - Máquinas de Vectores de Soporte

Prof. César A. Beltrán Castañón

INTRODUCCIÓN
CLASIFICACIÓN SVM LINEAL

Máquinas de vectores de soporte (SVM)

2012 - Vladimir N. Vapnik

Vladimir N. Vapnik's pioneering work became the foundation of a new research field known as "statistical learning theory" that has transformed how computers learn in tackling complex problems. Working with Alexey Chervonenkis in Moscow during the late 1960s/early 1970s, Dr. Vapnik developed the Vapnik-Chervonenkis (VC) learning theory. This theory established a fundamental quantity to represent the limitations of learning machines. Dr. Vapnik later created principles to handle the generalization factors defined by VC theory, known as structural risk minimization. Dr. Vapnik's research was unknown to the Western world until his arriving in the United States shortly before the collapse of the Soviet Union. Working with AT&T Laboratories in Holmdel, NJ, during the 1990s, he put his theories into practical use with support vector machine (SVM) algorithms for recognizing complex patterns in data for classification and regression analysis tasks. SVMs have become the method of choice for machine learning.

A member of the U.S. National Academy of Engineering and NEC Laboratories America Fellow, Dr. Vapnik is currently a professor with Columbia University, New York, NY, USA.

Figura: Vladimir Vapnik desarrolló los conceptos base de las SVM en los 60's, y sus aplicaciones en los 90's luego de emigrar a los Estados Unidos. En 2012 recibe la medalla IEEE Frank Rosenblatt. Todavía entonces las SVM eran el método más reconocido para el aprendizaje automático.

Máquinas de vectores de soporte (SVM)

- Un conjunto de datos linealmente separable admite un número infinito de fronteras de decisión.
- Para un algoritmo que sólo busque minimizar el error de clasificación, cualquiera de dichas fronteras de decisión será igualmente óptima.
- Sin embargo, no cualquier solución dará lugar al mismo error de generalización.

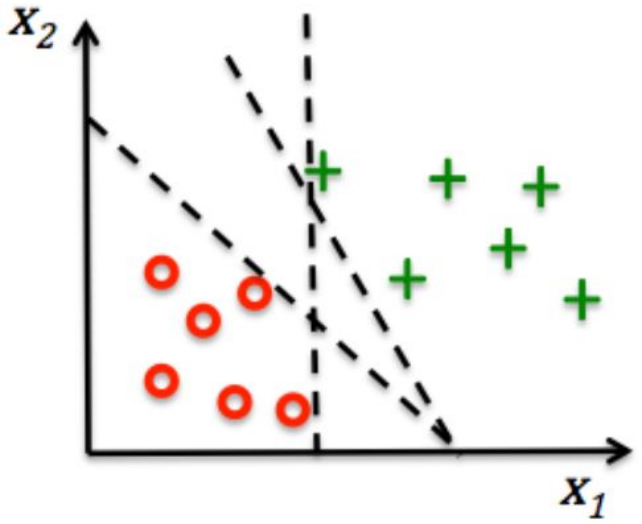


Figura: ¿Qué frontera de decisión es mejor? (Raschka 2015)

Máquinas de vectores de soporte (SVM): intuición

- Una máquina de vectores de soporte (SVM) es un clasificador de decisión (no probabilístico) que busca maximizar el margen o distancia entre la frontera de decisión y las instancias de entrenamiento más cercanas.

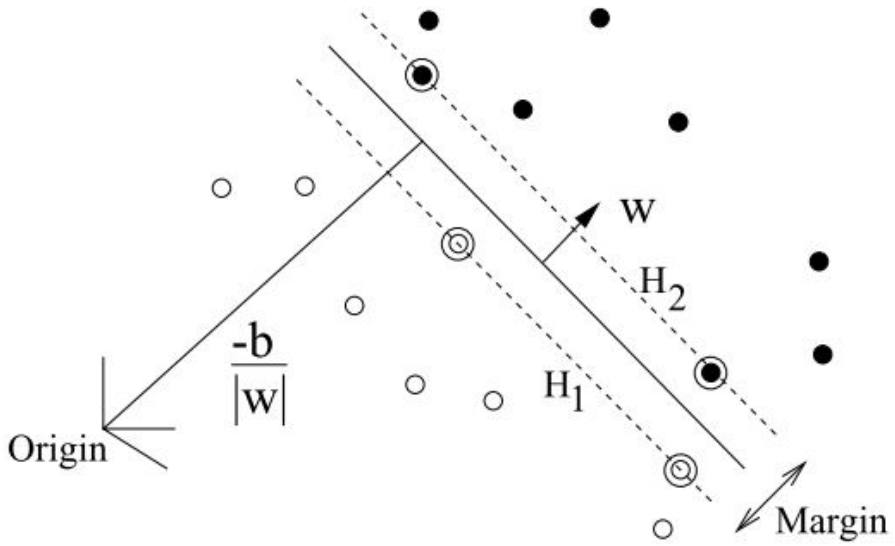


Figura: Frontera de decisión e hiperplanos SVM para un caso linealmente separable (Burges 1998)

Clasificación SVM Lineal: Vectores de soporte

- Una vez identificados, los vectores de soporte definen por completo la frontera de decisión.
- Añadir más instancias de entrenamiento “fuera de la avenida” no afecta por lo tanto la frontera de decisión.
- Ello disminuye la ocurrencia de “*overfitting*” en las SVM.

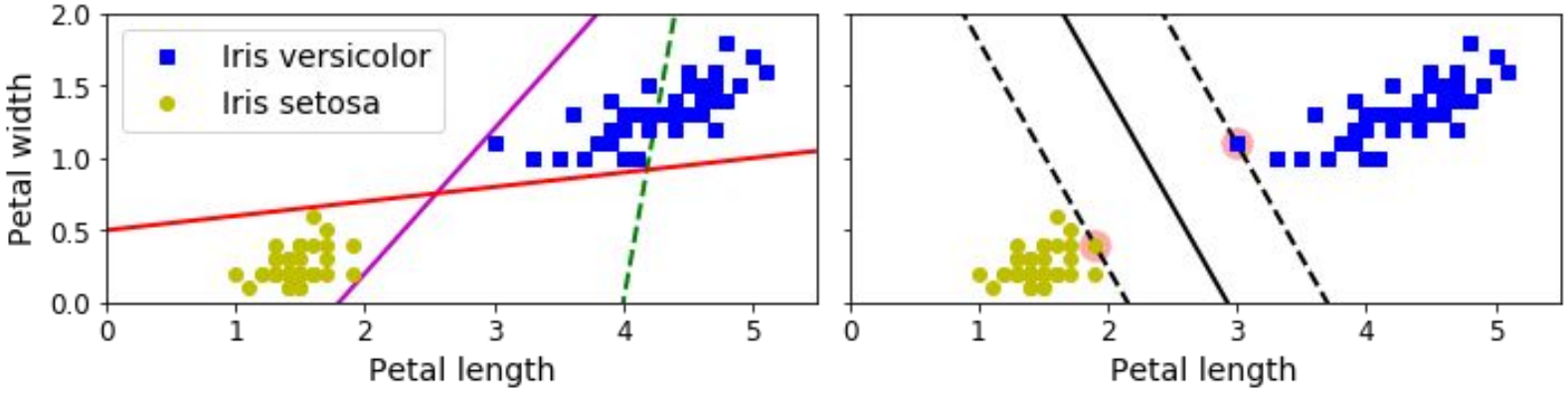


Figura: Conjunto de datos Iris. Las fronteras de decisión son completamente determinadas (“soportadas”) por los vectores de soporte (círculos rojos) (Geron 2019)

Clasificación SVM Lineal: Sensibilidad a la Magnitud

- Las SVM son muy sensibles a la magnitud o escala de las características.
- Es importante estandarizar primero las características.

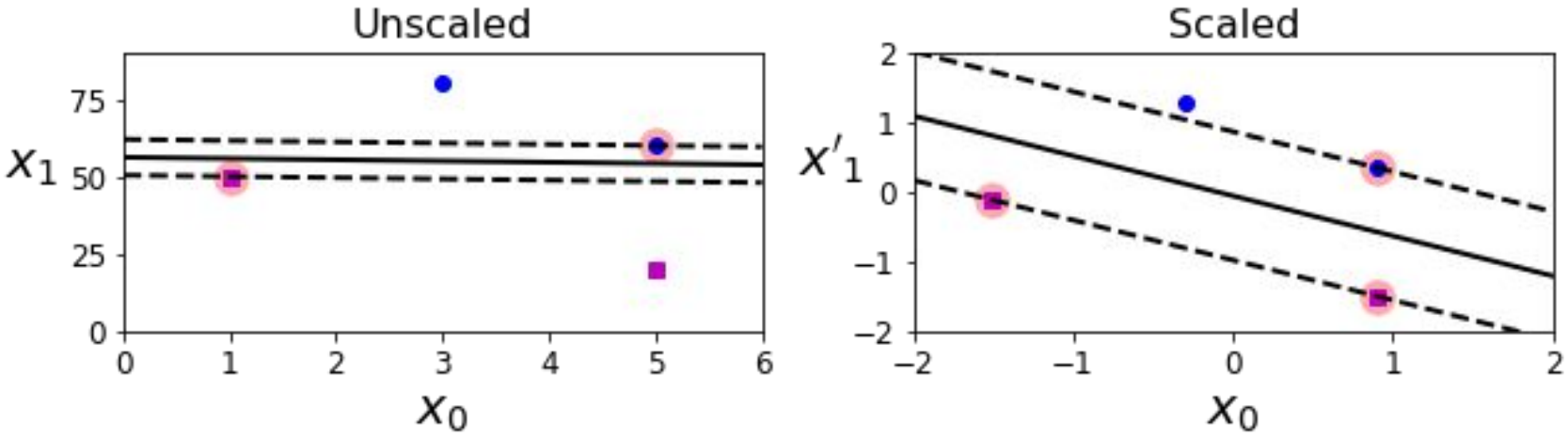


Figura: Izquierda: la magnitud vertical es mucho mayor que la horizontal, por ello la "avenida" más ancha es casi horizontal. Derecha: después de la estandarización, la frontera de decisión se ve mucho mejor. (Geron 2019)

Clasificación SVM Lineal: Problemas con el margen duro

- La clasificación de “margen duro” es la que exige que todas las instancias de entrenamiento estén fuera de la “avenida” y en la región correcta.
- Tiene dos principales problemas:
  1. Sólo funciona si los datos son linealmente separables.
  2. Es muy sensible a valores atípicos.

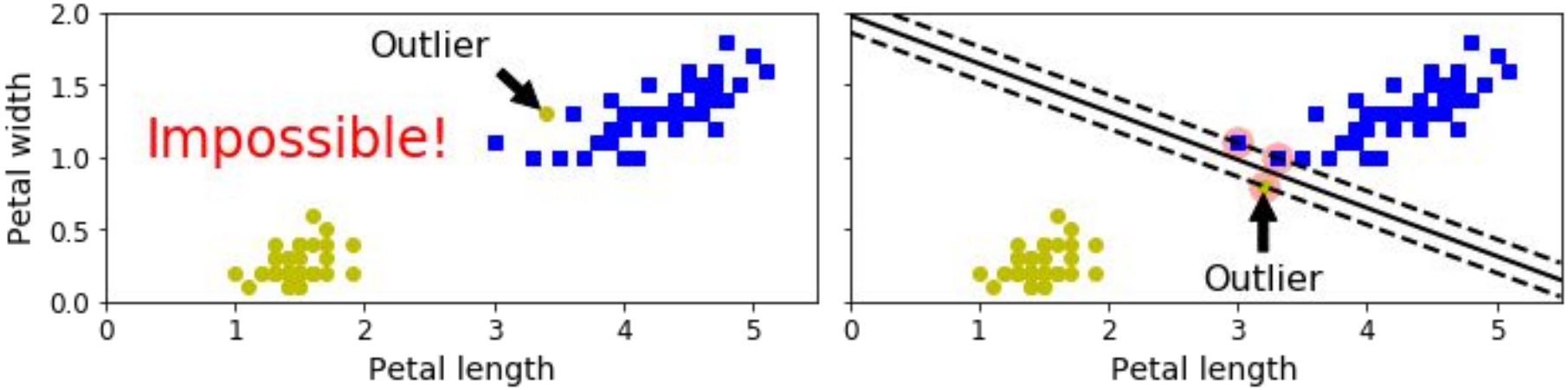


Figura: Conjunto de datos Iris con un valor atípico añadido. Izquierda: Será imposible encontrar un margen duro. Derecha: La frontera de decisión hallada seguramente no generalizará muy bien. (Geron 2019)

Clasificación SVM Lineal: Margen suave

- Una solución más flexible es la que se denomina clasificación de “margen suave”
  - La intuición es lograr un equilibro entre la maximización del margen y la minimización de las “violaciones del margen” (instancias fuera de la región correcta).
  - El mayor o menor peso atribuido a las violaciones del margen se controla mediante un hiperparámetro C.
  - Las violaciones al margen pasan a ser también vectores de soporte.

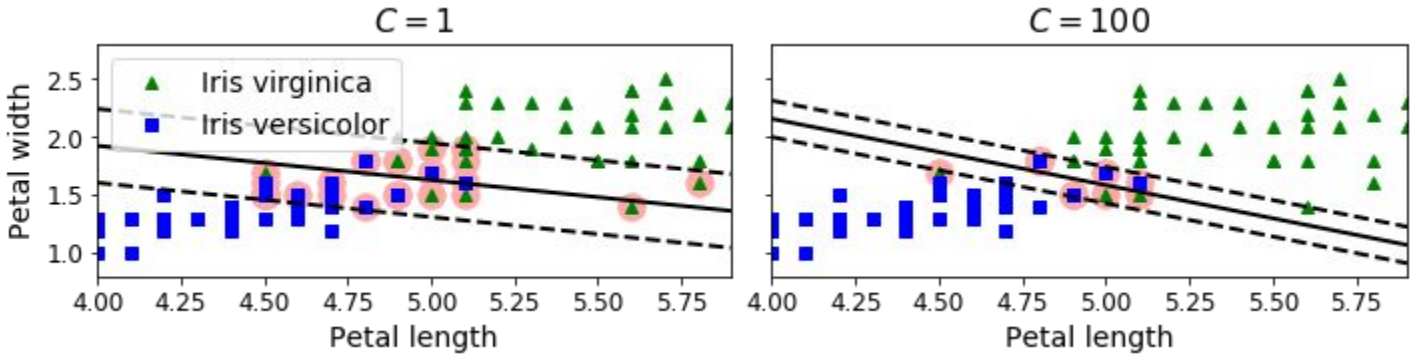
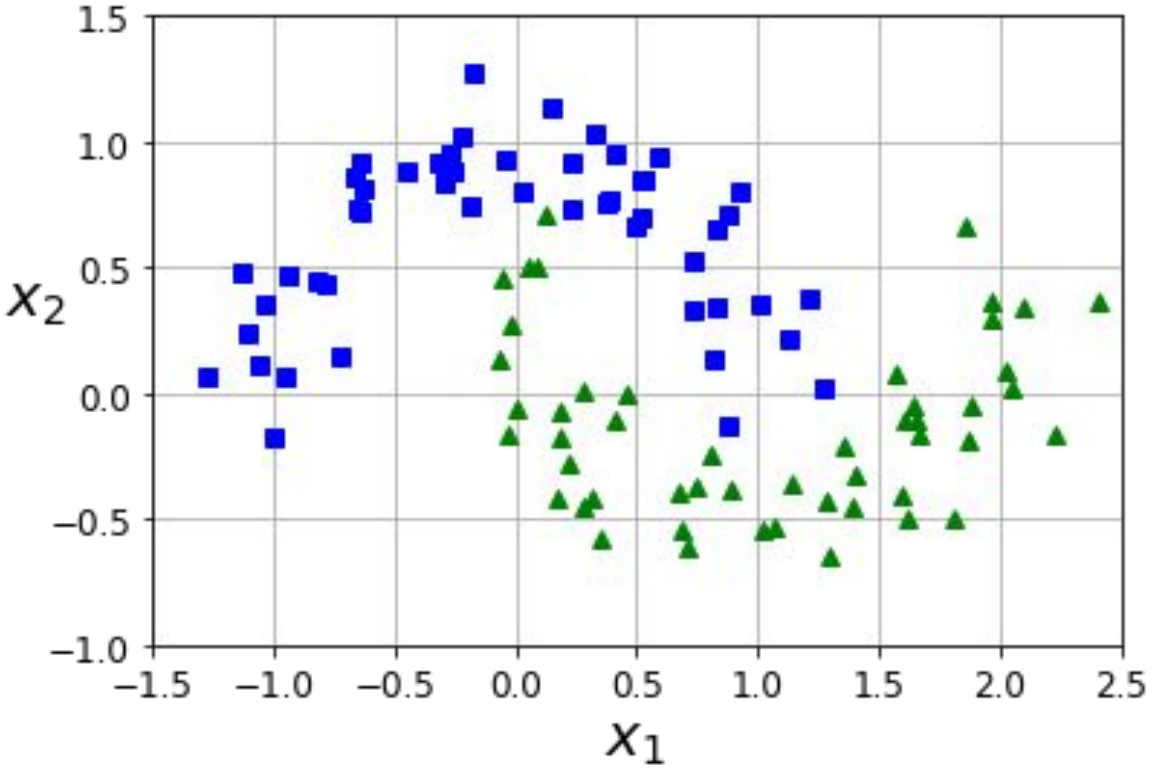


Figura: Izquierda: Un valor bajo de *C* prioriza un margen ancho. Derecha: Un valor alto de *C* minimiza las violaciones al margen e incrementa la posibilidad de “*overfitting*”. (Geron 2019)

CLASIFICACIÓN SVM NO LINEAL

Clasificación SVM No Lineal: Introducción

- La clasificación lineal funciona muy bien en muchos casos, pero muchos conjuntos de datos están muy lejos de ser linealmente separables.



Clasificación SVM No Lineal: Intuición

- En esos casos, la solución está en añadir más características (p.ej. polinómicas) que transporten los datos a un nuevo espacio que sí sea linealmente separable.

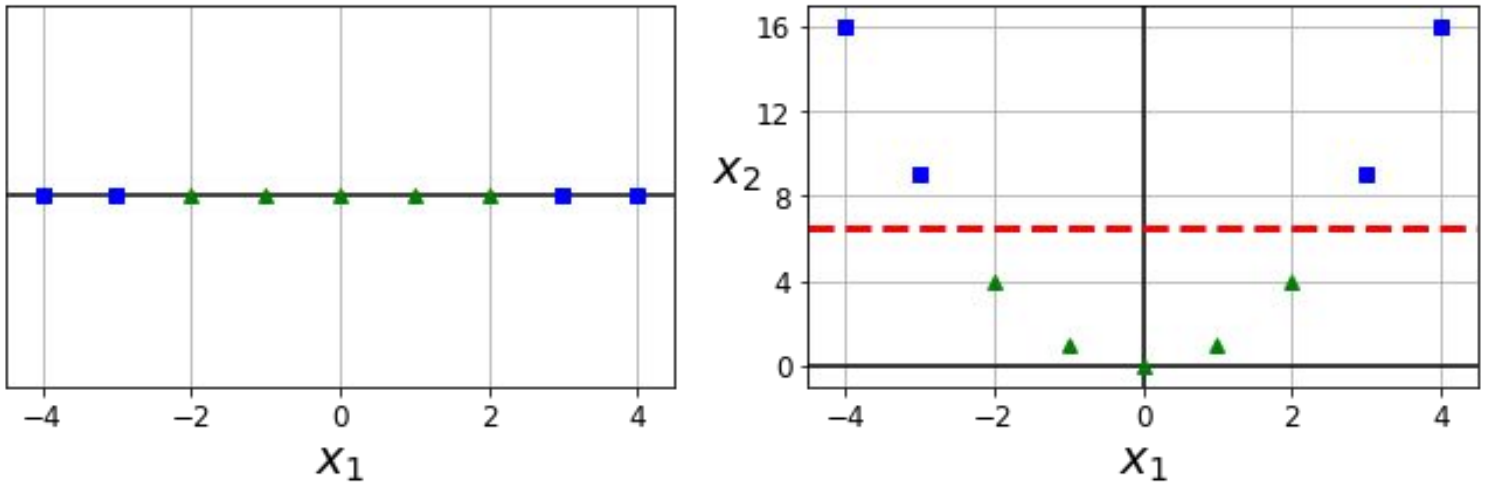


Figura: Añadir nuevas características puede hacer que un conjunto se vuelva linealmente separable. $x_2 = x_1^2$ (Geron 2019)

Clasificación SVM No Lineal: Intuición

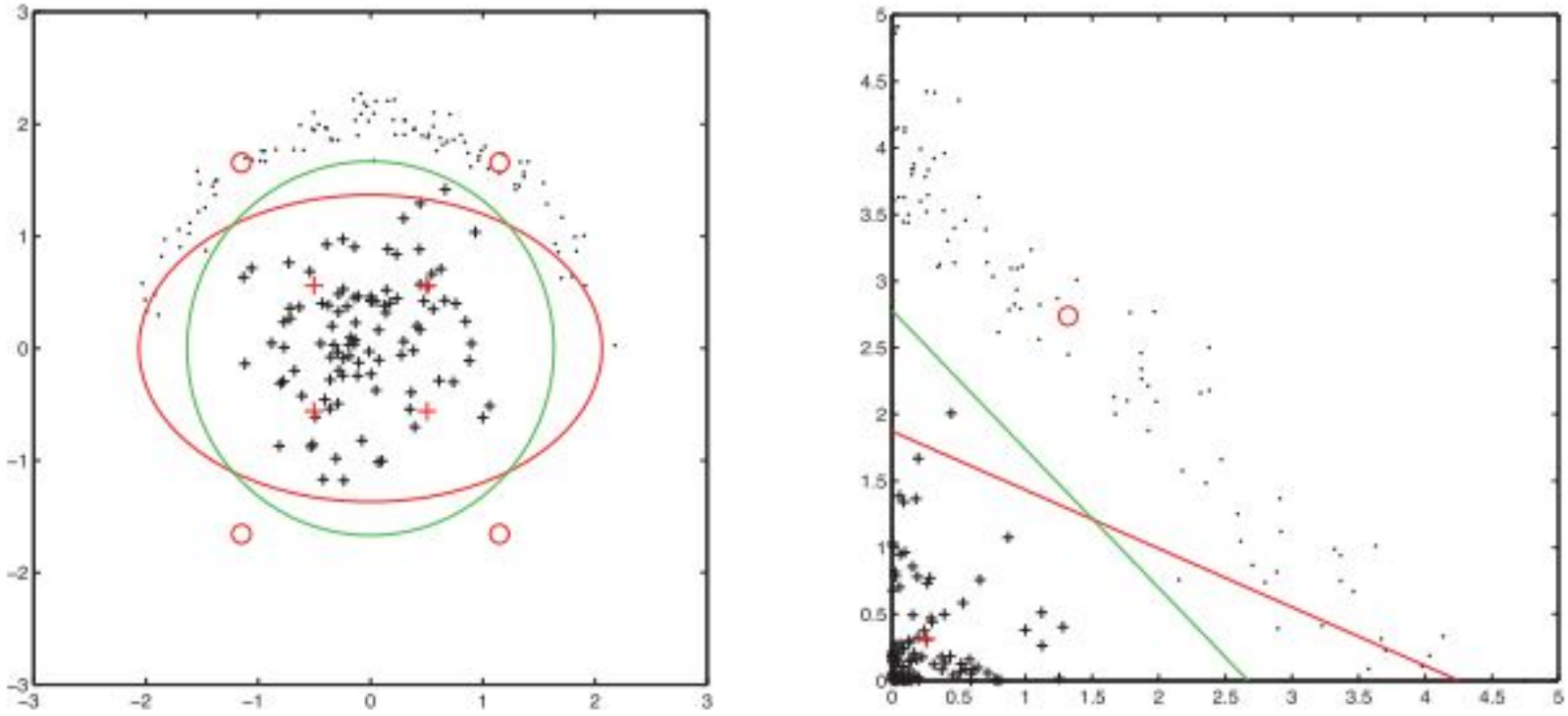


Figura: Añadir nuevas características puede hacer que un conjunto se vuelva linealmente separable. $x`_1 = x_1^2$ ; $x`_2 = x_2^2$ (Flach 2012)

Clasificación SVM No Lineal: Kernels

- Transportar las instancias a un nuevo espacio vectorial equivale a aplicar una función de transformación ϕ.

  - En el último ejemplo: $\boldsymbol{x} = \{x_1, x_2\} \rightarrow \phi(\boldsymbol{x}) = \{x_1^2, x_2^2\}$
  - Una expansión polinomial suele añadir más términos:
    $\boldsymbol{x} = \{x_1, x_2\} \rightarrow \phi(\boldsymbol{x}) = \{x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3\}$

- Añadir un número potencialmente grande de características se convierte muy pronto en un problema: insuficiente memoria, lentitud en los cálculos, etc.

Clasificación SVM No Lineal: Kernel

- Felizmente las SVMs pueden valerse de una elegancia matemática denominada *kernel trick*.

  - Es posible porque el entrenamiento de una SVM sólo depende del producto punto $\mathbf{x}_i \cdot \mathbf{x}_j$, por pares, de las instancias (cuyo resultado es un valor escalar).
  - Por lo tanto, no se hace necesario calcular explícitamente las nuevas características $\phi(\mathbf{x})$ si se puede definir una **función kernel** $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$.
  - La clasificación lineal corresponde a un **kernel lineal**: $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$

Clasificación SVM No Lineal: Kernel polinomial

- Un **kernel polinomial de grado** d se define:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\gamma \boldsymbol{x}_i \cdot \boldsymbol{x}_j + r)^d, \gamma > 0$$

- Un kernel polinomial requiere los siguientes hiperparámetros.

C	Costo de las violaciones al margen. A mayor valor, mayor ajuste a los datos y menos amplitud del margen. Disminuir C sirve como regularizador contra el «overfitting».
d	El grado del polinomio. Un grado mayor implica mayor complejidad y mayor probabilidad de «overfitting».
r (`coef0`)	Coeficiente que controla cuánto más peso se da a los polinomios de grado alto versus los de grado bajo.
γ (`gamma`)	Coeficiente del kernel. Los paquetes SVM suelen dar la opción de ajustarlo automáticamente a partir del número de características.

Clasificación SVM No Lineal: Kernel polinomial

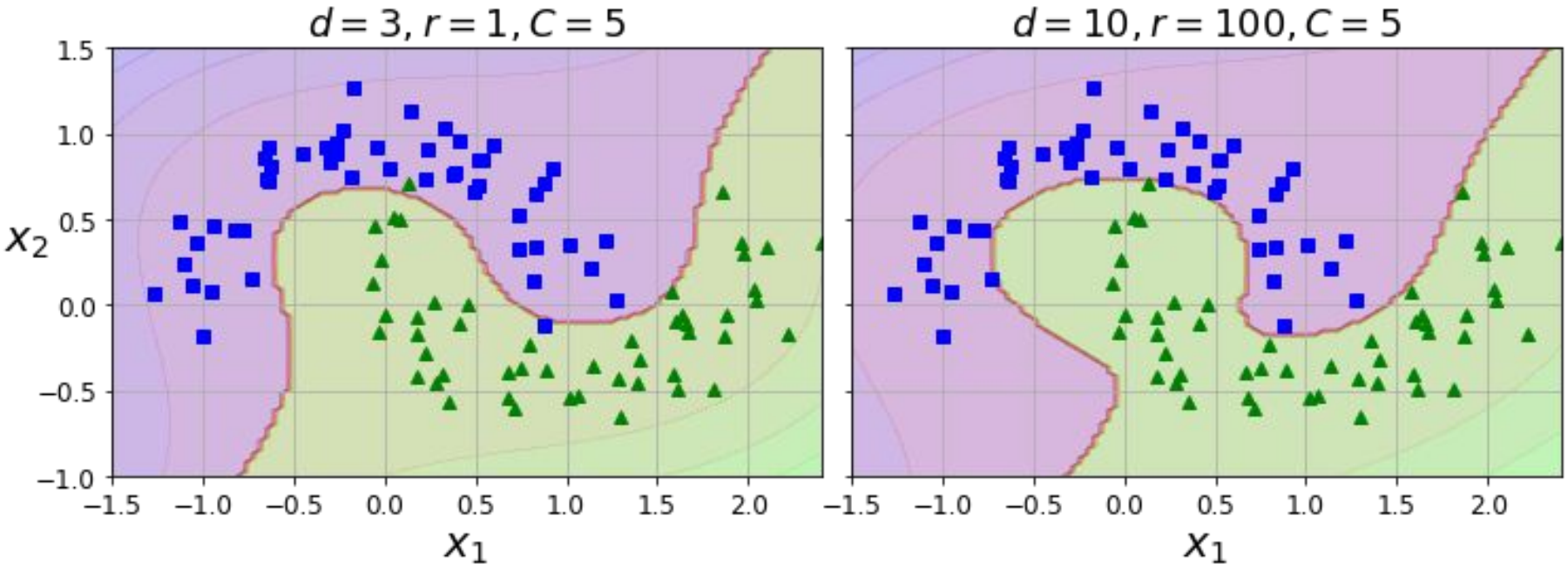


Figura: Izquierda: Clasificación con kernel polinomial de grado 3. Derecha: Usar un kernel de grado 10 en este caso muestra señales de sobreajuste. (Geron 2019)

Clasificación SVM No Lineal: Kernel gausiano

- Un **kernel gausiano** (Radial Basis Function - RBF) se define:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(\frac{-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2\sigma^2})$$

  Normalmente se escribe de modo simplificado:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||^2), \gamma = \frac{1}{2\sigma^2}$$

- Un kernel gausiano requiere los siguientes hiperparámetros:

  C Costo de las violaciones al margen. A mayor valor, mayor ajuste a los datos y menos amplitud del margen. Disminuir C sirve como regularizador contra el «overfitting».

  γ (`gamma`) Define cuán lejos llega la influencia de cada instancia individual. Es inverso a la varianza de la curva normal (gausiana). Disminuir γ sirve como regularizador contra el «overfitting».

Clasificación SVM No Lineal: Kernel gausiano

- Un kernel gausiano establece una *medida de semejanza* entre un par de ejemplos.

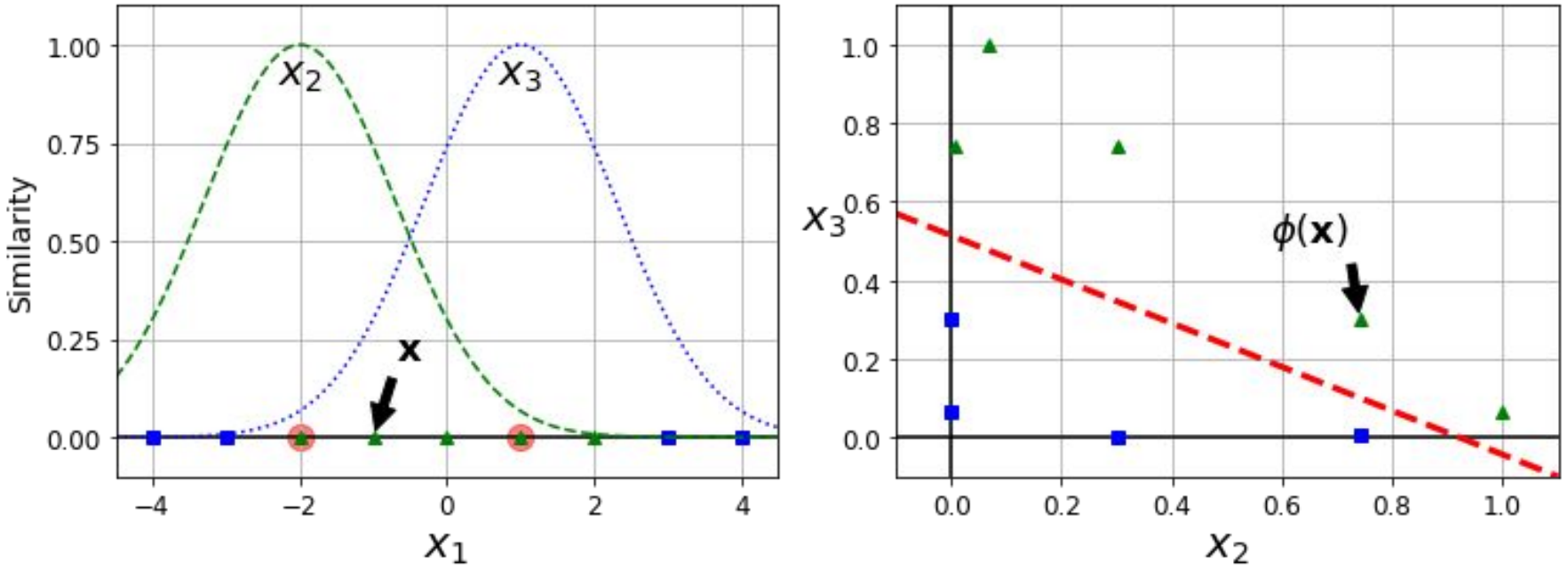


Figura: Nuevas características creadas con una medida de semejanza gausiana. (Geron 2019)

Clasificación SVM No Lineal: Kernel gausiano

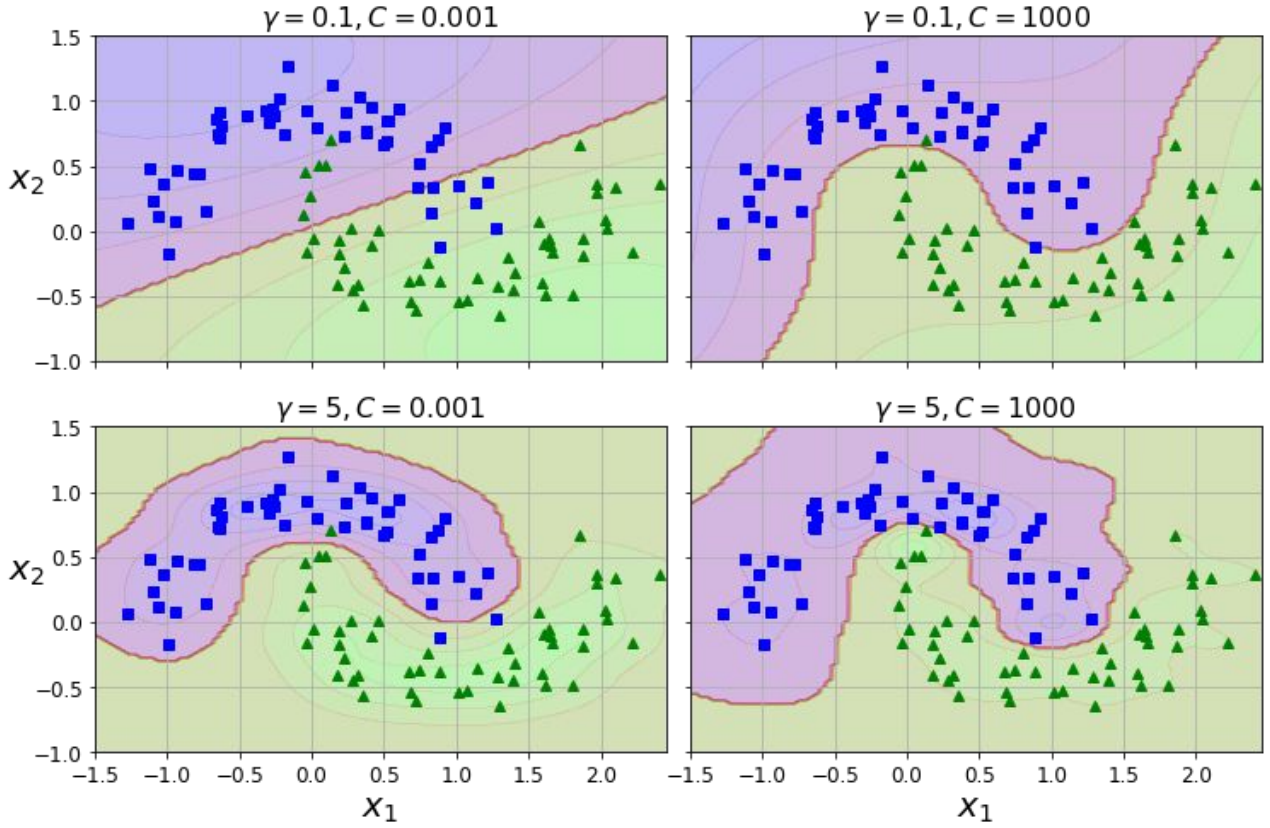


Figura: Clasificación SVM con kernel gausiano, usando diferentes valores de γ (gamma) y C. Valores altos de γ definen fronteras de decisión irregulares, pegadas a las instancias individuales. Valores bajos definen fronteras de decisión más generalizables. (Geron 2019)

RECOMENDACIONES PRÁCTICAS

BÚSQUEDA EN GRILLA

Recomendaciones prácticas

Hsu (2003) propone la siguiente metodología práctica para experimentaciones iniciales:

- Normalizar los datos al rango [-1, +1] o [0, 1]. (Ojo que se debe transformar siempre usando los parámetros de normalización aprendidos con el conjunto de entrenamiento)
- Probar SVM con el kernel gausiano (RBF).
- Usar validación cruzada para encontrar los mejores valores de C y γ.
- Entrenar todo el conjunto de entrenamiento (incluido validación) con los valores hallados de C y γ.
- Evaluar en el conjunto de pruebas.

Búsqueda en grilla (Grid search) con validación cruzada

	C = 0.001	C = 0.01	...	C = 10
gamma=0.001	SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	...	SVC(C=10, gamma=0.001)
gamma=0.01	SVC(C=0.001, gamma=0.01)	SVC(C=0.01, gamma=0.01)	...	SVC(C=10, gamma=0.01)
...	...	...	...	...
gamma=100	SVC(C=0.001, gamma=100)	SVC(C=0.01, gamma=100)	...	SVC(C=10, gamma=100)

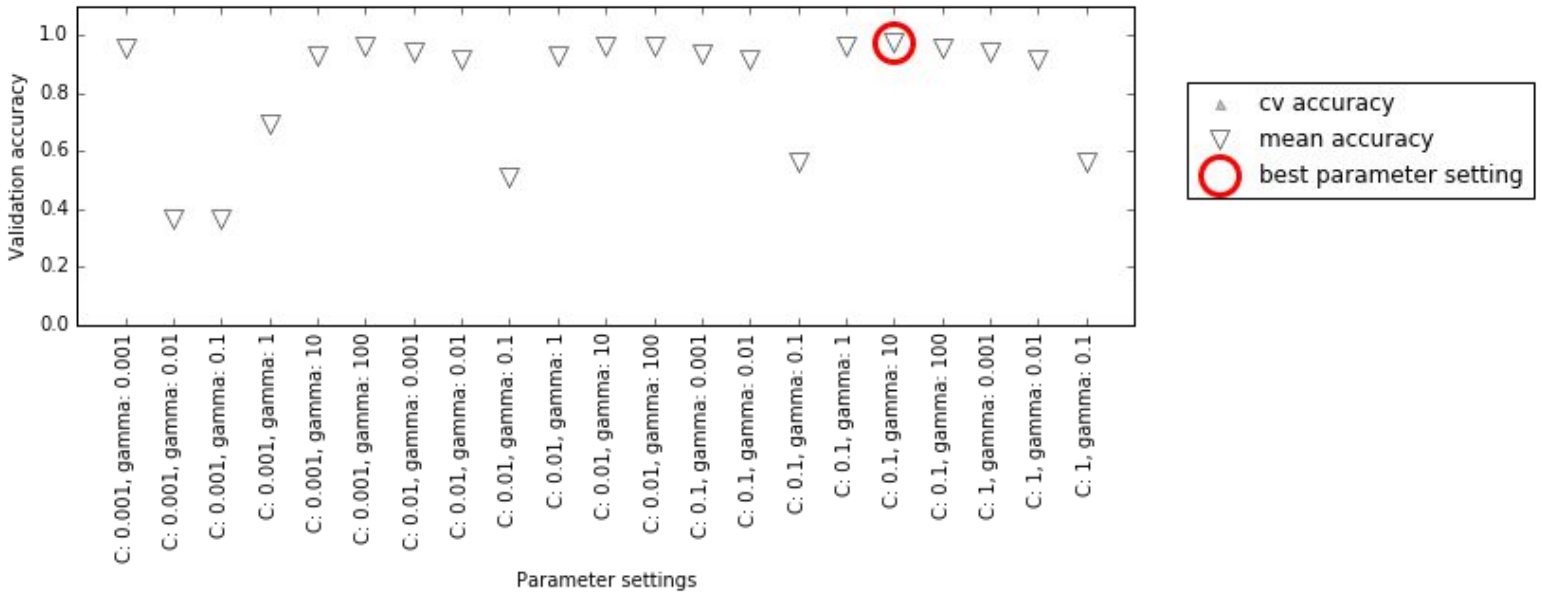


Figura: Resultados de búsqueda en grilla con validación cruzada (Mueller 2016)

Búsqueda en grilla (Grid search) con validación cruzada

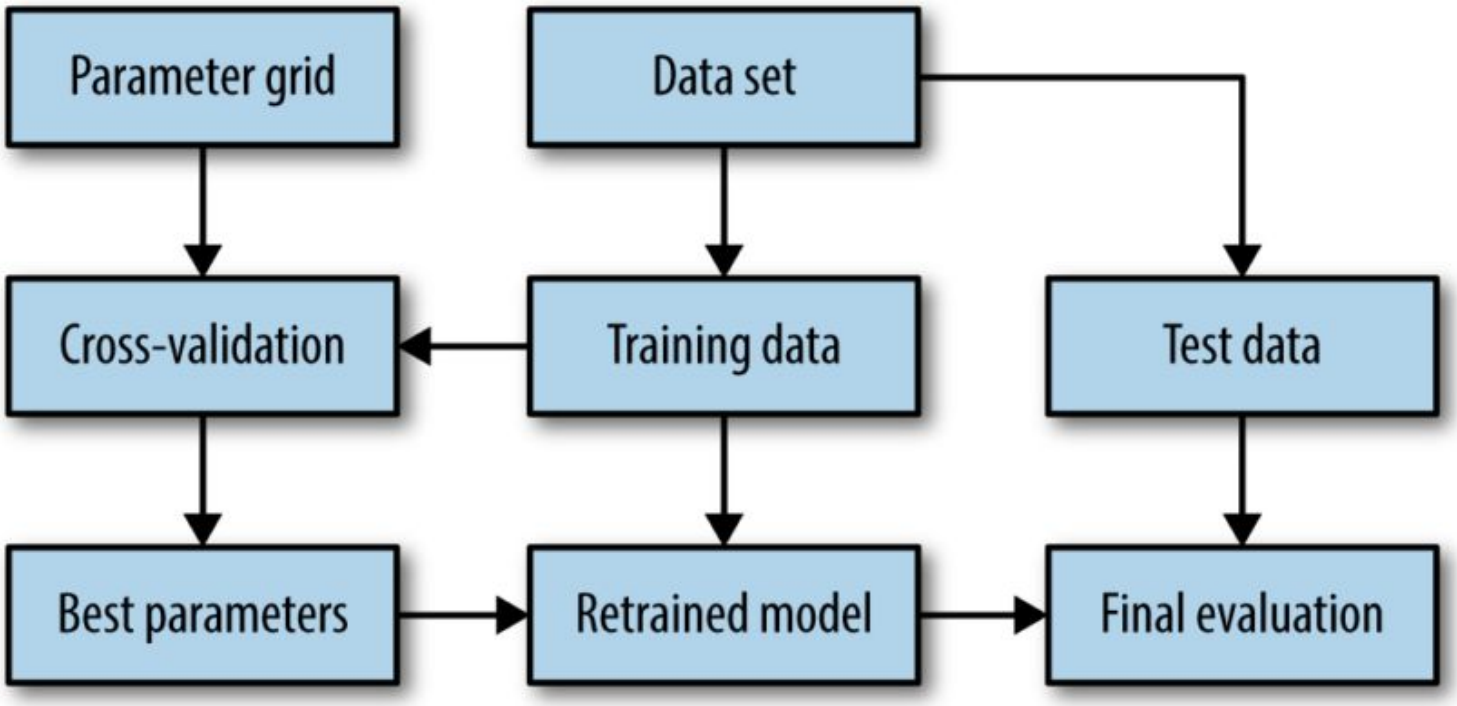


Figura: Proceso de selección de parámetros y evaluación de modelo usando búsqueda en grilla con validación cruzada (Mueller 2016)

REGRESIÓN SVM

Regresión SVM

- SVM sirve para clasificación, regresión y detección de anomalías (SVM de 1-clase).
- Puede también ser lineal o no lineal.
- Intuición: se invierte el objetivo.
  - Clasificación: busca maximizar las instancias que están fuera de la «avenida» y en la región correcta.
  - Regresión: Busca maximizar las instancias *dentro* de la «avenida» y reducir el número de las que queden fuera. El ancho del margen lo define el hiperparametro ϵ.

Regresión SVM lineal

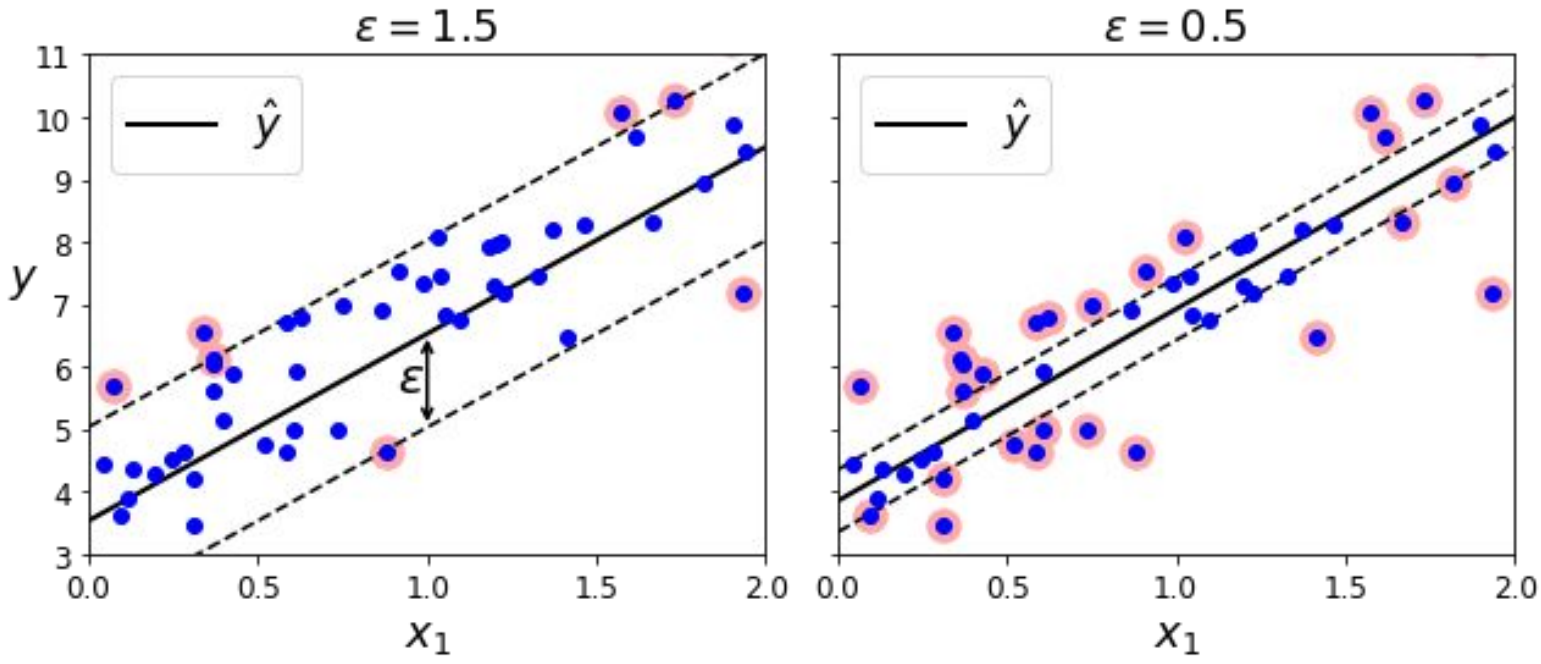


Figura: Modelos de regresión SVM lineales entrenados en datos lineales aleatorios. Izquierda: margen ancho ($\epsilon = 1{,}5$). Derecha: margen delgado ($\epsilon = 0{,}5$). (Geron 2019)

Regresión SVM no lineal

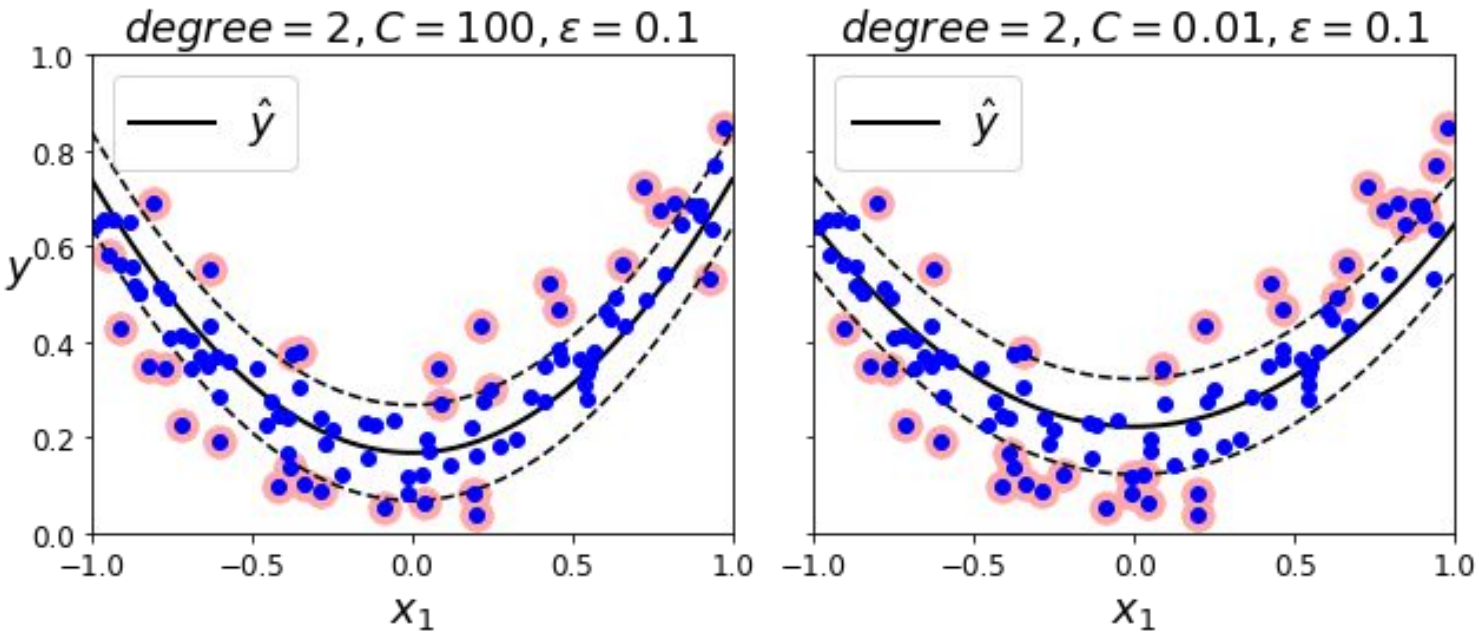


Figura: Modelos de regresión SVM no lineales, con kernel polinomial de segundo grado. Izquierda: poca regularización ($C \models 100$). Derecha: bastante regularización ($C = 0{,}01$). (Geron 2019)

DETECCIÓN DE VALORES ATÍPICOS
SVM 1-CLASS

SVM de una sola clase: Detección de valores atípico

- SVM de una sola clase (One-Class SVM) es un algoritmo *no supervisado* que aprende la distribución de los datos con lo que ha sido entrenado, y luego clasifica nuevos datos como semejantes o diferentes a los datos de entrenamiento.
- Se puede tratar de emplear cuando se tiene clases muy desbalanceadas.
  - Se entrena sólo con los datos de la clase mayoritaria.
  - Se debe excluir características en las que ambas clases tengan una distribución muy superpuesta.
- Además de C y demás hiperparámetros propios del kernel que se emplee, se requiere el siguiente hiperparámetro:

  ν (`nu`) Proporción de valores atípicos que se esperaría observar, p.ej. $0{,}01 (1\,\%)$

- Ver Outlier Detection with One-Class SVMs

SVM de una sola clase: Detección de valores atípico

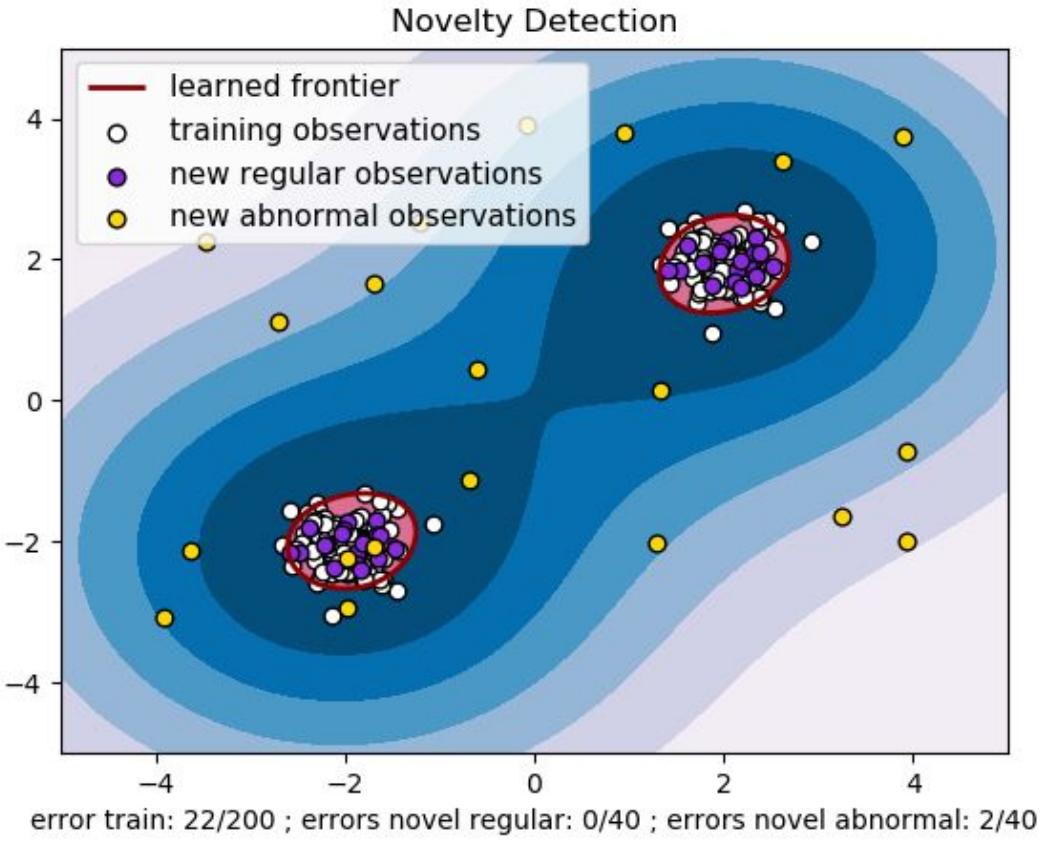


Figura: SVM de una sola clase para detección de valores atípicos. Fuente: Scikit learn

Bibliografía

- Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2), 121–167.
- Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.
- Hsu, C. W., Chang, C. C., & Lin, C. J. (2016). A practical guide to support vector classification. Initial version: 2003. Last updated: May 19, 2016. https://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf
- Mueller, A.C. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media.
- Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. Springer, 8(6), 187.