dcc
CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

Curso DM

Clasificación

Algoritmos (KNN, Naive Bayes, SVM)

Primavera 2023

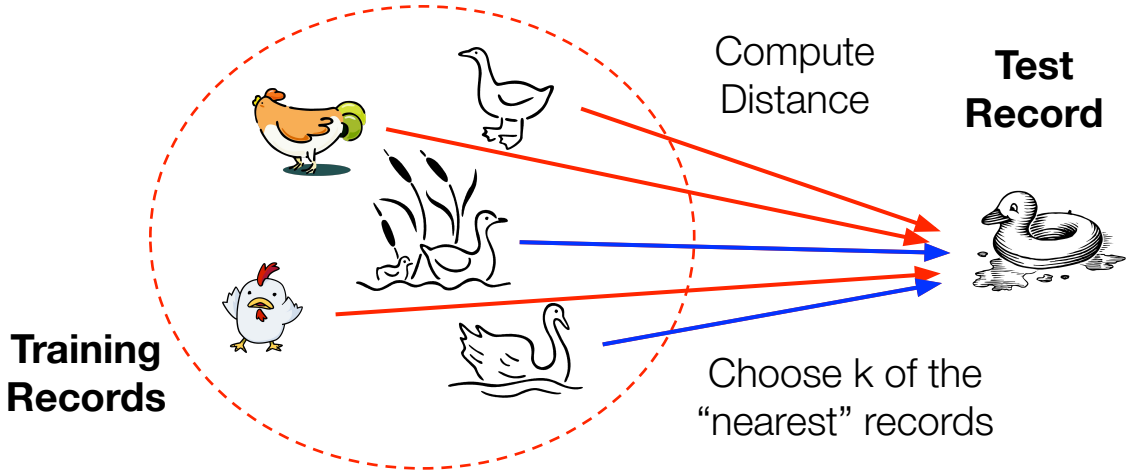
Basado en las slides de Bárbara Poblete y Felipe Bravo

Clasificador KNN

Nearest Neighbor Classifier (o k-nn): Clasificador basado en **instancias**. Conocido como **lazy**.

- Usa los k puntos más cercanos (nearest neighbors) para realizar la clasificación

Idea: If it walks like a duck, quacks like a duck, then it's probably a duck.



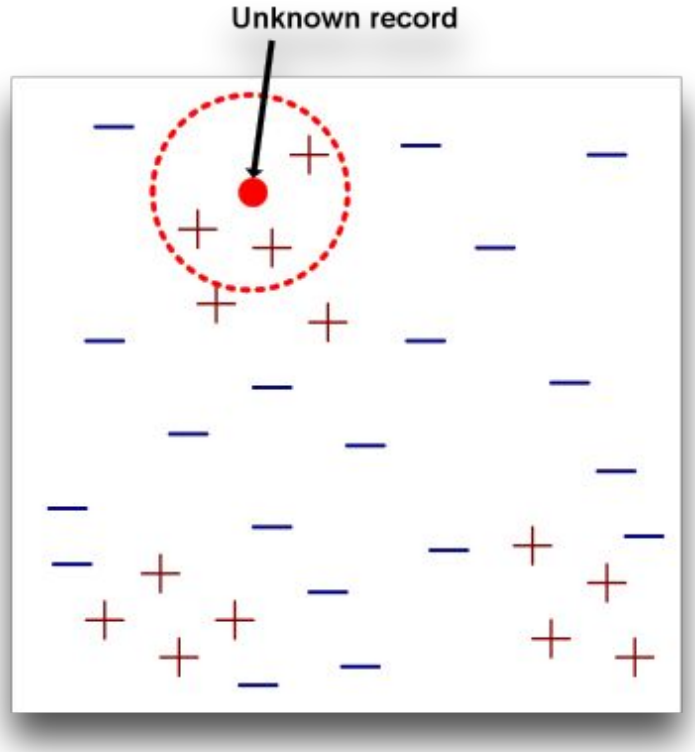
Clasificador KNN

Necesita 3 cosas:

- Set de **records** almacenados.
- **Métrica de distancia** para calcular la distancia entre records.
- Valor de **k**, el número de vecinos cercanos a obtener.

Para clasificar un récord nuevo:

1. Calcular la distancia los los récords almacenados.
2. Identificar k nearest neighbors .
3. Utilizar la clase de los knn para asignar la clase al record nuevo (e.j. voto de la mayoría).



Métricas de distancia

Para atributos númericos usamos la distancia **euclidiana**:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2},$$

Una versión más general es la distancia de **Minkowsky** (r=1 => distancia Manhattan, r=2 => distancia euclidiana)

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

-> *Es muy importante que los atributos estén normalizados.*

Escalando atributos

Problemas de escalas: Atributos deben ser escalados para prevenir que algún atributo domine la métrica de distancia. Ejemplos:

- La altura de una persona puede variar entre 1.5m a 1.8m
- El peso puede variar entre 40 kg a 150 kg
- El ingreso de una persona puede variar entre $150K a $10M

Técnicas para escalar atributos

Normalización a media cero y varianza unitaria:

$$\frac{x - \mu_x}{\sigma_x}$$

sklearn.preprocessing.
StandardScaler

Normalización a rango entre 0 y 1:

$$\frac{x - min_x}{max_x - min_x}$$

sklearn.preprocessing.
MinMaxScaler

OJO: Apliquen la misma transformación a los datos de training y testing -> los valores de normalización se calculan sobre los datos de training.

Distancia y similitud

Es importante distinguir entre métricas de similitud y métricas de distancia.

- **Similitud:** entre más cerca dos objetos mayor el valor de la métrica.
- **Distancia:** entre más lejos dos objetos mayor el valor de la métrica.

Attribute Type	**Dissimilarity**	**Similarity**
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = \lvert x - y \rvert / (n-1)$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - d$
Interval or Ratio	$d = \lvert x - y \rvert$	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Similitud Coseno

Corresponde al coseno del ángulo entre los dos vectores.

Cuando nuestros objetos son vectores sparse (muchas columnas con cero) es conveniente usar la similitud coseno.

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|} \qquad \|\mathbf{x}\| = \sqrt{\sum_{k=1}^{n} x_k^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}.$$

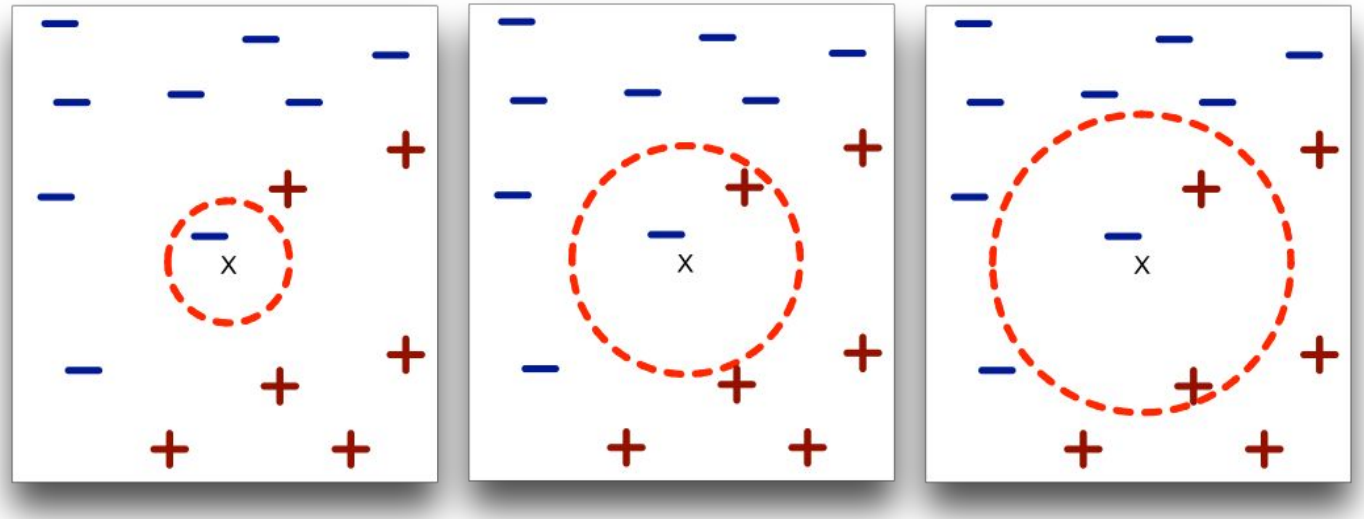
Similitud Coseno

Un ejemplo común es cuando tratamos documentos como bolsas de palabras (cada columna es una palabra del vocabulario). Por ejemplo:

$$\mathbf{x} = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$$
$$\mathbf{y} = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$$

$$\mathbf{x} \cdot \mathbf{y} = 3 * 1 + 2 * 0 + 0 * 0 + 5 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 1 + 0 * 0 + 0 * 2 = 5$$
$$\|\mathbf{x}\| = \sqrt{3 * 3 + 2 * 2 + 0 * 0 + 5 * 5 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 2 + 0 * 0 + 0 * 0} = 6.48$$
$$\|\mathbf{y}\| = \sqrt{1 * 1 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 1 * 1 + 0 * 0 + 2 * 2} = 2.24$$
$$\mathbf{cos}(\mathbf{x}, \mathbf{y}) = \mathbf{0.31}$$

Definición de NN

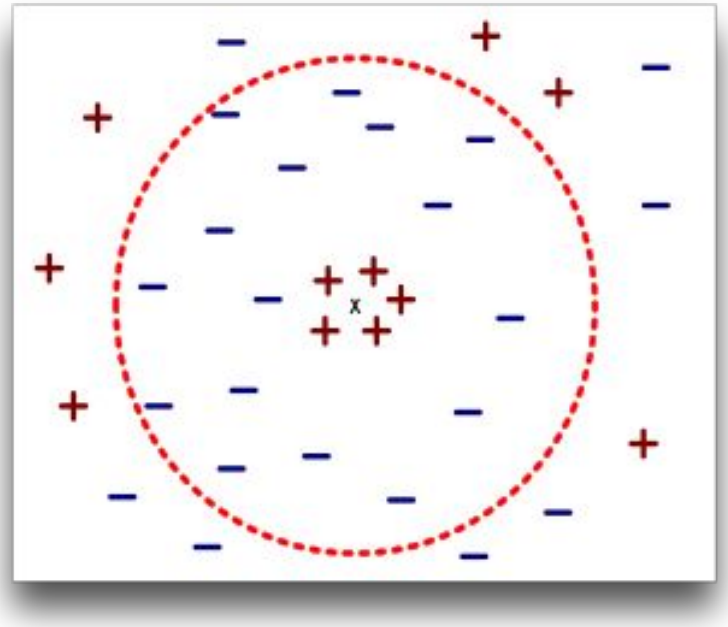


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-NN de un record x son los puntos que tienen las k menores distancias a x

Eligiendo el valor de K

- k muy pequeño es susceptible a ruido
- k muy grande puede incluir puntos de otra clase



Clasificación kNN

Los clasificadores k-NN son **lazy learners**.

- No construyen modelos explícitos, es más flexible ya que no necesita comprometerse con un modelo global a priori.
- Al contrario de otros **eager learners** como los árboles de decisión o clasificadores basados en reglas.
- Es independiente del nro. de clases.
- La clasificación es más costosa (memoria y tiempo).

Comentarios

- La Maldición de la Dimensionalidad
  - Cuando los datos tienen una alta dimensionalidad, KNN está sujeta a la Maldición de la Dimensionalidad.
  - Fenómeno en que muchos tipos de análisis de datos se vuelven significativamente más difíciles a medida que aumenta la dimensionalidad de los datos.
  - Para la clasificación, esto puede significar que no haya suficientes ejemplos para crear un modelo que asigne de forma confiable una clase a todos los ejemplos posibles.
  - Para técnicas basadas en distancias (KNN, K-means) las distancias entre objetos se vuelven menos claras cuando hay muchas dimensiones.

Ejemplo

https://colab.research.google.com/drive/1gIDJo0yHoTZIAboBAk4y9VGi3YyMQlLU?usp=sharing

Clasificación basada en Naïve Bayes

- Familia de modelos basados en Bayes.
- Veremos Clasificador de Naive Bayes.
- También existen las Redes Bayesianas.

Clasificación Basada en Naïve Bayes

- Modelo que busca **modelar la relación probabilística** entre atributos y clase.
- Modelo generativo, asume una distribución conjunta entre X e Y.
- Supuesto: atributos independientes dado la clase (**naive assumption**).

Clasificador Bayesiano

Esquema probabilístico para resolver problemas de clasificación.

- Probabilidad condicional:

$$P(C \mid A) = \frac{P(A,C)}{P(A)}$$

$$P(A \mid C) = \frac{P(A,C)}{P(C)}$$

- Teorema de Bayes:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Ejemplo Teorema de Bayes

Dado:

- Un doctor sabe que la meningitis produce rigidez de cuello el 50% de las veces.
- La probabilidad previa de que cualquier paciente tenga meningitis es 1/50,000.
- La probabilidad previa de que cualquier paciente tenga rigidez en el cuello es de 1/20.

¿Si un paciente tiene el cuello rígido, cuál es la probabilidad de que tenga meningitis?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Clasificador Naïve Bayes

- Considerar cada atributo como variable condicionalmente independiente de la clase (eso es “naive”).
- Dado un record con atributos (A1, A2,…, An).
  - La meta es predecir la clase C.
  - Específicamente queremos encontrar el C que maximice P(C| A1, A2,…,An).
- ¿Podemos estimar P(C| A1, A2,…,An) directamente de los datos?

Clasificador Naïve Bayes

Aproximación

- Computar la probabilidad posterior P(C | A1, A2, …, An) para todos los valores de C usando el Teorema de Bayes.

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

- Elegir un valor de C que maximice P(C | A1, A2, …, An).
- Equivalente a elegir un valor de C que maximice P(A1, A2, …, An|C) P(C).
- Esto es porque el numerador P(A1A2...An) es constante para todas las clases.

Clasificador Naïve Bayes

- Asume independencia entre los atributos Ai cuando la clase está dada (independencia condicional):
- $P(A1, A2, \ldots, An \mid C) = P(A1 \mid Cj)\, P(A2 \mid Cj) \ldots P(An \mid Cj)$.
- Se puede estimar $P(Ai \mid Cj)$ para todos los Ai y Cj.
- Un punto nuevo A, se clasifica como Cj si $P(Cj) \prod P(Ai \mid Cj)$ es máxima (en comparación con otros valores de C).

¿Cómo estimar probabilidades a partir de los datos?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

¿Cómo calculamos P(Aik=b|Ck) cuando el atributo Ai es numérico?
-> Una opción es discretizar el atributo y proceder de la forma anterior.

- Clase: $P(C_k) = \frac{count(C_k)}{N}$
  - e.g., P(No) = 7/10, P(Yes) = 3/10
- Para atributos discretos:

$$P(A_i = b|C_k) = \frac{count(A_{ik} = b)}{count(C_k)}$$

  - donde count(A_{ik}=b) es el número de instancias que tiene el valor b para el atributo A_i y que pertenecen a la clase C_k
  - Ejemplos:
  - P(Status =Married|No) = 4/7
    P(Refund= Yes |Yes) = 0

Laplace Smoothing

- P(C | A1, A2, …, An) se puede ir a cero cuando |Aik=b| = 0, osea cuando para alguna clase Ck no hay ningún ejemplo con Ai=b.
- En ese caso Naive Bayes le asignaría probabilidad cero a la clase Ck a cualquier ejemplo con |Aik=b| = 0, ignorando el valor de los otros atributos (acuérdense que las probabilidades se multiplican).
- Eso no es bueno para la generalización del modelo.

Laplace Smoothing: soluciona el problema sumándole 1 a todos los conteos para que ninguna probabilidad quede en cero:

$$P(A_i = b|C_k) = \frac{count(A_{ik} = b) + 1}{count(C_K) + values(A_i)}$$

Donde values(Ai) es la cantidad de categorías del atributo Ai.

Con Laplace smoothing P(Status =Married|No) = (4+1)/(7+3)

Naïve Bayes (Resumen)

- Es robusto ante puntos de ruido aislados.
- Maneja valores faltantes ignorando la instancia durante los cálculos de estimación de probabilidades.
- Robusto a atributos irrelevantes (afectan de igual manera a todas las clases).
- El supuesto de independencia entre atributos puede no ser cierto en todos los casos.
- Las redes Bayesianas o los modelos gráficos dirigidos permiten hacer modelos probabilísticos con supuestos de independencia menos restrictivos.

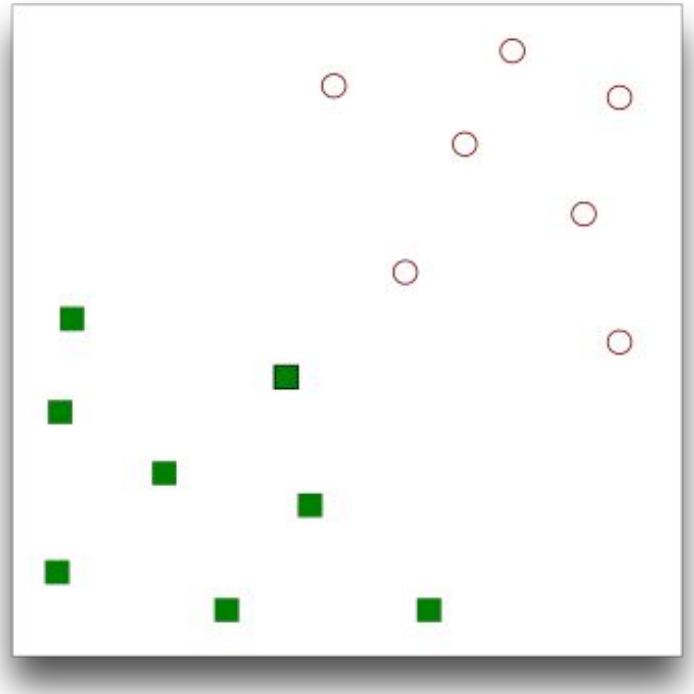
Support Vector Machines

Se formula el problema de clasificación como un problema de optimización.

Encontrar un hiperplano lineal (decision boundary).

- Representa una frontera de decisión usando un subconjunto de ejemplos de entrenamiento, conocidos como support vectors.
- Trabaja bien con datos de alta dimensionalidad y evita la "curse of dimensionality"

Support Vector Machines

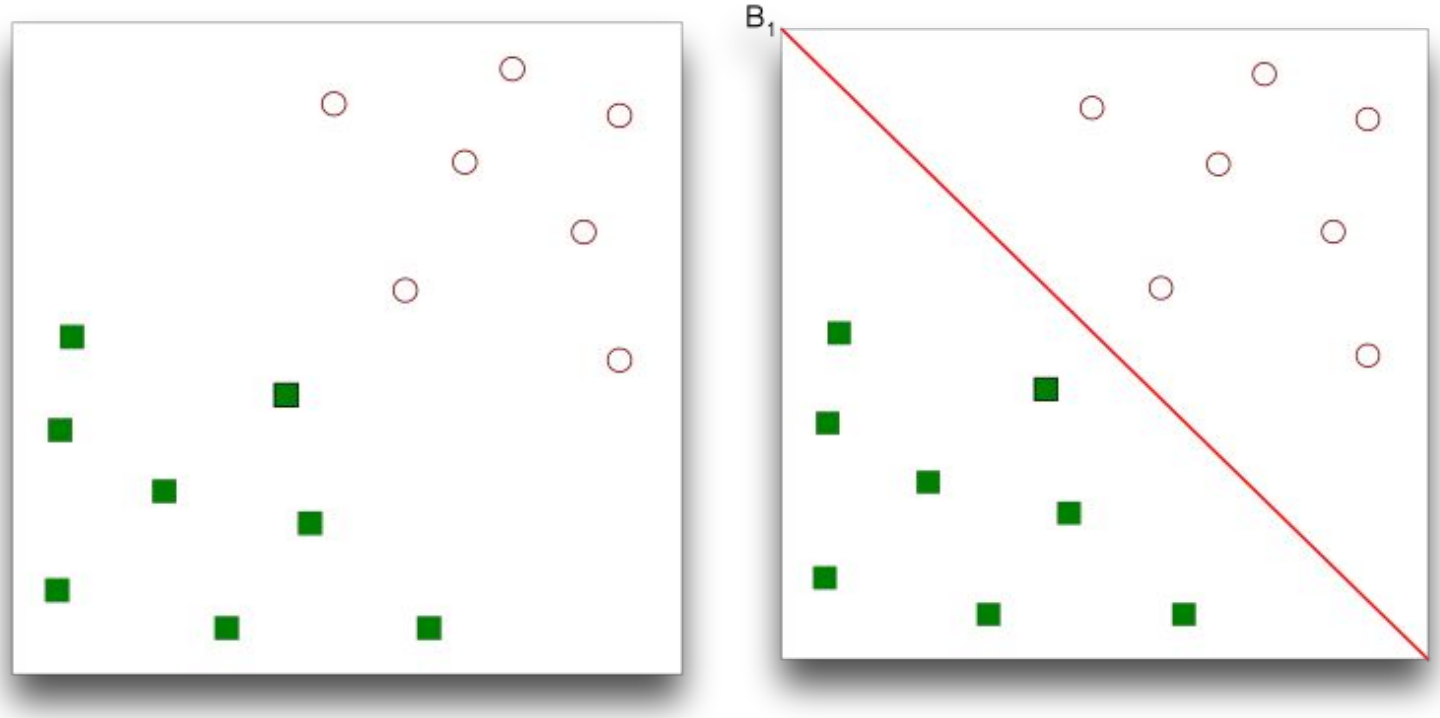


Ejemplo: Tenemos un dataset de n-dimensiones (proyectado a 2).

Podemos decir que estos datos son linealmente separables (separables por un hiperplano)

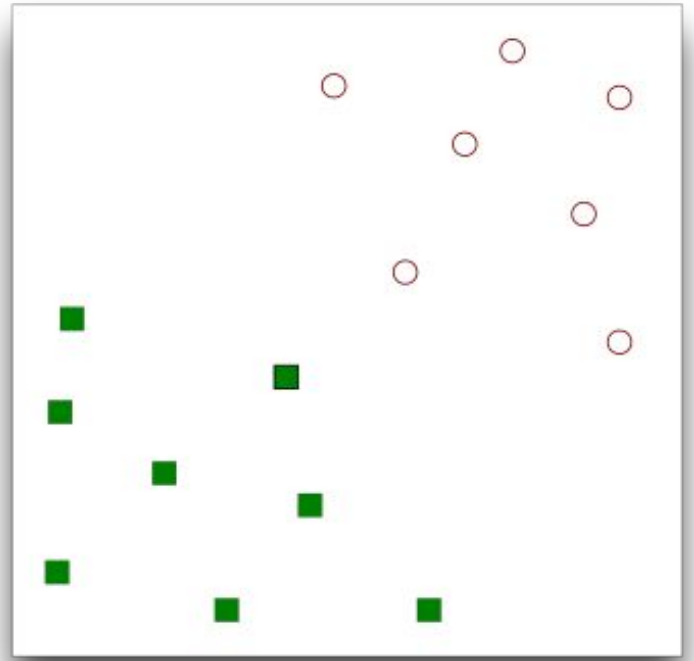
- Encontrar un hiperplano lineal (decision boundary) que separe los ejemplos positivos de los negativos.

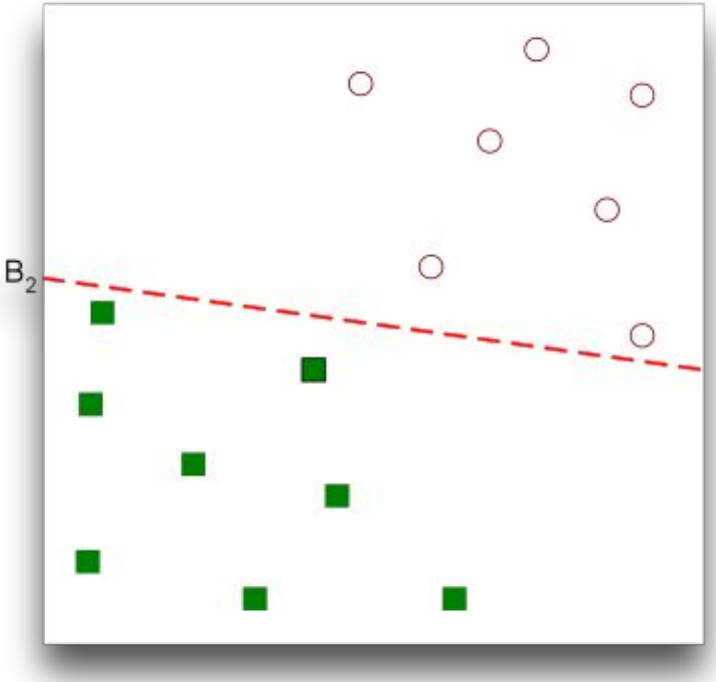
Support Vector Machines



Una posible solución

Support Vector Machines

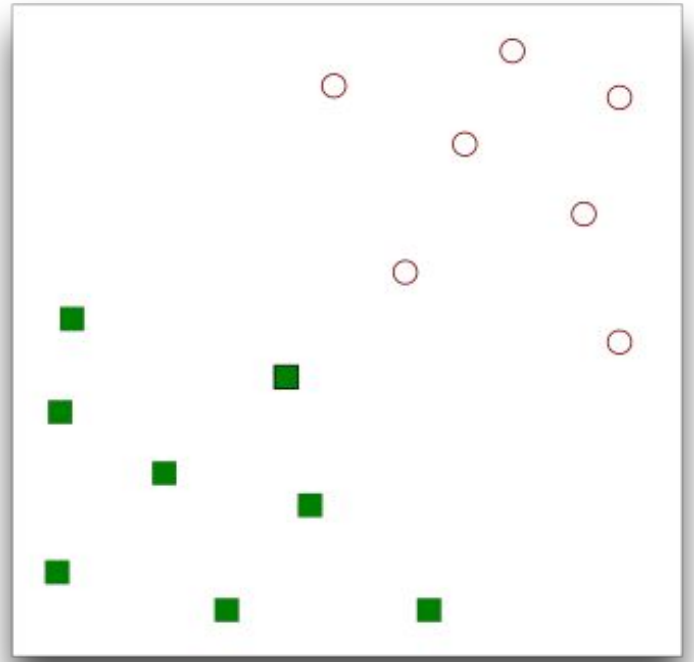


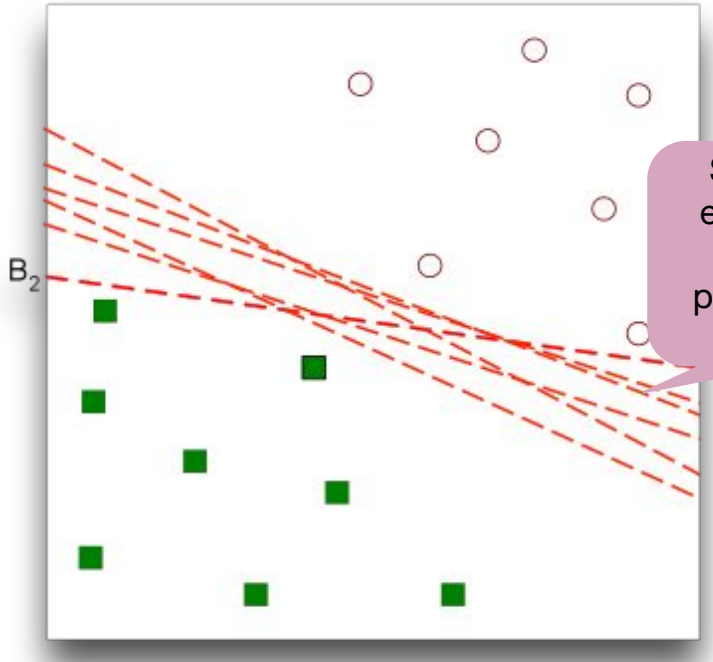


Otra posible solución

Support Vector Machines

Este modelo puede extenderse a datos que no son linealmente separables.

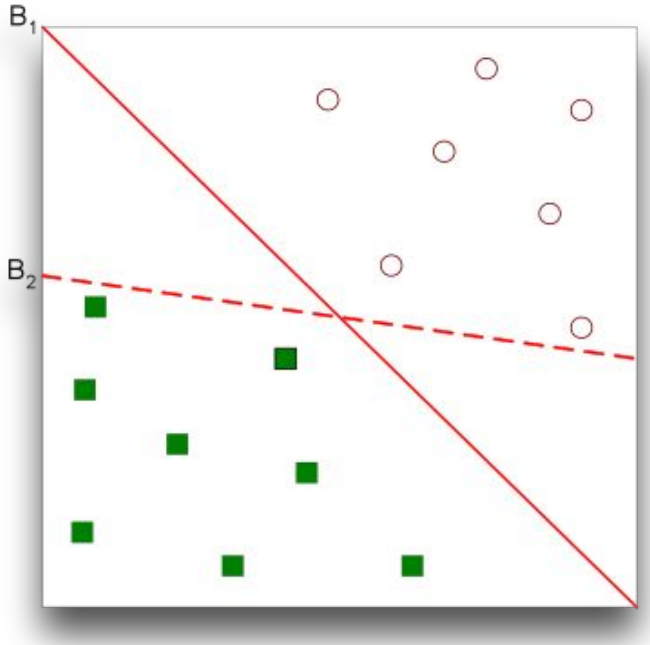




SVM se basa en aprender a encontrar el plano marginal maximal

¡Infinitas soluciones!

Support Vector Machines



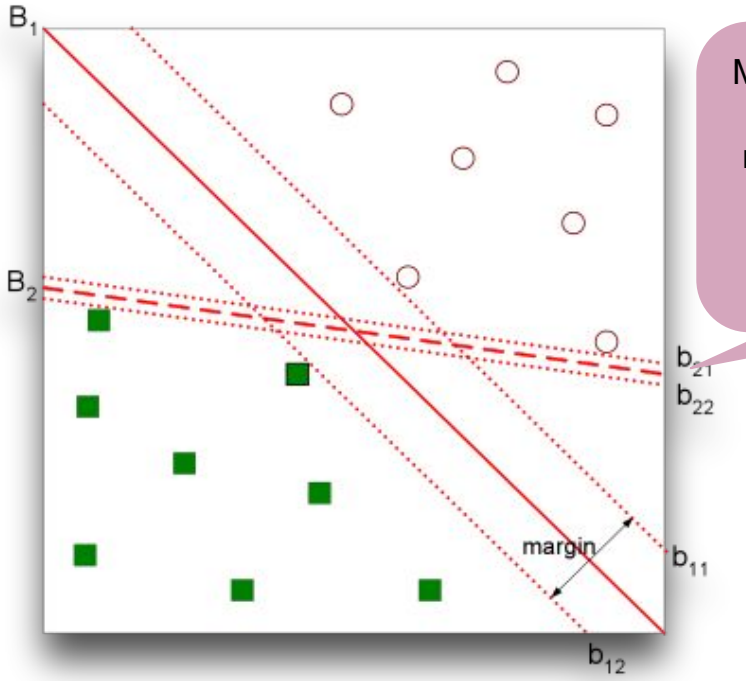
¿Cuál es mejor? ¿B1 o B2?
¿Cómo definimos qué es mejor?

Support Vector Machines

Encontrar un hiperplano que **maximice** el margen de entrenamiento (menores errores de generalización)

- El margen del hiperplano es la distancia que hay de los puntos positivos y negativos más cercanos al hiperplano.

Entre más ancho el margen, mayor el poder de generalización.
=> B1 es mejor que B2



Menor margen hace que sea más susceptible a ruido y perturbaciones, lo que puede llevar a overfitting y mala generalización.

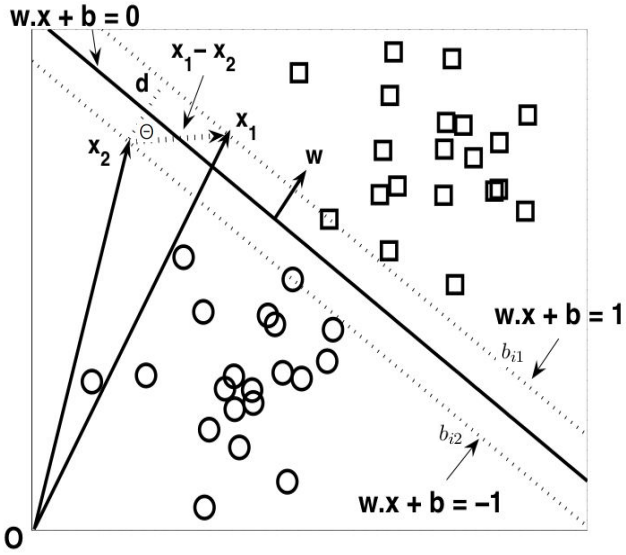
Support Vector Machines

Un SVM lineal es un clasificador que busca un hiperplano con el máximo margen.

- Sea un problema de clasificación binaria con N ejemplos, cada ejemplo se representa como la tupa xi,yi (i=1,2,...,N), donde xi es un vector de d dimensiones $(x_{i1},x_{i2},...,x_{id})^T$ e $y_i \in \{-1, 1\}$ (ejemplos negativos y positivos).
- El límite de decisión de un clasificador lineal se escribe de la siguiente manera:

$$\mathbf{w} \cdot \mathbf{x} + b = 0,$$

- Donde w y b son parámetros del modelo.

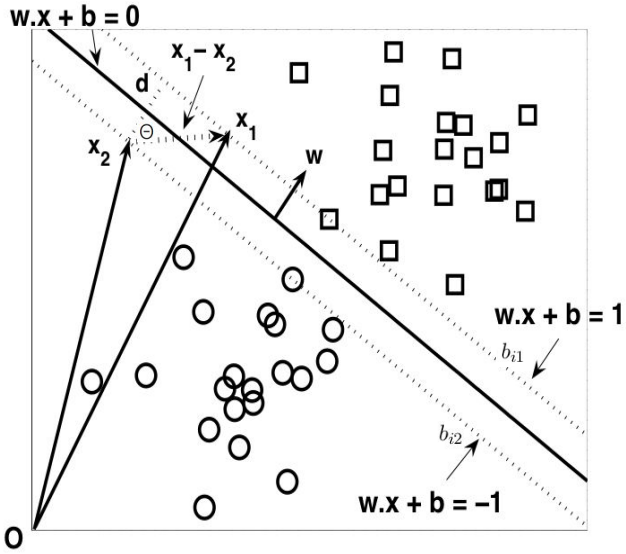


Support Vector Machines

Consiste en:

1. Formular el margen en función de los parámetros.
2. Formular un problema de optimización que permita encontrar el hiperplano de máximo margen.

Support vector machine (SVM) a detalle. Prof. Felipe Bravo. https://www.youtube.com/watch?v=_P_ArDrCQSM



Support Vector Machines

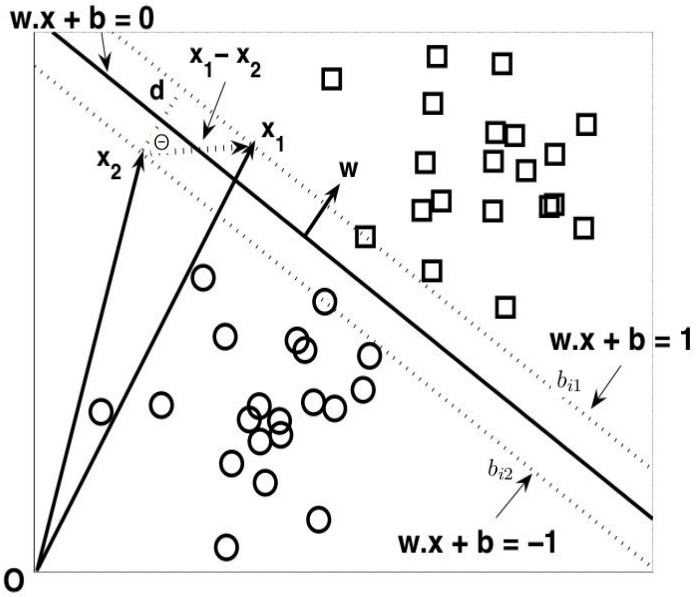
- El **hiperplano w·x+b=0** separa los ejemplos positivos (cuadrados) de los negativos (círculos)
- Cualquier ejemplo que se encuentre en el límite de decisión debe satisfacer la ecuación del hiperplano.
- Por ejemplo, sean x_a y x_b dos puntos localizados en el límite de decisión:

$$\mathbf{w} \cdot \mathbf{x}_a + b = 0,$$
$$\mathbf{w} \cdot \mathbf{x}_b + b = 0.$$

- Si restamos las dos ecuaciones obtenemos:

$$\mathbf{w} \cdot (\mathbf{x}_b - \mathbf{x}_a) = 0,$$

- Cómo x_b-x_a es paralelo al límite de decisión, entonces w es perpendicular al hiperplano (el producto punto de dos vectores ortogonales es cero).



Support Vector Machines

- Para cualquier cuadrado x_s localizado **sobre el límite de decisión**, se cumple que:

$$\mathbf{w} \cdot \mathbf{x}_s + b = k,$$

donde k > 0.

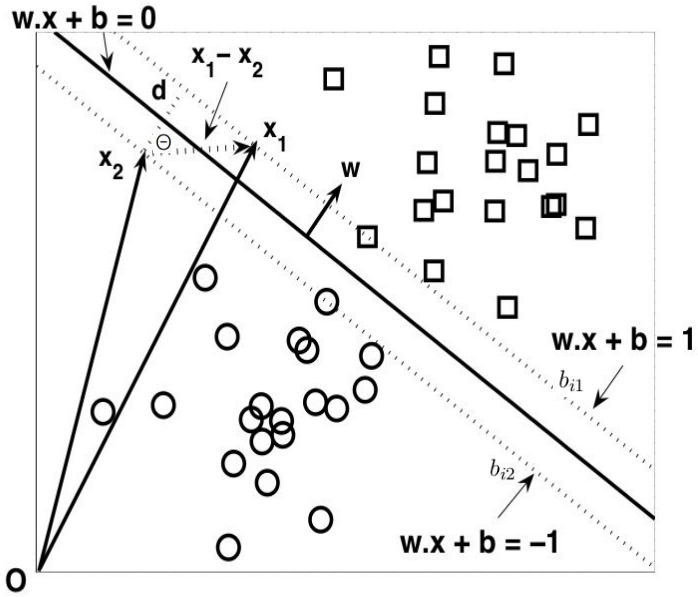
- De manera análoga, para cualquier círculo x_c localizado **bajo el límite de decisión**, se cumple que:

$$\mathbf{w} \cdot \mathbf{x}_c + b = k',$$

done k'<0.

- Si etiquetamos todos los cuadrados como la clase positiva +1 y todos los círculos como la clase negativa -1, podemos predecir la etiqueta **y** para cualquier ejemplo de test **z** de la siguiente manera:

$$y = \begin{cases} 1, & \text{if } \mathbf{w} \cdot \mathbf{z} + b > 0; \\ -1, & \text{if } \mathbf{w} \cdot \mathbf{z} + b < 0. \end{cases}$$



El Margen de un Clasificador Lineal

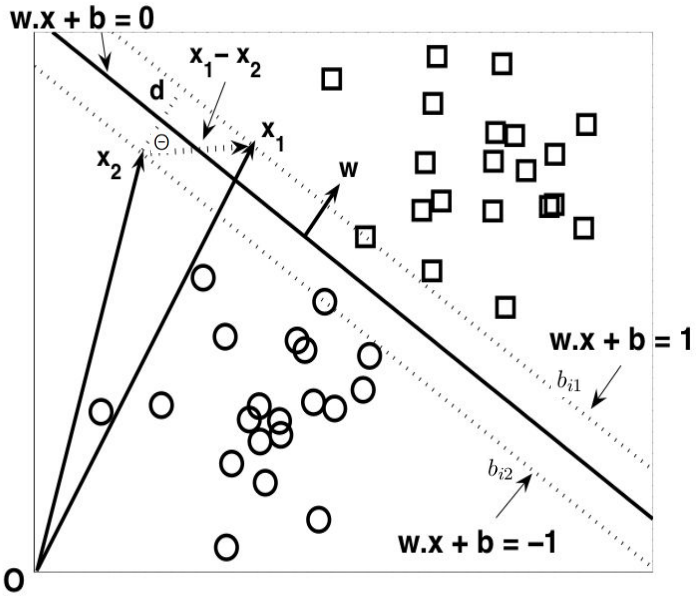
- Podemos re-escalar los parámetros w y b del límite de decisión de tal manera que los hiperplanos paralelos b_{i1} y b_{i2} puedan expresarse de la siguiente manera:

$$b_{i1} : \ \mathbf{w} \cdot \mathbf{x} + b = 1,$$
$$b_{i2} : \ \mathbf{w} \cdot \mathbf{x} + b = -1.$$

- El margen del límite de decisión se calcula como la distancia entre estos dos hiperplanos.
- Calculamos la distancia del círculo x_1 y el cuadrado x_2. Esto se hace sustituyendo x_1 y x_2 en las ecuaciones de b_{i1} y b_{i2} respectivamente. Lo que nos da:

1) **$w \cdot x_1 + b = 1$** y 2) **$w \cdot x_2 + b = -1$**.

- Si sustraemos la segunda ecuación de la primera obtenemos **$w \cdot (x_1 - x_2) = 2$**

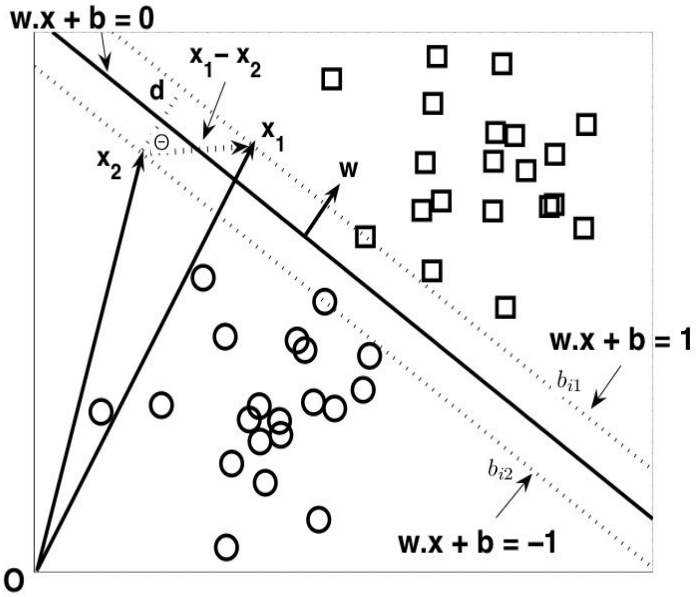


El Margen de un Clasificador Lineal

- El producto punto de dos vectores (a·b) se puede representar como la norma ||a||*||b||*cos(Θ).
- Tenemos entonces que $||w||*||x_1-x_2||*\cos(\Theta)=2$
- Si miramos la figura anterior podemos ver que $||x_1-x_2||*\cos(\Theta) = d$.
- Entonces

$$\|\mathbf{w}\| \times d = 2$$
$$\therefore d = \frac{2}{\|\mathbf{w}\|}.$$

Queremos maximizar este margen



¡ Encontramos una expresión del margen que depende de w !

Aprendiendo una SVM Lineal

La fase de **entrenamiento de una SVM lineal** implica la estimación de los parámetros w y b del límite de decisión a partir de los datos de entrenamiento.

Los parámetros deben elegirse de manera que se cumplan las dos siguientes condiciones:

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq 1 \text{ if } y_i = 1,$$
$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 \text{ if } y_i = -1.$$

- Estas condiciones imponen el requisito de que todas las instancias de entrenamiento de la clase y = 1 (los cuadrados) deben estar situadas en o **sobre el hiperplano w·x+b = 1**.
- Mientras que las instancias de la clase y = -1 (los círculos) deben estar situadas **en o debajo del hiperplano w·x+b = -1**.

Aprendiendo una SVM Lineal

Maximizar el margen equivale a minimizar la siguiente función objetivo:

$$f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2}.$$

$$\max_w \frac{2}{||w||} \Leftrightarrow \min_w \frac{||w||^2}{2}$$

La tarea de aprendizaje en SVM puede formalizarse como el siguiente problema de optimización con restricciones:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to} \qquad y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

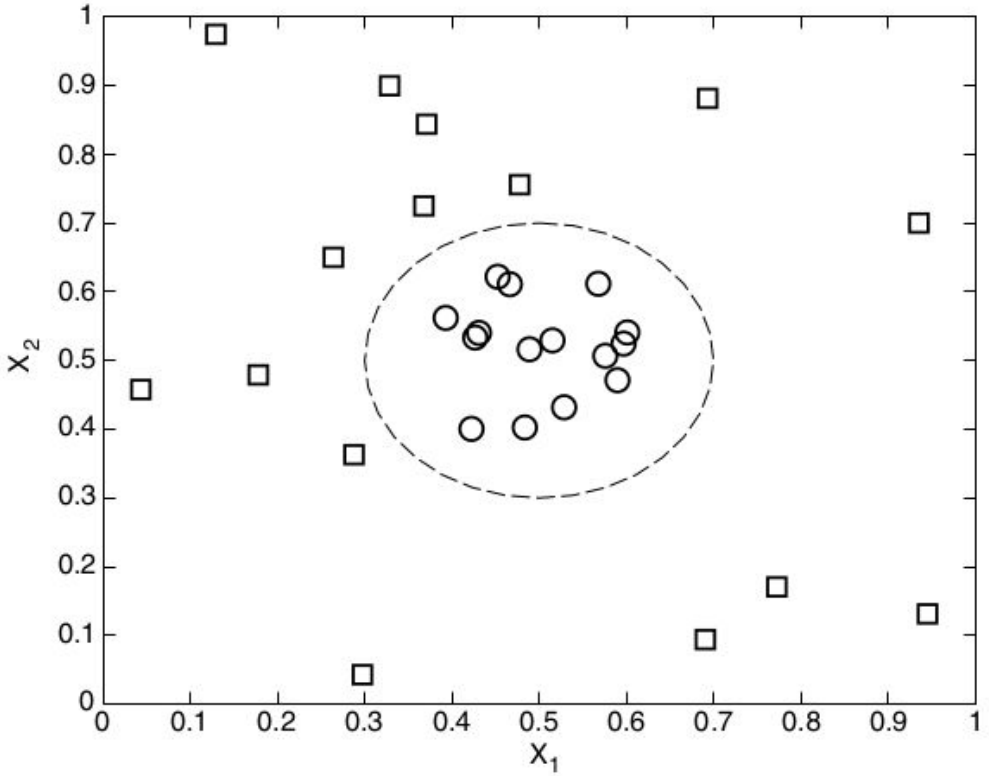
Dado que la función objetivo es cuadrática y las restricciones son lineales para los parámetros w y b, esto se conoce como un **problema de optimización convexa**.

Support Vector Machines No-Lineales

El diagrama muestra un ejemplo de un dataset bidimensional compuesto por cuadrados (y = 1) y círculos (y = -1).

Todos los círculos están agrupados cerca del centro del diagrama y todos los cuadrados se distribuyen más lejos del centro.

Este problema no se puede resolver con una SVM lineal.



Support Vector Machines No-Lineales

Podríamos aplicar una **transformación no lineal Φ** para mapear los datos de su espacio de original a un nuevo espacio (de más dimensiones) donde el límite de decisión se vuelva lineal.

Supongamos que elegimos la siguiente transformación que transforma de 2 dimensiones a 5 dimensiones:

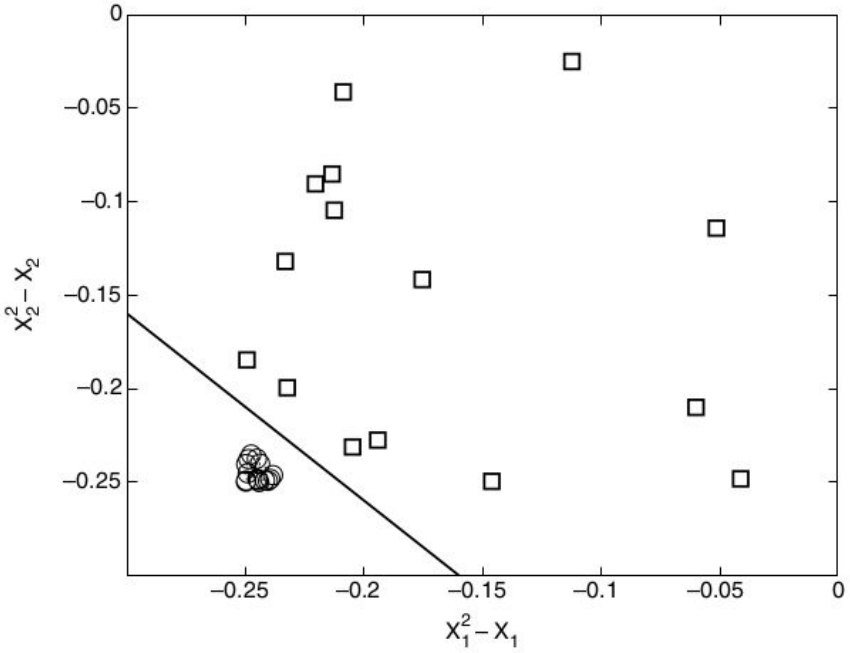
$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

En este espacio transformado si es posible encontrar parámetros $\mathbf{w} = (w_0, w_1, \ldots, w_4)$ que separen linealmente los datos:

$$w_4 x_1^2 + w_3 x_2^2 + w_2\sqrt{2}x_1 + w_1\sqrt{2}x_2 + w_0 = 0.$$

Support Vector Machines No-Lineales

- La figura muestra que en el espacio transformado se puede construir un límite de decisión lineal para separar las clases.
- Un problema potencial de este enfoque es que puede sufrir de la **maldición la dimensionalidad**: para datos de alta dimensión muchas técnicas de data mining no escalan o no funcionan bien.
- Mostraremos cómo una SVM no lineal evita este problema usando un truco llamado **Kernel Trick**.



Support Vector Machines No-Lineales

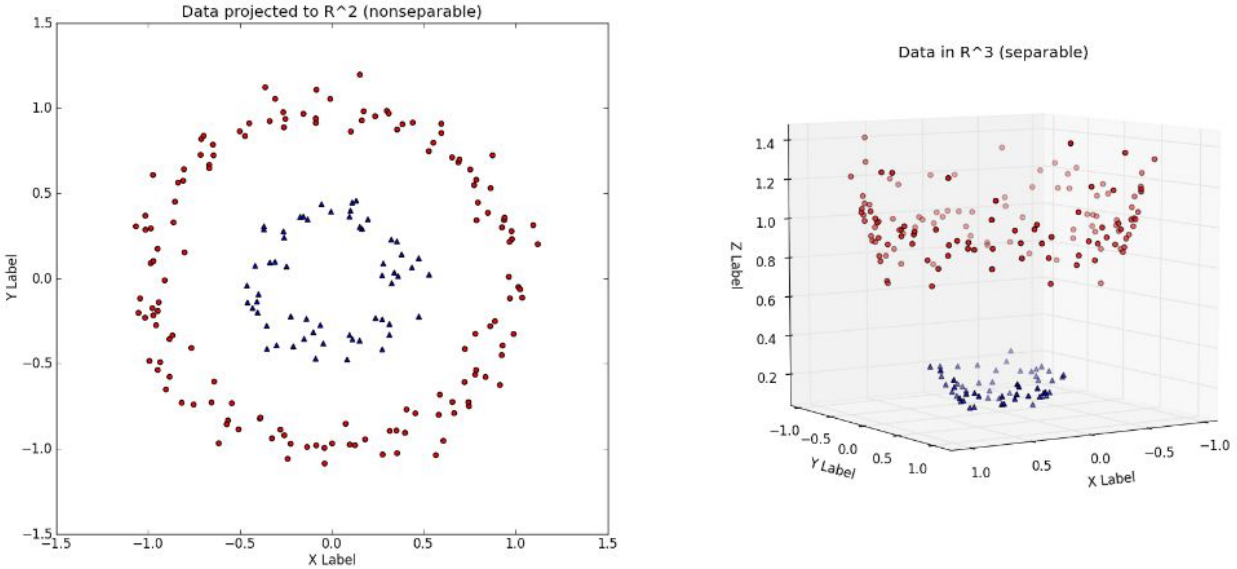


Figure 5: (Left) A dataset in $\mathbb{R}^2$, not linearly separable. (Right) The same dataset transformed by the transformation: $[x_1, x_2] = [x_1, x_2, {x_1}^2 + {x_2}^2]$.

source: http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Support Vector Machines No-Lineales

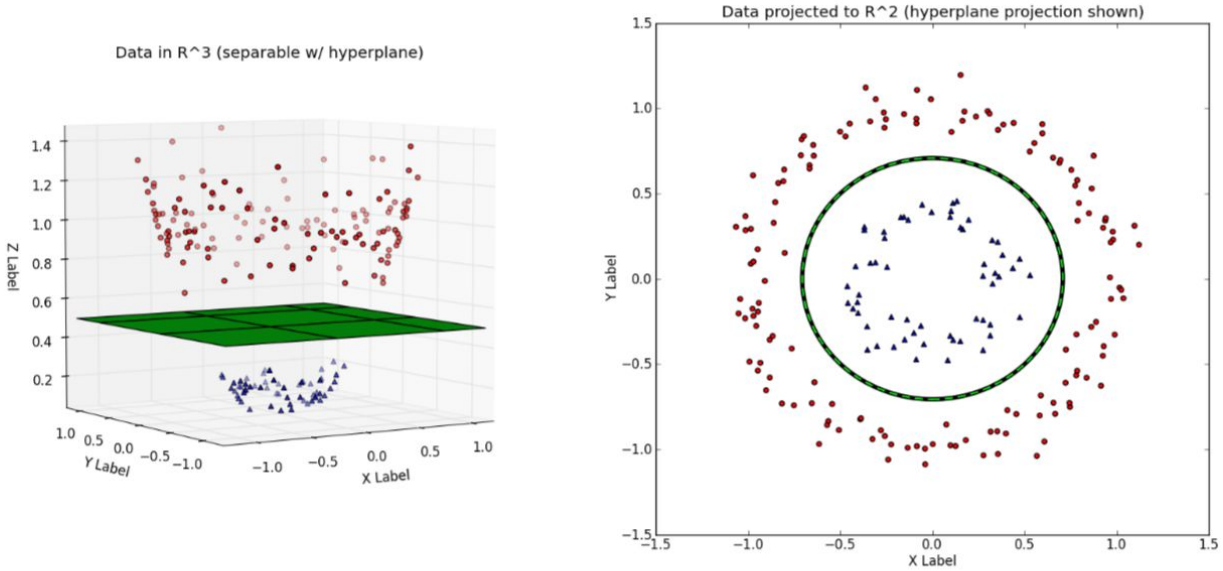
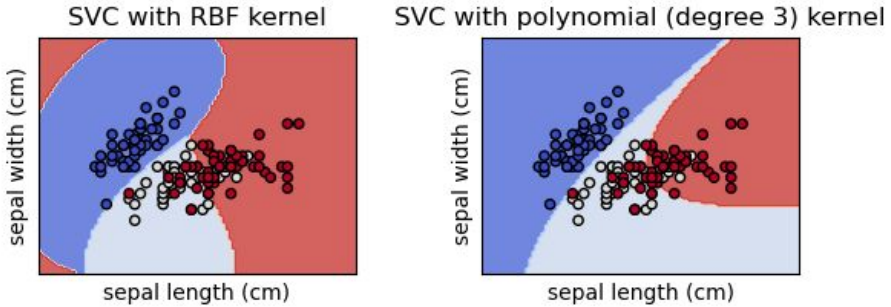


Figure 6: (Left) The decision boundary $\vec{w}$ shown to be linear in $\mathbb{R}^3$. (Right) The decision boundary $\vec{w}$, when transformed back to $\mathbb{R}^2$, is nonlinear.

source: http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Support Vector Machines No-Lineales

- Se pueden especificar diferentes funciones de kernel para la función de decisión. Se proporcionan kernels comunes, pero también es posible especificar kernels personalizados.
  - Polinomial
  - Exponencial
  - Radial Basis Function (RBF)
  - etc.



Conclusiones

- La formulación de SVM presentada en esta clase se limita a problemas de clasificación binaria. Existen adaptaciones para trabajar con múltiples clases.
- El problema de aprendizaje de una SVM se formula como un problema de optimización convexa en donde hay algoritmos eficientes para encontrar el óptimo global. Otros métodos de clasificación como como los árboles de decisión y las redes neuronales tienden a encontrar óptimos locales.
- La SVM optimiza explícitamente la capacidad de generalización al maximizar el margen del límite de decisión.
- En una SVM el usuario debe ajustar hiper-parámetros, como el tipo de función de Kernel y el costo C para las variables de holgura (esto puede ser caro).
- La gran limitación de las SVMs es que no escalan bien para datasets masivos.

Ejemplo

https://colab.research.google.com/drive/1gIDJo0yHoTZIAboBAk4y9VGi3YyMQlLU?usp=sharing

dcc

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

www.dcc.uchile.cl