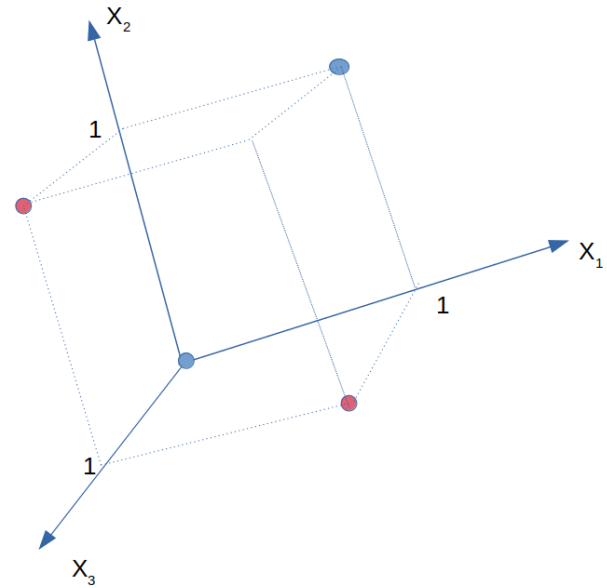
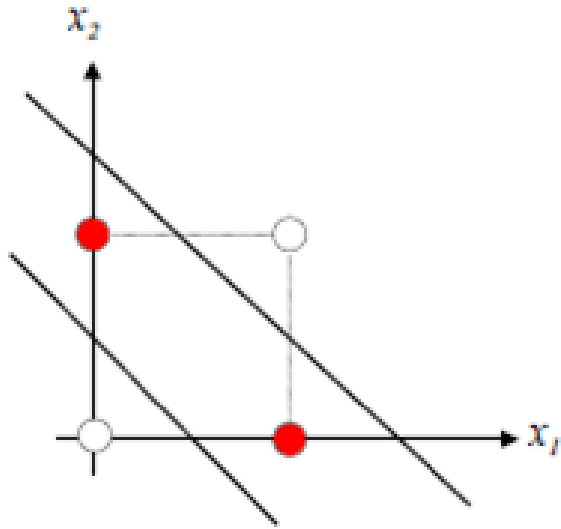




Support Vector Machines (SVM)

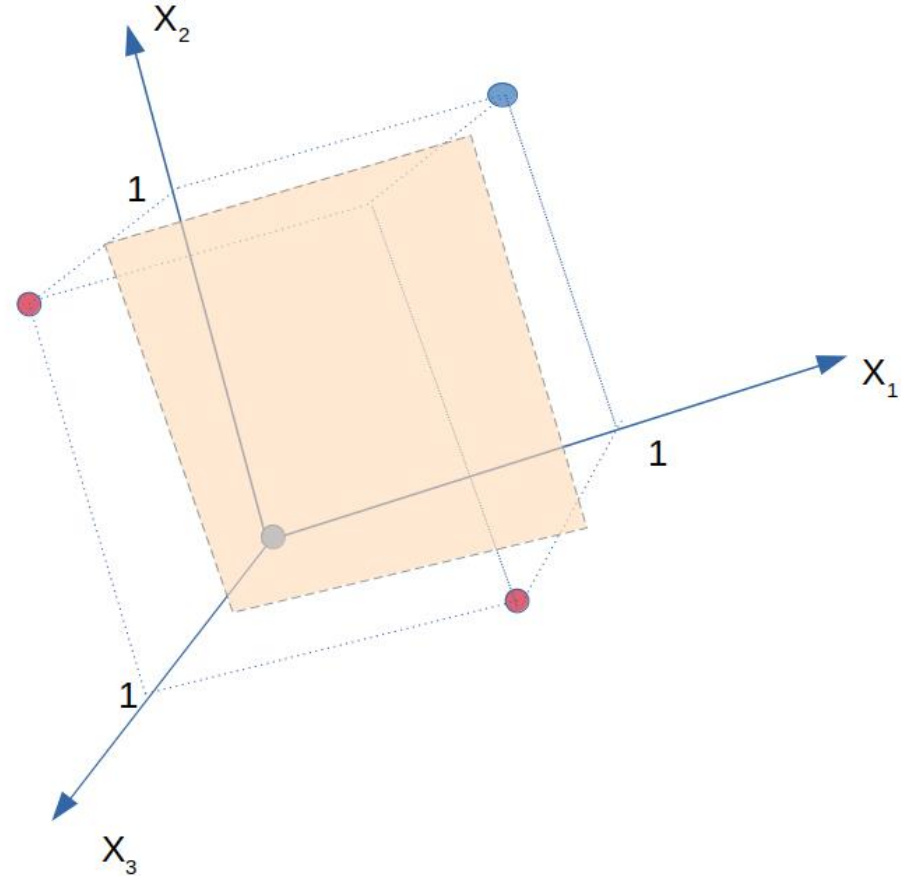
Ing. Juan M. Rodríguez

Dimensionalidad



XOR: No es linealmente separable

Dimensionalidad



XOR: Es linealmente separable !

Clasificadores basados en el margen, respecto del umbral



Masa de un puñado de ratones

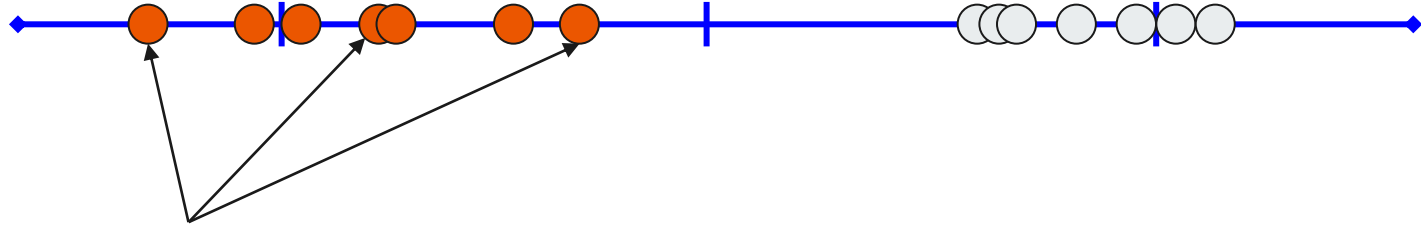
Masa (g)





Masa de un puñado de ratones

Masa (g)

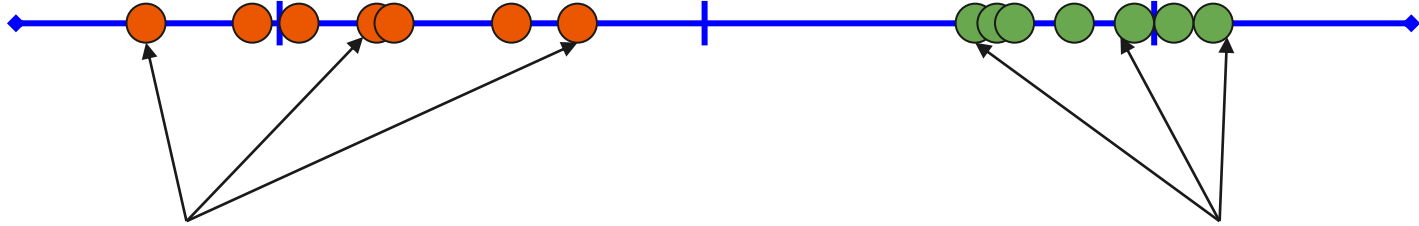


Ratones NO obesos



Masa de un puñado de ratones

Masa (g)

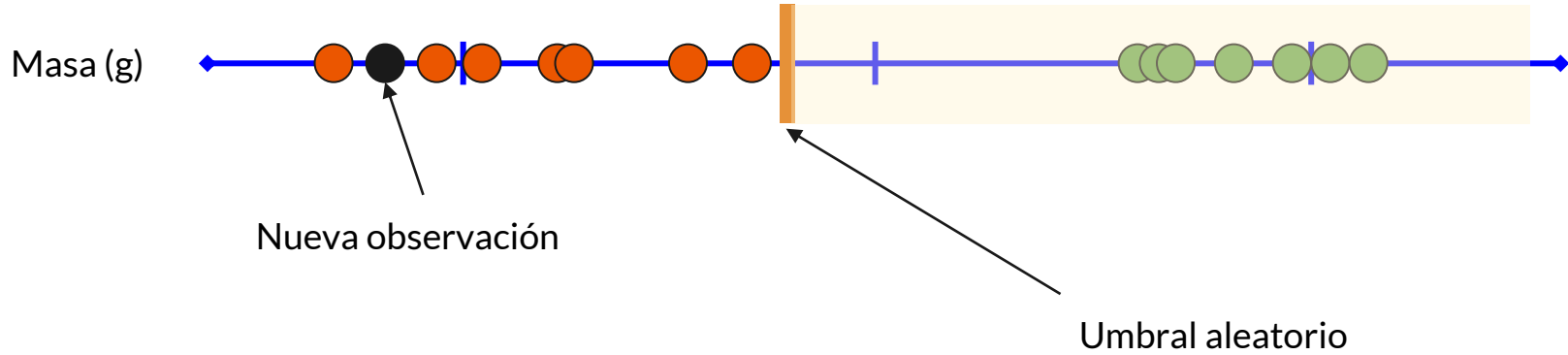


Ratones NO obesos

Ratones obesos

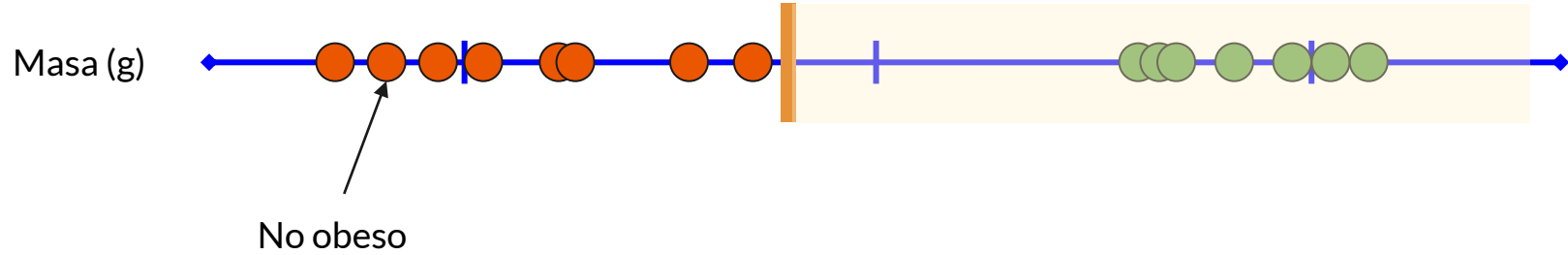


Masa de un puñado de ratones



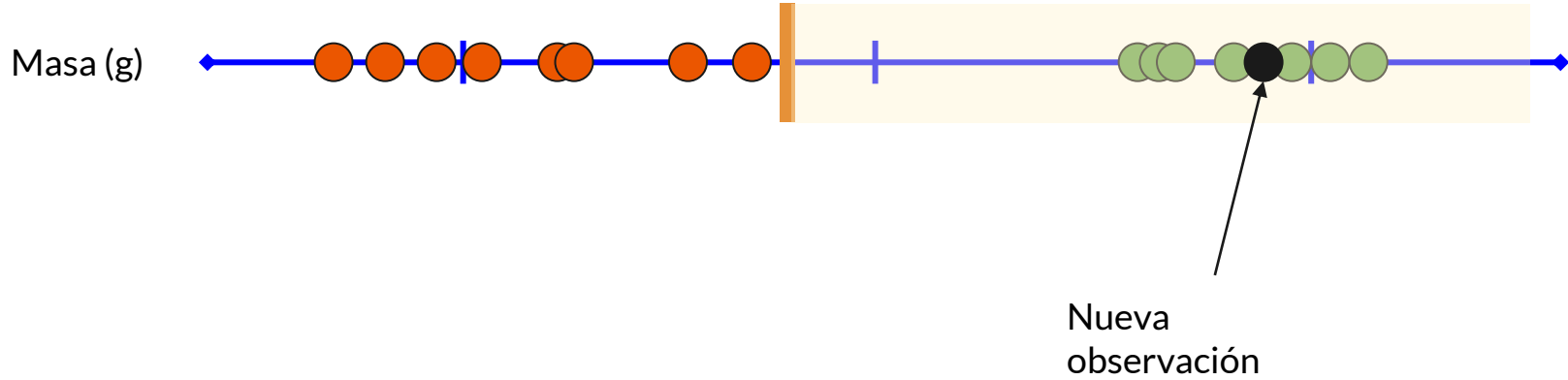


Masa de un puñado de ratones

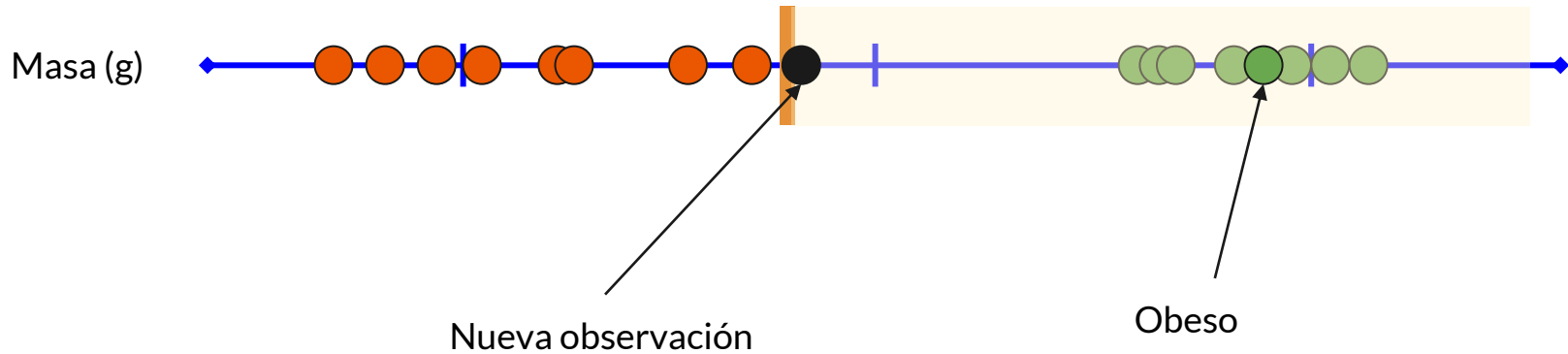




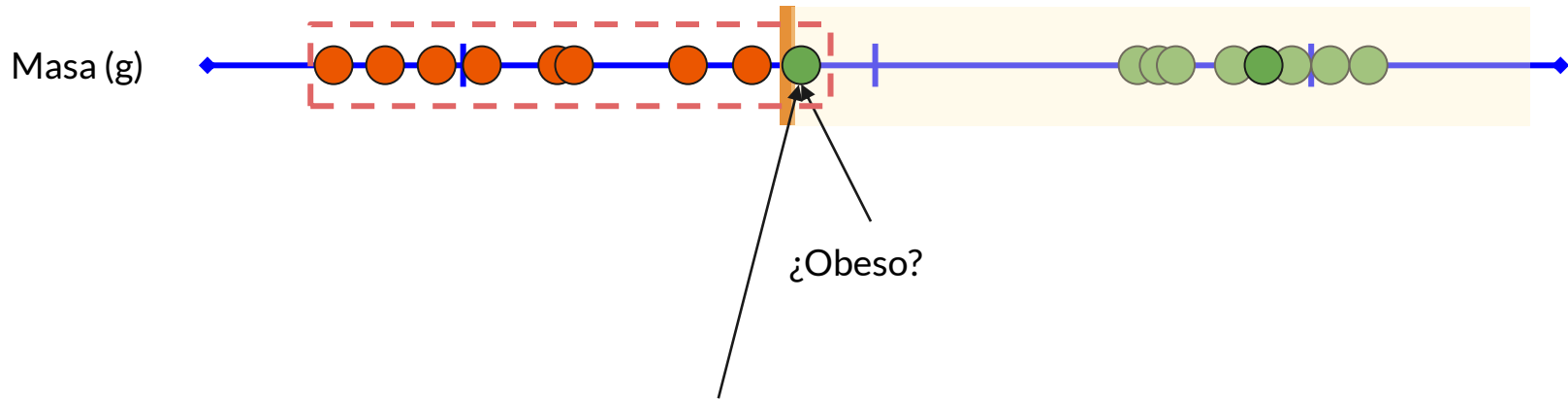
Masa de un puñado de ratones



Masa de un puñado de ratones



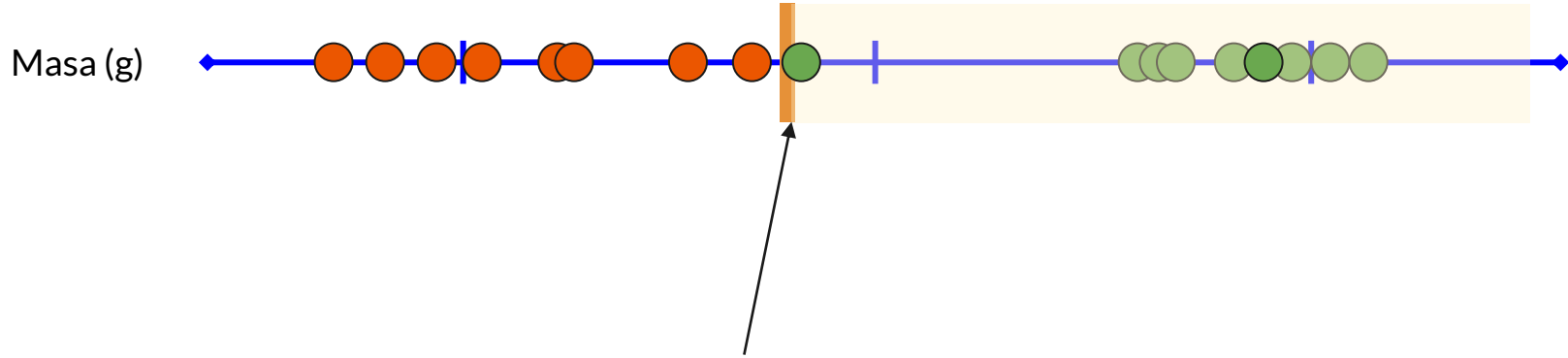
Masa de un puñado de ratones



Está mucho más cerca de los ratones no-obesos



Masa de un puñado de ratones



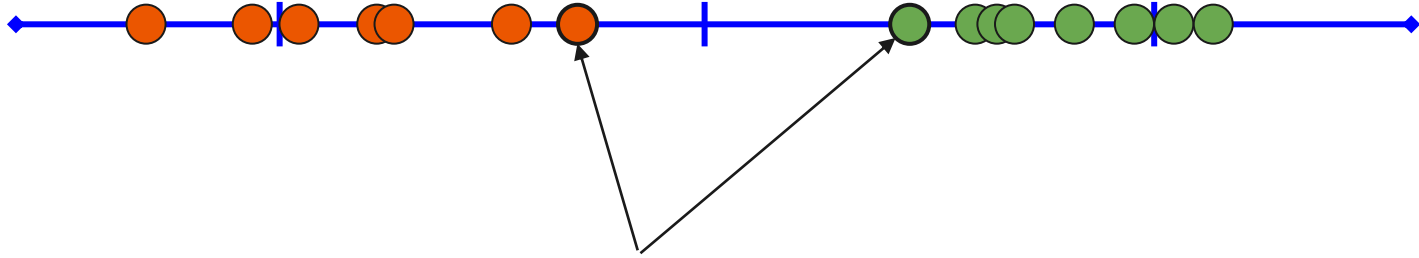
Este límite o umbral es malísimo

¿Cómo podemos mejorarlo?

Masa de un puñado de ratones

1. Volvemos al conjunto original

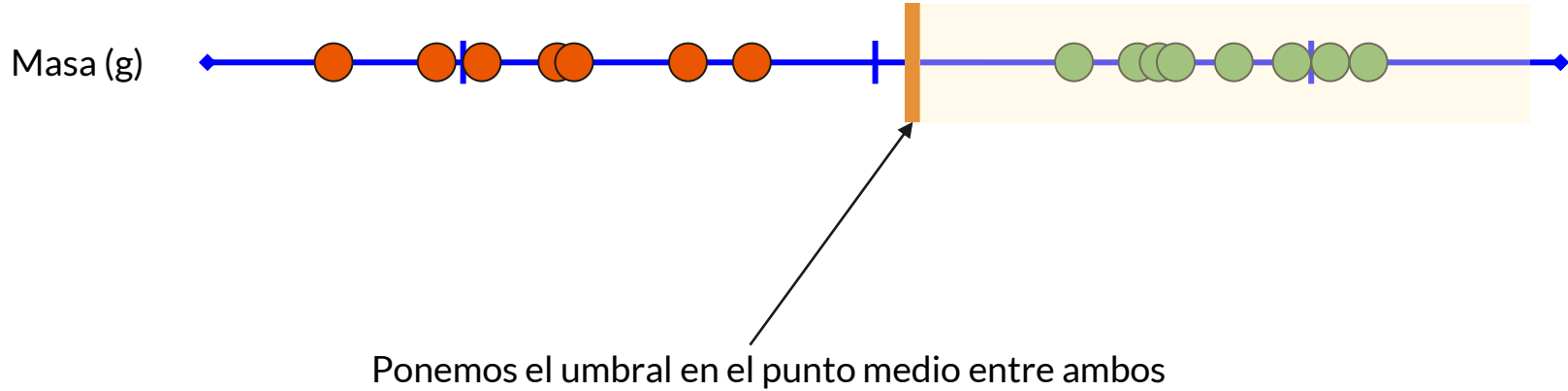
Masa (g)



Observamos los ratones en los límites de cada grupo

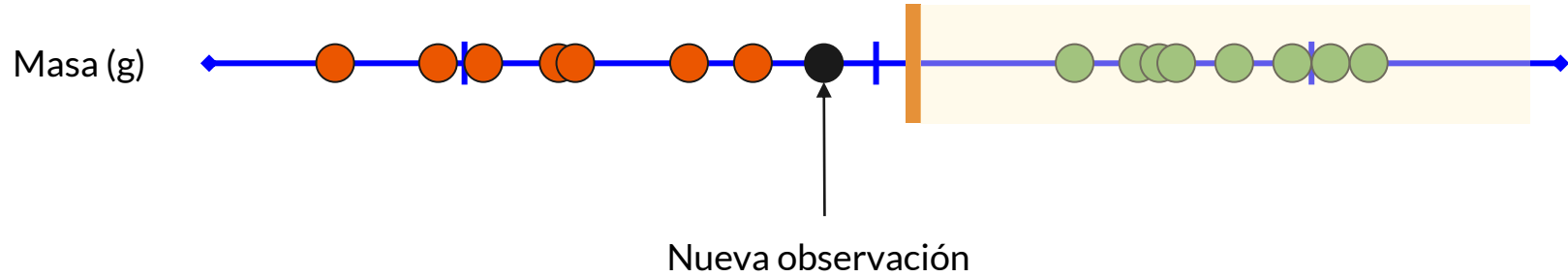


Masa de un puñado de ratones

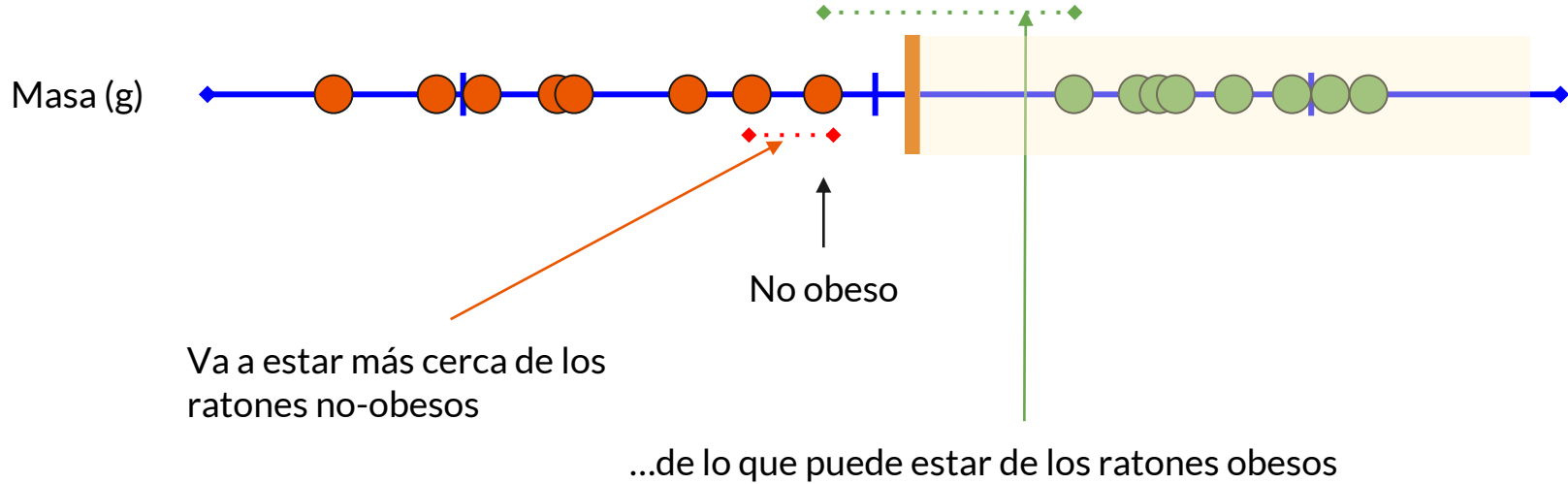




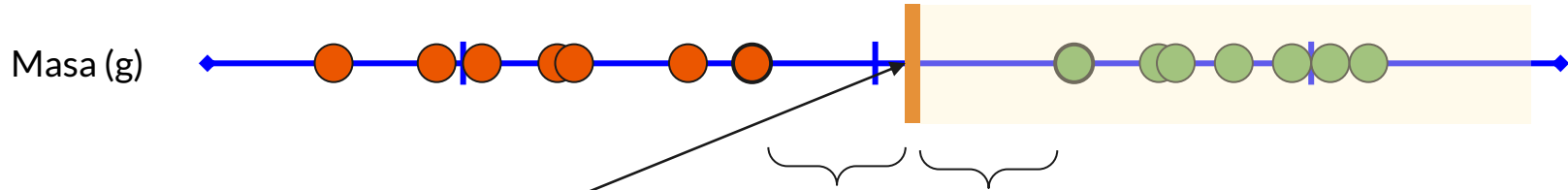
Masa de un puñado de ratones



Masa de un puñado de ratones



Masa de un puñado de ratones



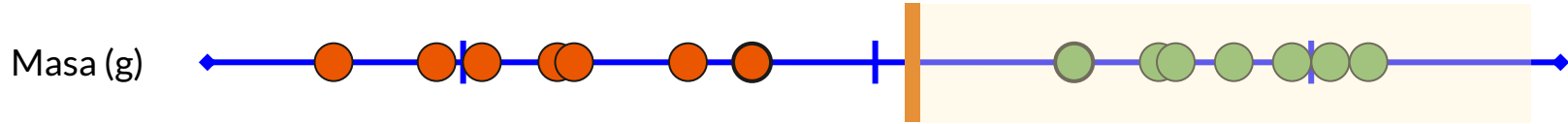
La distancia *mínima* entre cualquier observación y el umbral se llama: **margen**

Cómo pusimos el umbral en el medio de las dos observaciones límite, el **margen** es igual a ambos lados.

El margen es el más grande posible.

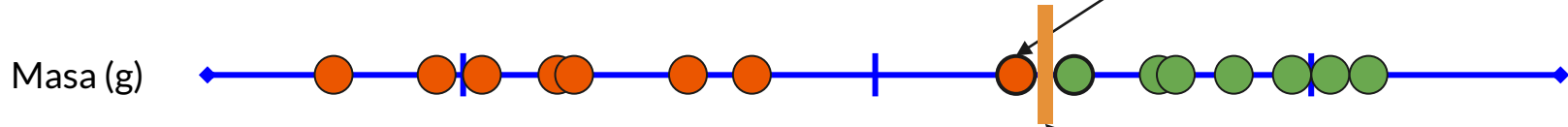


Masa de un puñado de ratones



Cómo el margen es el más grande posible nuestro clasificador es un clasificador llamado de Margen Máximo (**Maximal Margin Classifier**)

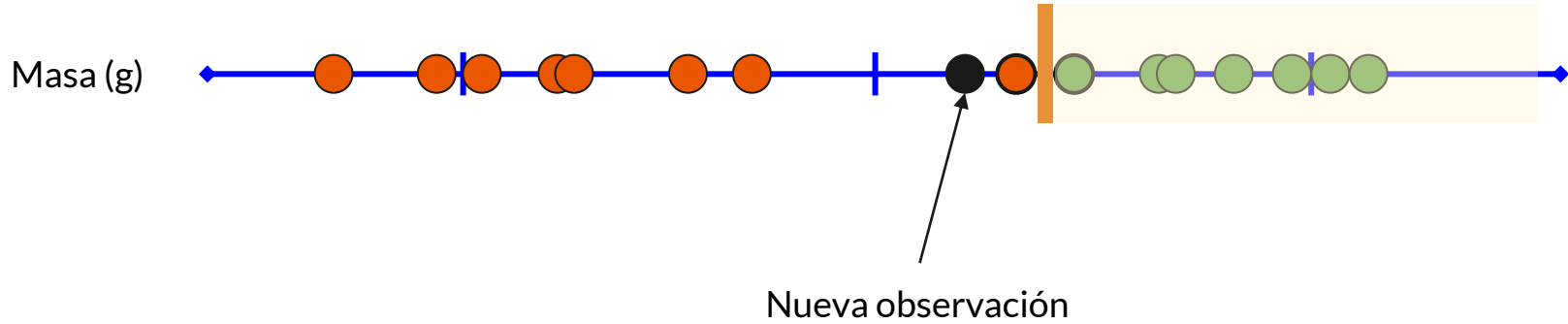
Masa de un puñado de ratones



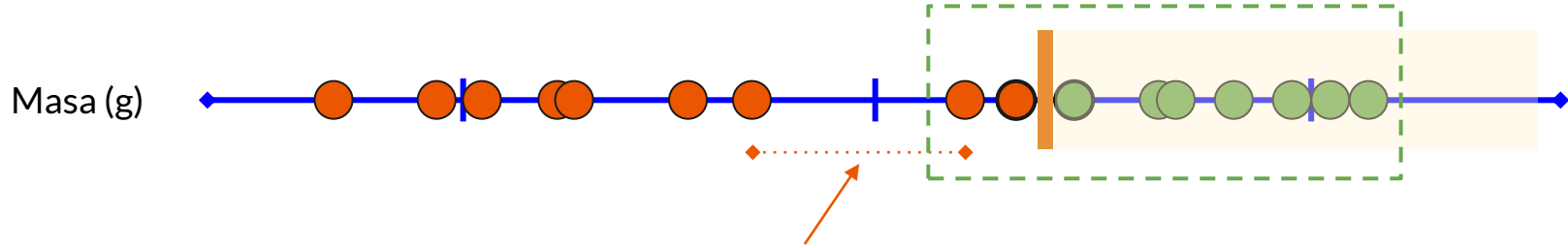
¿Qué pasaría si en nuestro conjunto de datos tuviésemos algo así?

Un modelo de clasificación de margen máximo nos obligaría a poner aquí el umbral

Masa de un puñado de ratones



Masa de un puñado de ratones

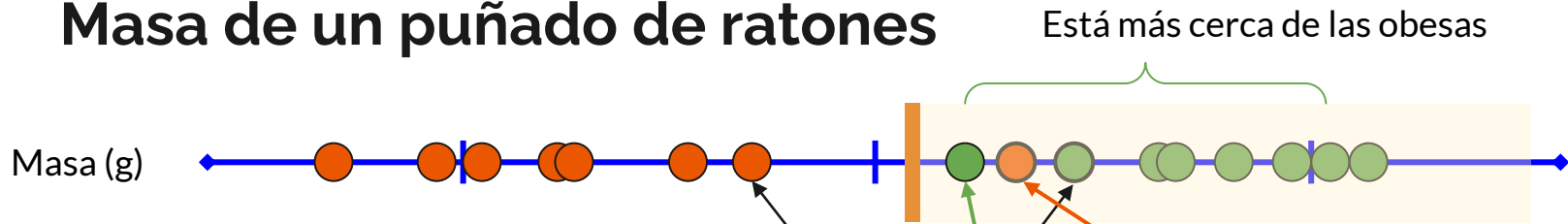


Está más lejos del resto de los ratones no-obesos
que de los obesos (sin contar al *outlier*)

Los clasificadores **Maximal Margin Classifier** son super sensibles a los outliers. Y en este caso va a provocar muchas clasificaciones erróneas

¿Cómo podemos mejorar?

Masa de un puñado de ratones



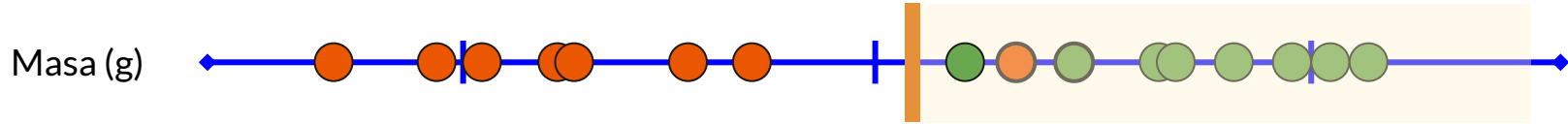
Si ponemos el umbral entre estas dos clasificaciones (como antes).

Esta de acá quedará **mal** clasificada

Para mejorar tenemos que permitir las clasificaciones erróneas.

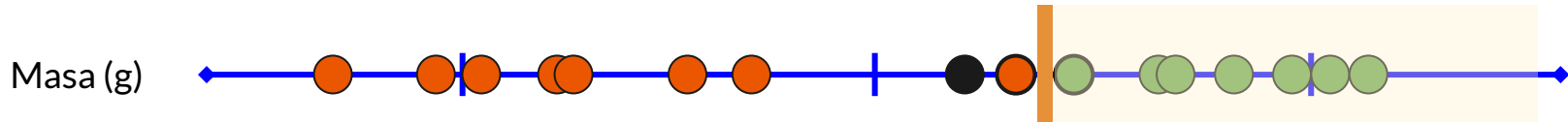
Pero, cuando aparezca una nueva observación acá la clasificará como obesa. Y tiene sentido.

Masa de un puñado de ratones



Umbral poco sensible al conjunto de entrenamiento = *higher bias*

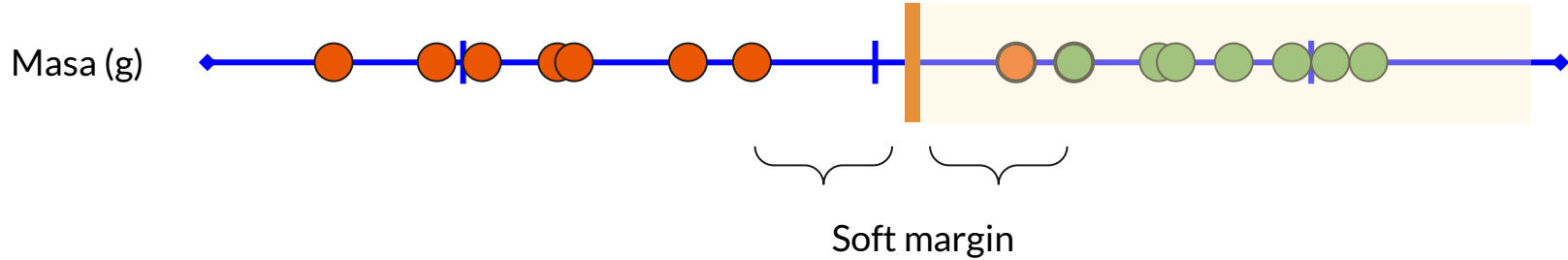
Desempeño bueno con nuevos datos = *low variance*



Umbral muy sensible al conjunto de entrenamiento = *low bias*

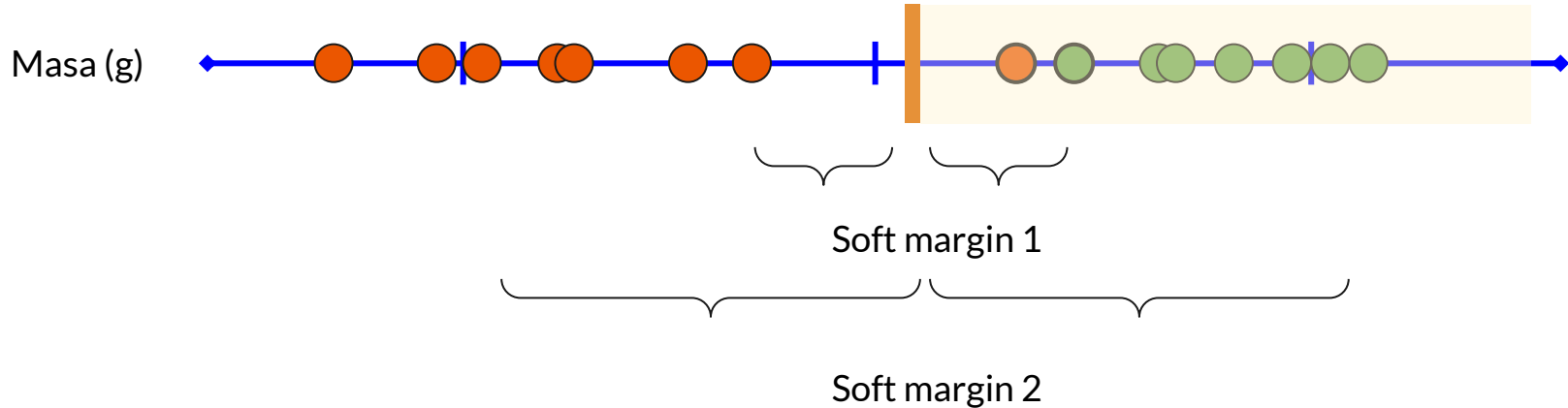
Desempeño pobre con nuevos datos = *high variance*

Masa de un puñado de ratones



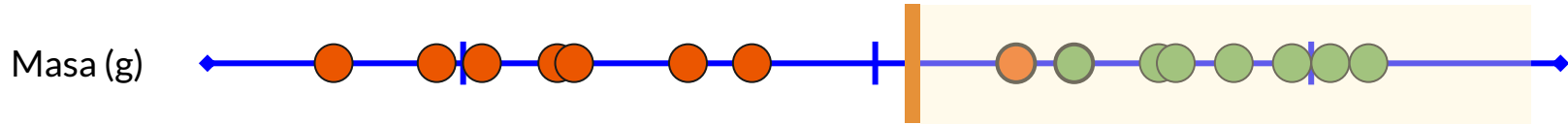
Cuando permitimos la clasificación errónea, la distancia entre las observaciones u el umbral se llama: **Soft margin**

Masa de un puñado de ratones



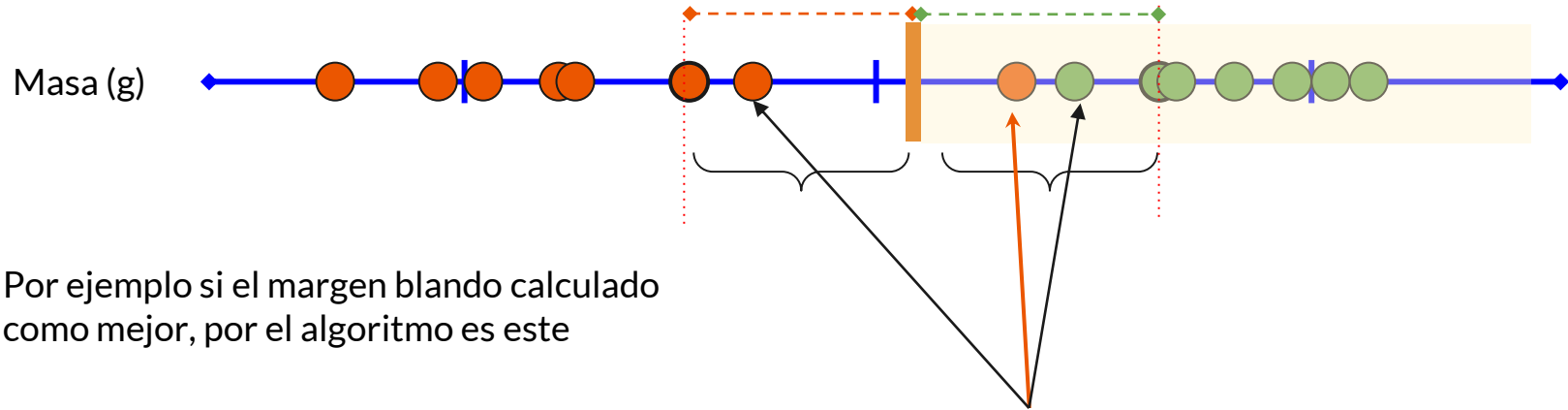
¿Cómo sabemos que Soft margin tomar? ¿Por qué el **Soft margin 1** es mejor que el **Soft margin 2**?

Masa de un puñado de ratones



- El Soft margin, será calculado por el algoritmo que usemos, utilizando validación cruzada.
 - Partirá el conjunto en dos (70% - 30%, o a veces en 10 subconjuntos)
 - Utilizará pares de observaciones similares para calcular la distancia media del umbral y validará la clasificación con el segundo conjunto.
 - Cómo sabe de antemano que va a soportar clasificaciones erróneas, tendrá que definir cuántas soportará en su margen blando y cuantas observaciones.

Masa de un puñado de ratones

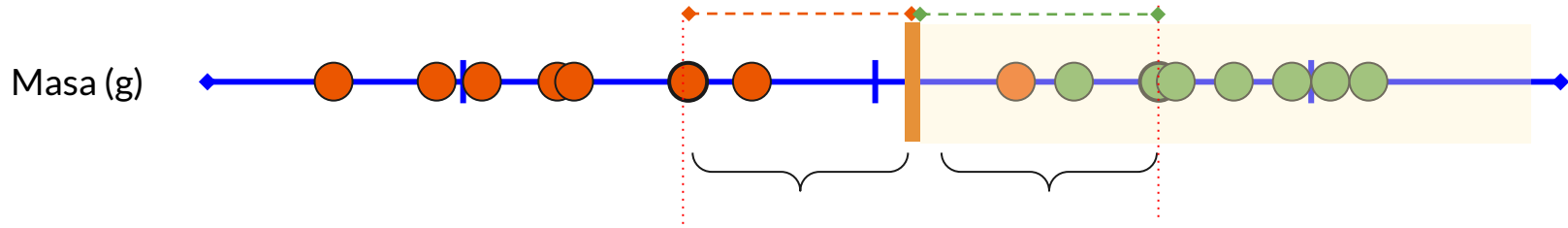


Por ejemplo si el margen blando calculado como mejor, por el algoritmo es este

Entonces el clasificador permite:

- 1 clasificación errónea
 - 2 observaciones correctamente clasificadas
- Dentro del **Soft Margin**

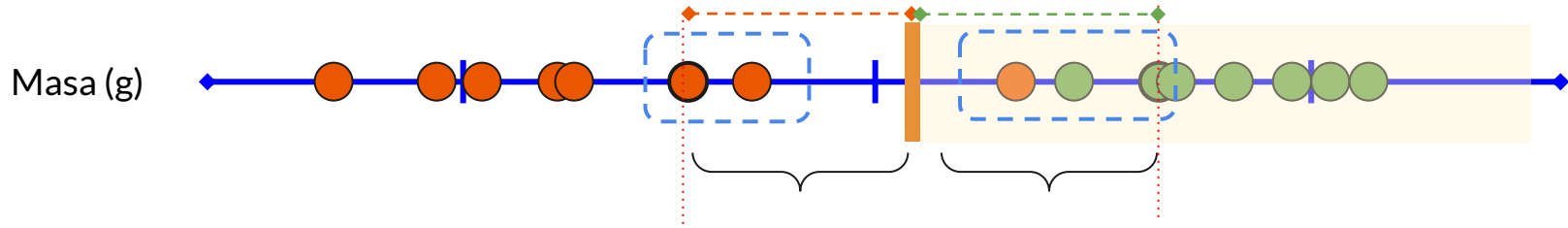
Masa de un puñado de ratones



Cuando usamos un margen blando (*Soft Margin*) para determinar la ubicación del umbral. Estamos utilizando un: **Soft Margin Classifier**

También conocido como: **Support Vector Classifier**

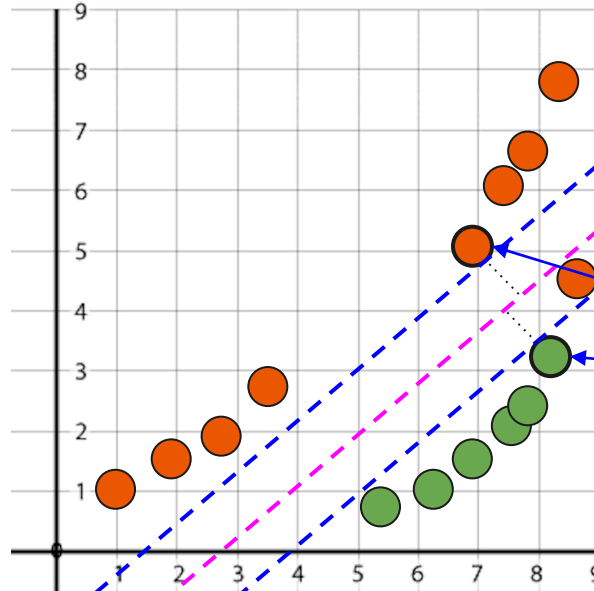
Masa de un puñado de ratones



Support Vector Classifier: Este nombre se debe a que las observaciones en los límites y dentro del *Soft Margin*, se llaman **Support Vectors** (o vectores soporte). Son vectores ya que son puntos en un espacio de dimensión N , en este caso $N=1$. Y soportan (o definen) el área llamada *Soft Margin*.

Masa y altura de un puñado de ratones

Altura (cm)



Masa (gr)

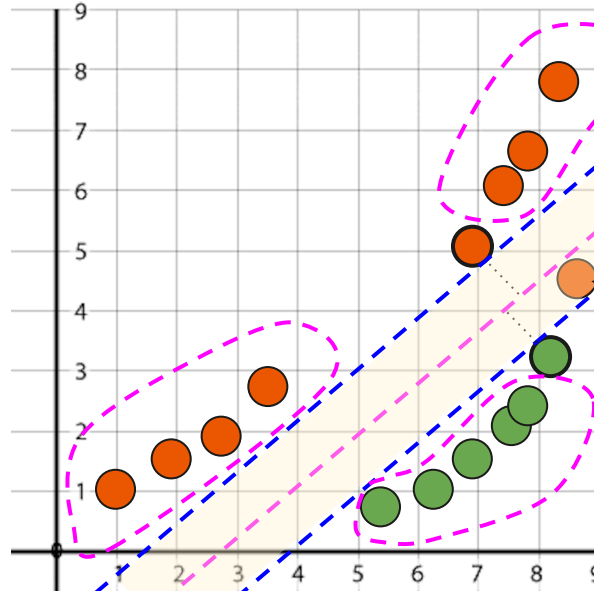
En 2D, el **Support Vector Classifier** es una línea.

El **Soft Margin** está dado por estos 2 puntos.

Masa y altura de un puñado de ratones

Altura (cm)

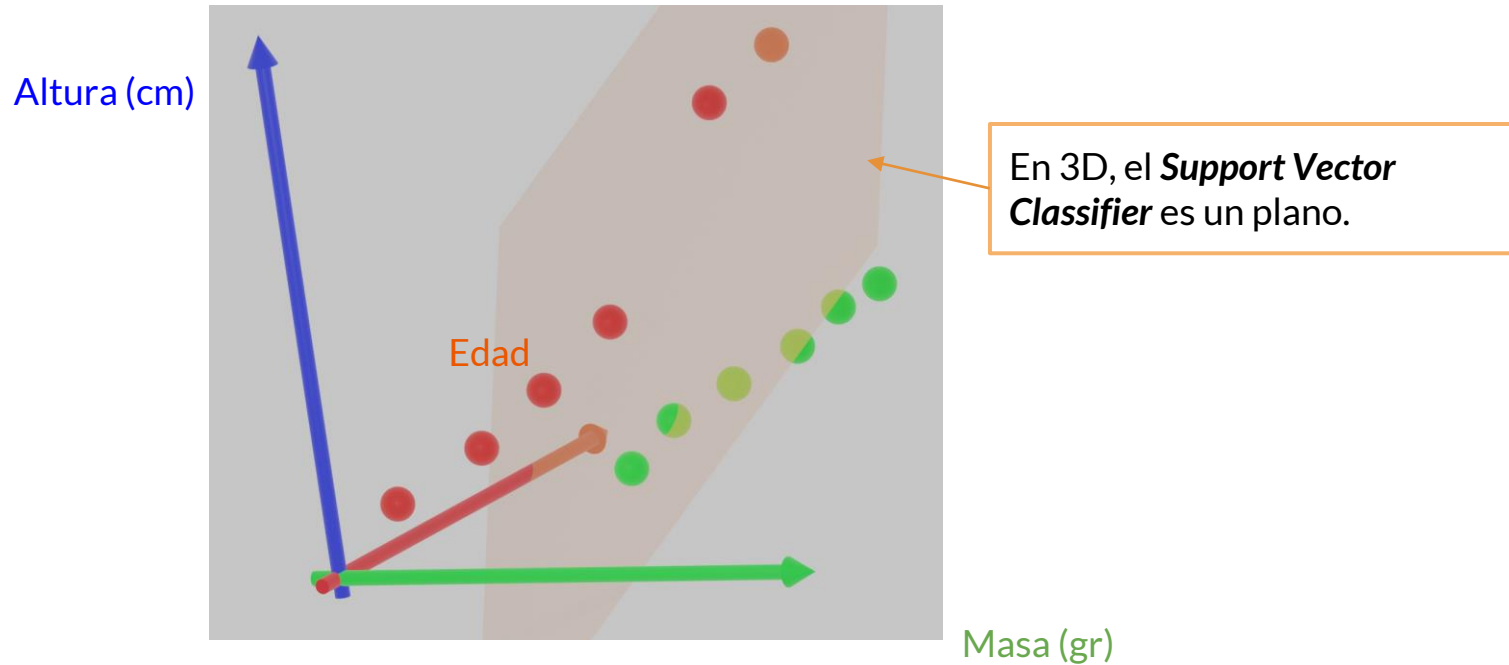
Estas observaciones están
fuera del **Soft Margin**



Esta observación está
dentro del **Soft Margin** y
clasificada erróneamente

Masa (gr)

Masa, altura y edad de un puñado de ratones

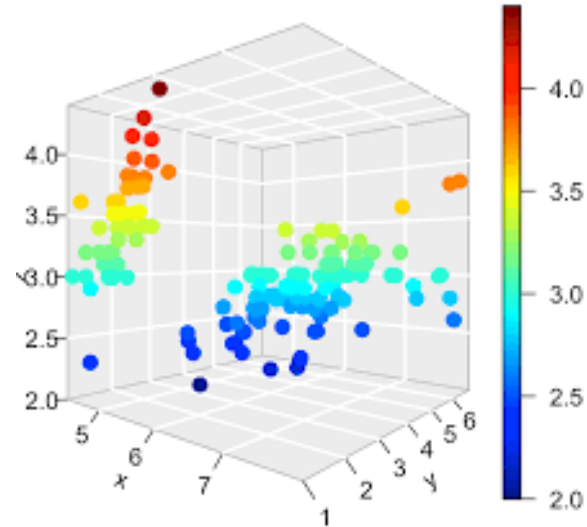


Masa, altura, presión sanguínea y edad de un puñado de ratones

4D: Graph

No es completamente imposible de graficar, pero si es difícil de interpretar

En colores la cuarta dimensión

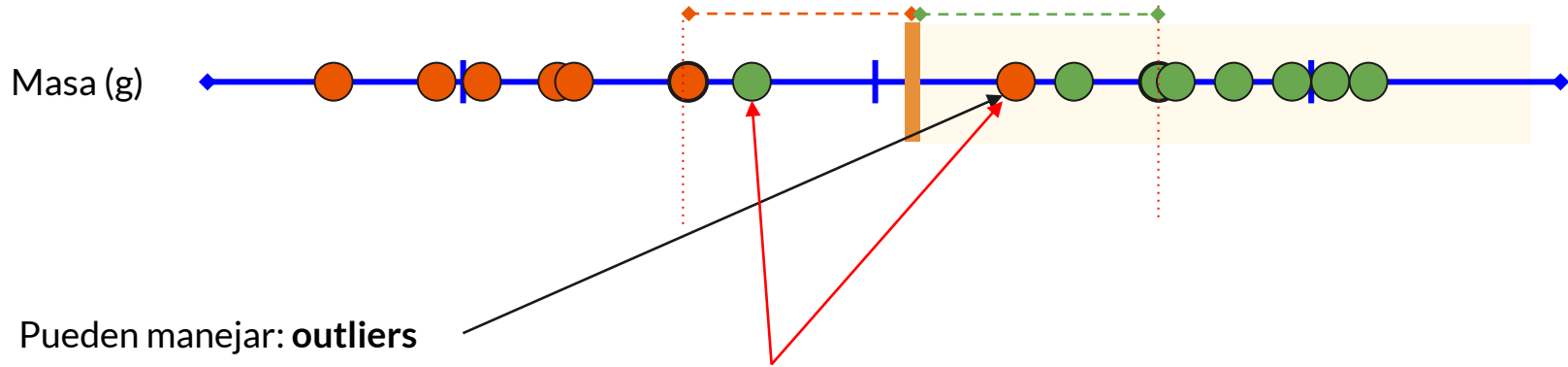




Observaciones N-dimensionales

Dimensionalidad de datos	<i>Support Vector Classifier</i>	Nombre
1D	<i>0 dimensiones</i>	punto
2D	<i>1 dimensión</i>	línea
3D	<i>2 dimensiones</i>	plano
4D	<i>3 dimensiones</i>	espacio
5D	<i>4 dimensiones</i>	hiper-plano
ND	<i>N-1 dimensiones</i>	hiper-plano

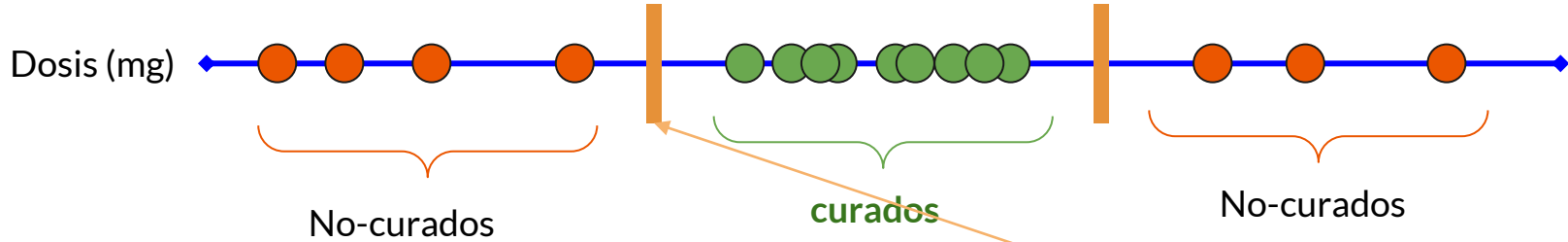
Support Vector Classifier



Pueden manejar: **outliers**

Y cómo pueden manejar clasificaciones erróneas, pueden manejar clasificaciones superpuestas

Dosis de una droga



¿Pero qué pasa si tengo esta distribución de datos?

No importa donde pongamos el umbral
Siempre tendremos errores groseros de
clasificación

Es un problema NO separable linealmente



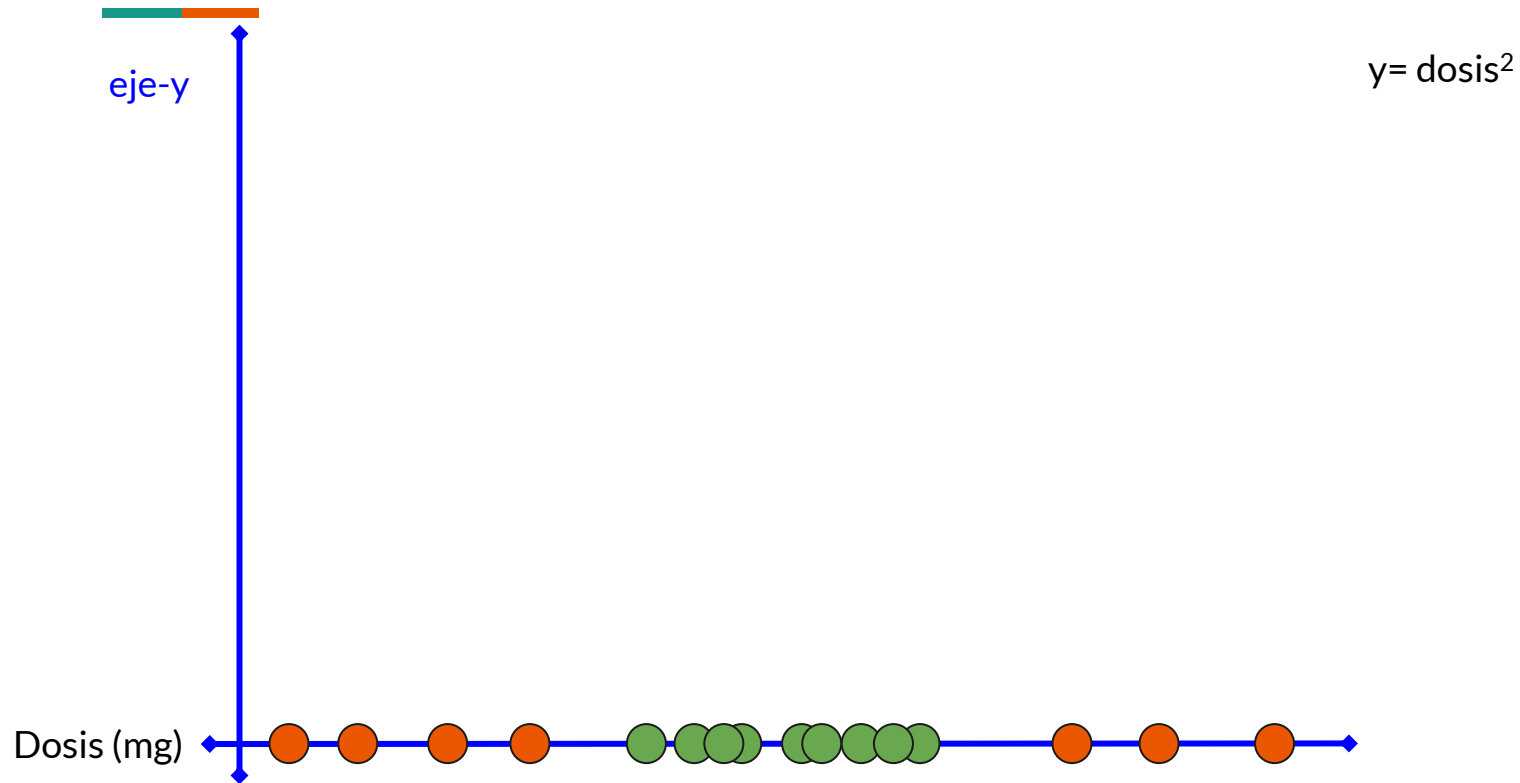
¿Cómo mejorar?

Los *Maximal Margin Classifiers* y los *Support Vector Classifiers* (clasificadores de márgenes blandos) no pueden clasificar problemas que no son linealmente separables.

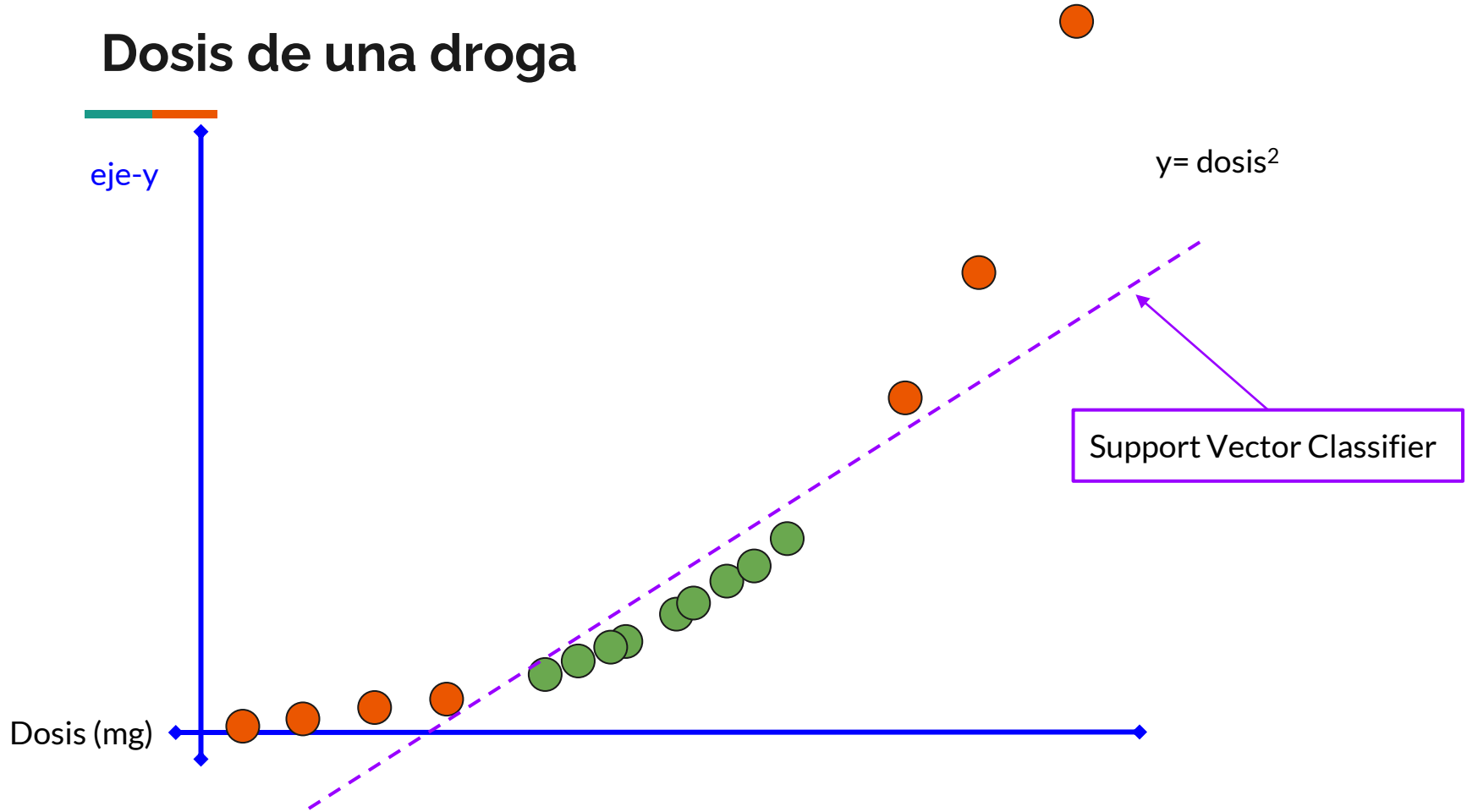
¿Cómo mejorar?

Support Vector Machines

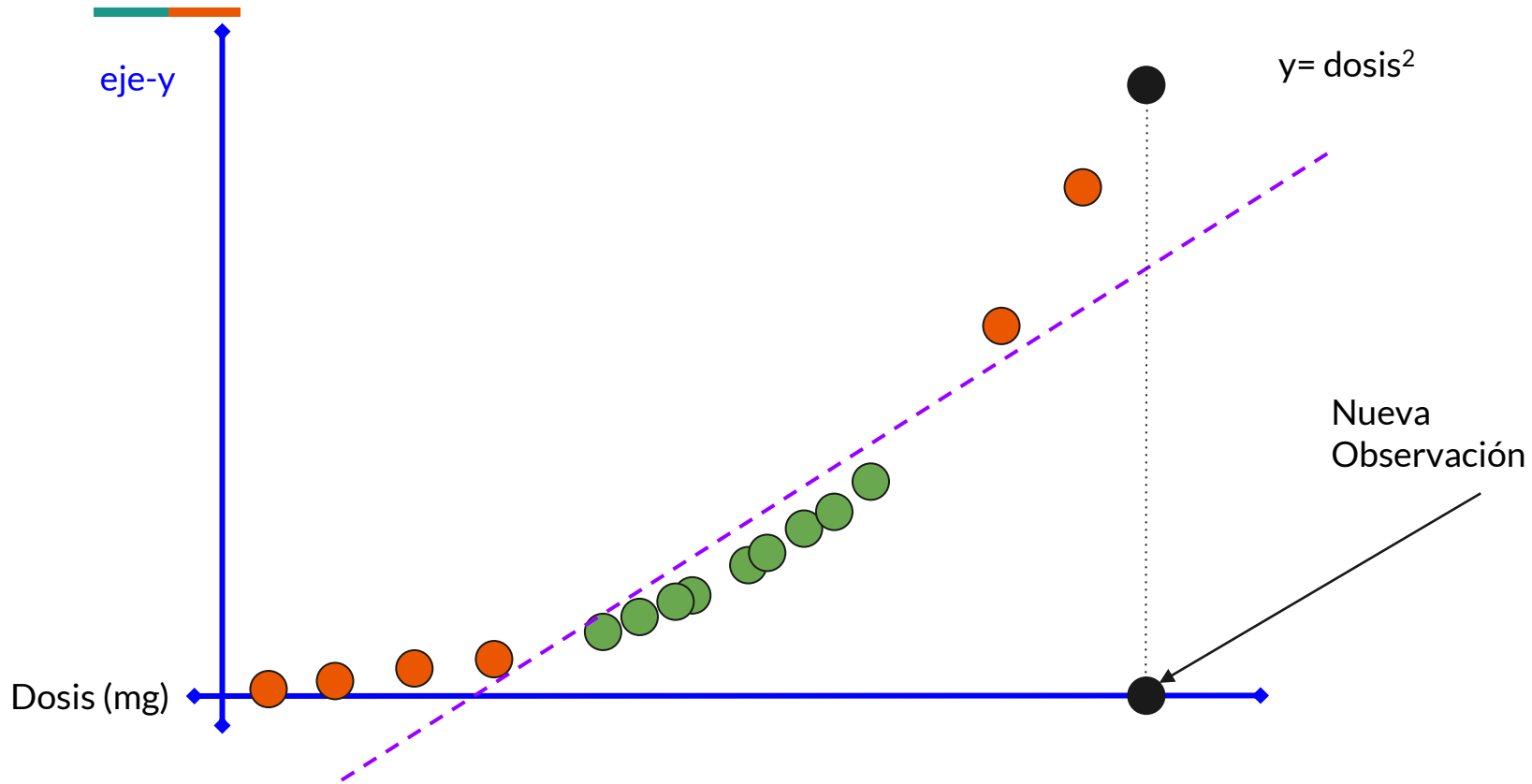
Dosis de una droga



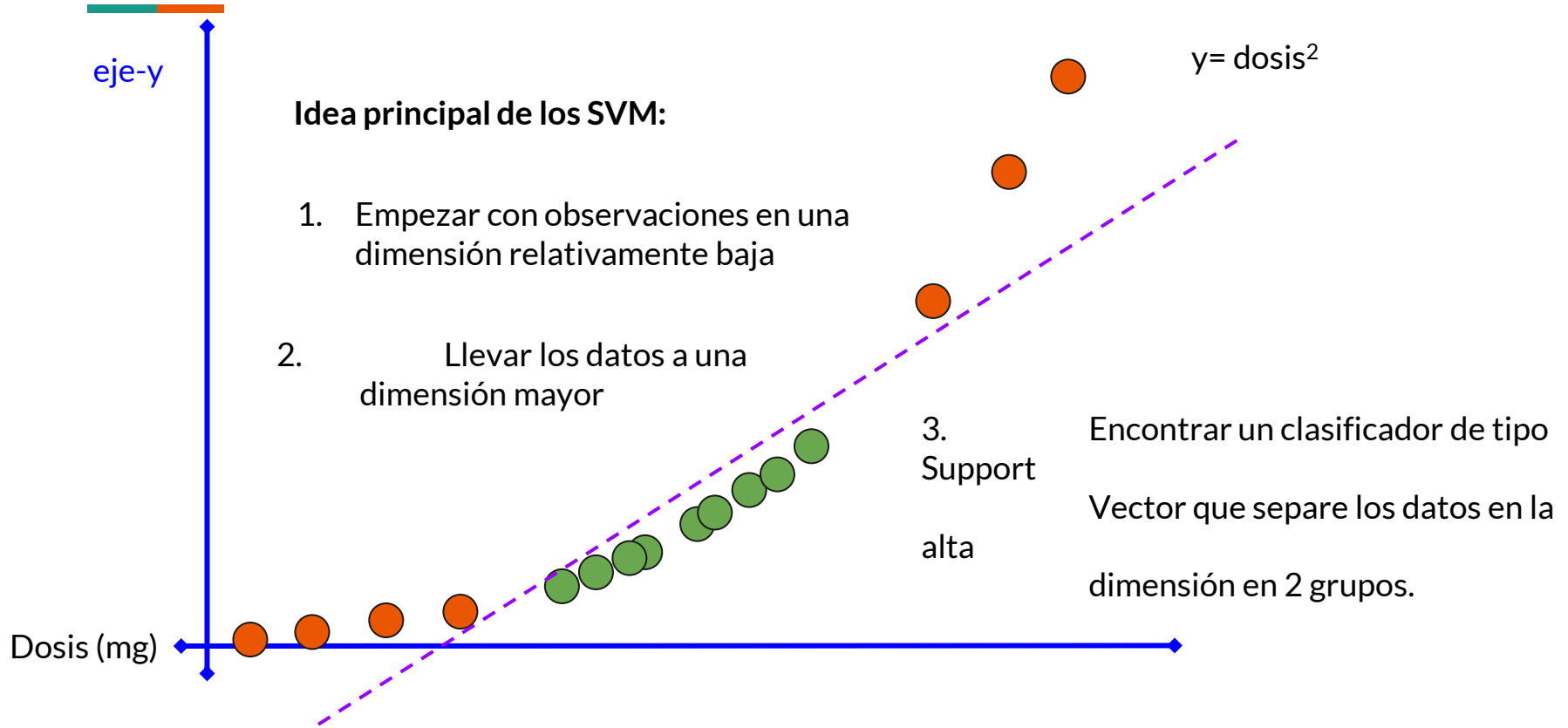
Dosis de una droga



Dosis de una droga



Dosis de una droga



Dosis de una droga

¿Por qué elegimos $y = x^2$?

El algoritmo detrás de **Support Vector Machines**, usa algo llamado **Funciones Kernel** que sistemáticamente buscan clasificadores de tipo **Support Vector Classifiers** en dimensiones más altas.

En el ejemplo anterior usamos un **Kernel** de tipo Polinómico, el cual toma un parámetro **d**, para el grado del polinomio

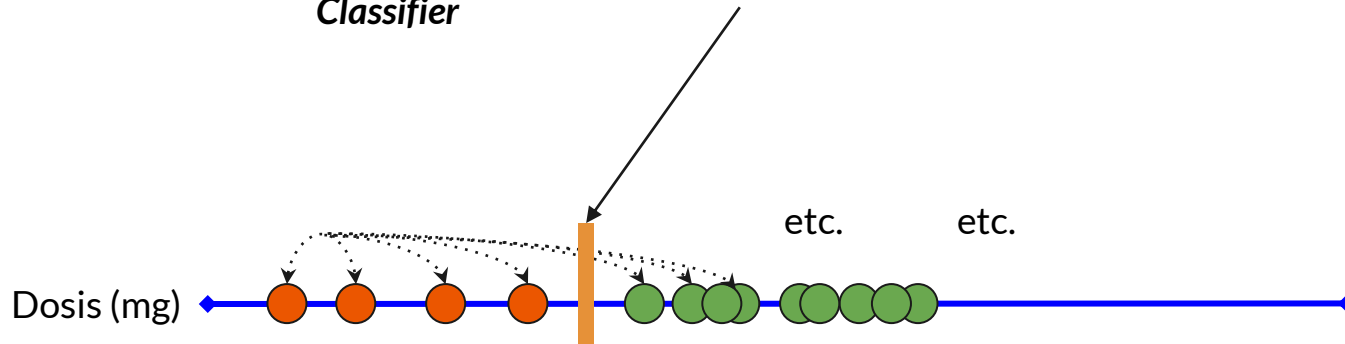


SVM Kernel polinomico

d=1

En 1 dimensión computa la relación entre cada par de observaciones, para encontrar....

Se utilizan para encontrar el **Support Vector Classifier**

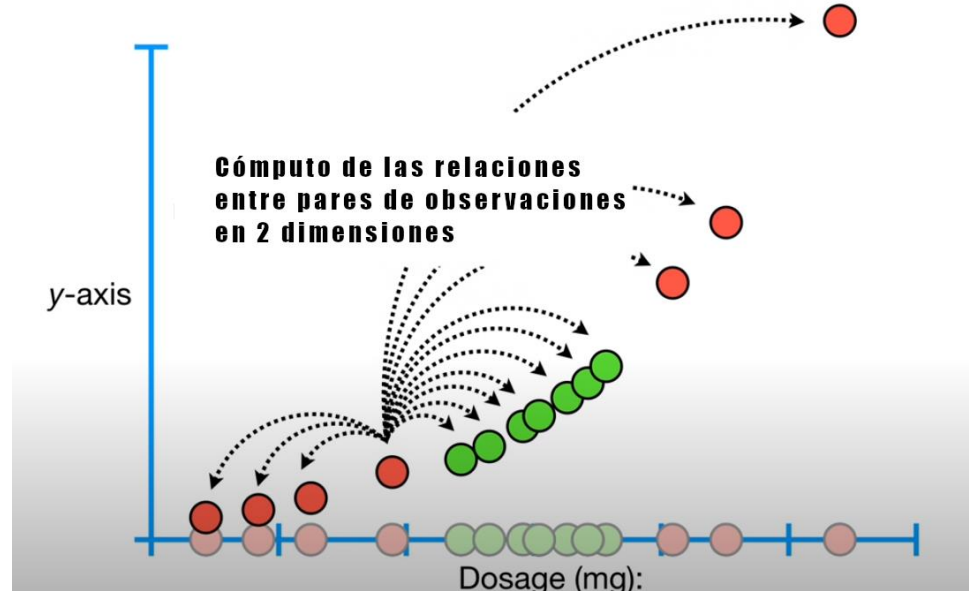


SVM Kernel polinomico

$d=2$

Con $d=2$ obtiene una segunda dimensión elevando al cuadrado las observaciones.

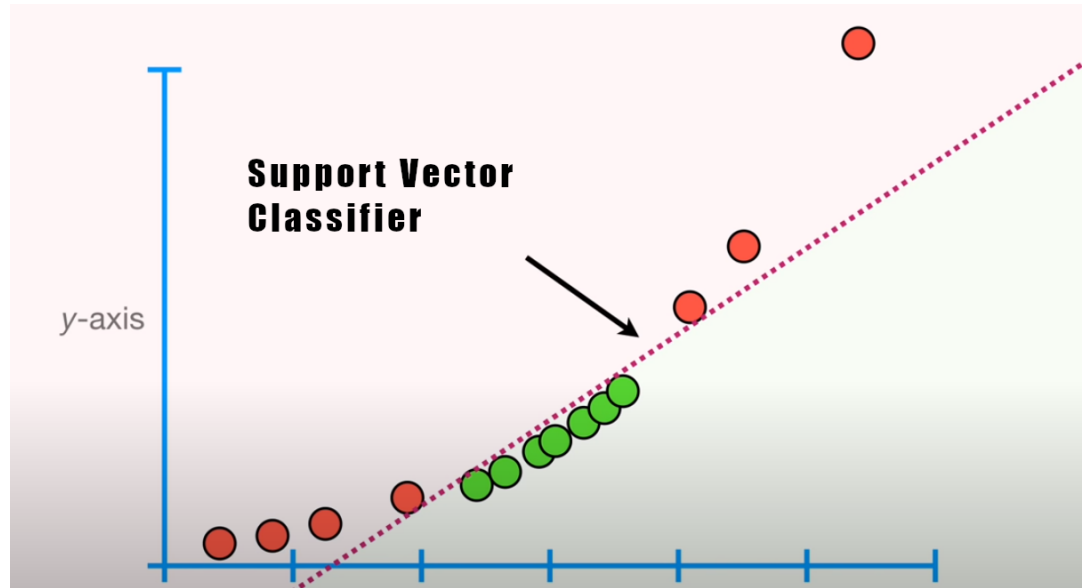
Y luego computa la relación entre cada par de observaciones....



SVM Kernel polinomico

$d=2$

Utilizando las relaciones calculadas
Calcula el clasificador: **Support Vector Classifier**



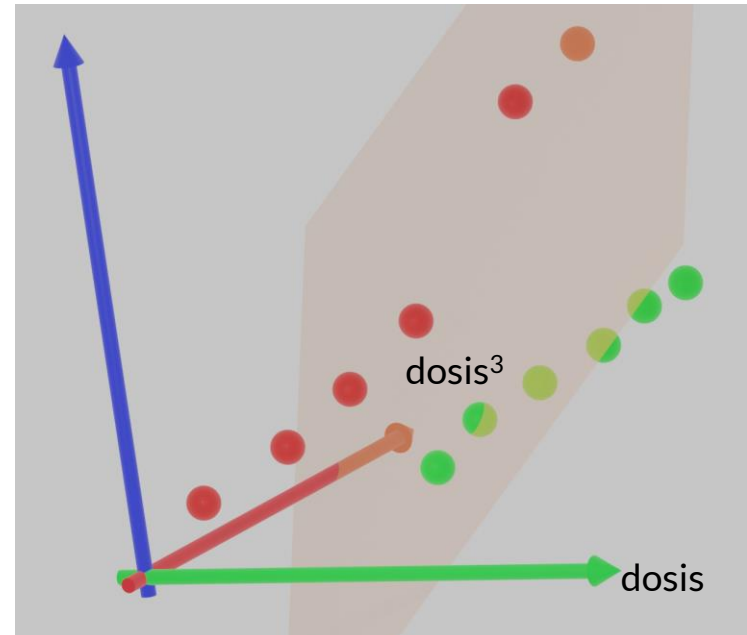
SVM Kernel polinomico

d=3

Agregamos a la que ya teníamos una nueva dimensión basada en x^3

Calcula el algoritmo nuevamente las relaciones entre todas las observaciones y busca un plano que parta el conjunto en 2.

$dosis^2$





SVM Kernel polinomico

Con $d=4$, repite el proceso agregando una dimensión adicional. Lo mismo para cualquier $d=n$.

En resumen:

1. El Kernel Polinómico de SVM sistemáticamente incrementa las dimensiones seteando d , desde 1 (o el valor base de las observaciones) hasta el valor del parámetro.
2. Calcula en cada caso las relaciones entre las observaciones para encontrar un *hiper-plano* que parta el conjunto en 2.
3. Usando **validación cruzada**, en cada caso calcula el error y se queda con el mejor d .

SVM Kernel Radial (Radial Basis Function Kernel)

Soporta **Support Vector Classifiers** en infinitas dimensiones

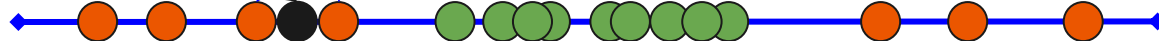


Nueva
observación

Simplificando: RBFK funciona como un *Nearest Neighbor* model, (vecinos más cercanos) ponderado.

Utiliza a las observaciones más cercanas (vecinas),
para clasificar la nueva observación

Dosis (mg)

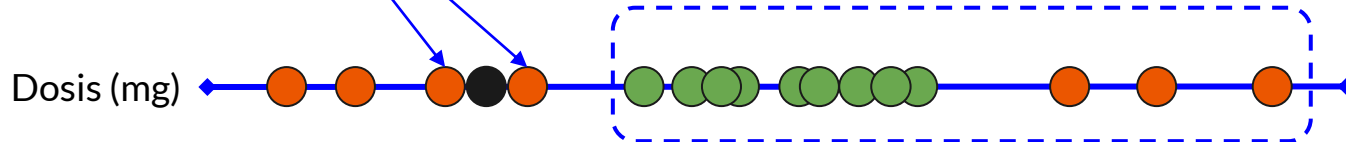


SVM Kernel Radial (Radial Basis Function Kernel)

Soporta **Support Vector Classifiers** en infinitas dimensiones

El **Kernel Radial** usa la clasificación de las cercanas para la nueva observación

Las clasificaciones más lejanas tienen poca influencia en la nueva





SVM: The Kernel Trick

Las *Kernel Functions*, sólo calculan la relación entre pares de observaciones como si estuviesen en otra dimensión, pero no realizan una verdadera transformación del espacio.

Este truco del Kernel, reduce la cantidad de tiempo de cómputo necesario ya que evita toda la matemática relacionada a la transformación del espacio.

Además permite que se realicen los cálculos del Kernel Radial en infinitas dimensiones.