SVMs: Máquinas de Vectores de Soporte (Support Vector Machines)

Andrés Villa

Computer Science Department, PUC

- SVMs corresponden a una técnica de aprendizaje supervisado.
- Por tanto, necesitan de un set de datos de entrenamiento previamente rotulado.
- SVMs son una de las técnicas de clasificación y predicción más precisas entre los modelos no jerárquicos.
- Como veremos más adelante, una de las claves de su buen rendimiento es su estrategia de entrenamiento discriminativa que maximiza el margen de clasificación.
- Si bien es posible utilizar SVMs para regresión (predicción), en este curso nos concentraremos en su uso como clasificador.

SVM: Separadores lineales para clasificación binaria

Un clasificador lineal puede ser representado por:

$$g(x) = w_1 x + w_0$$

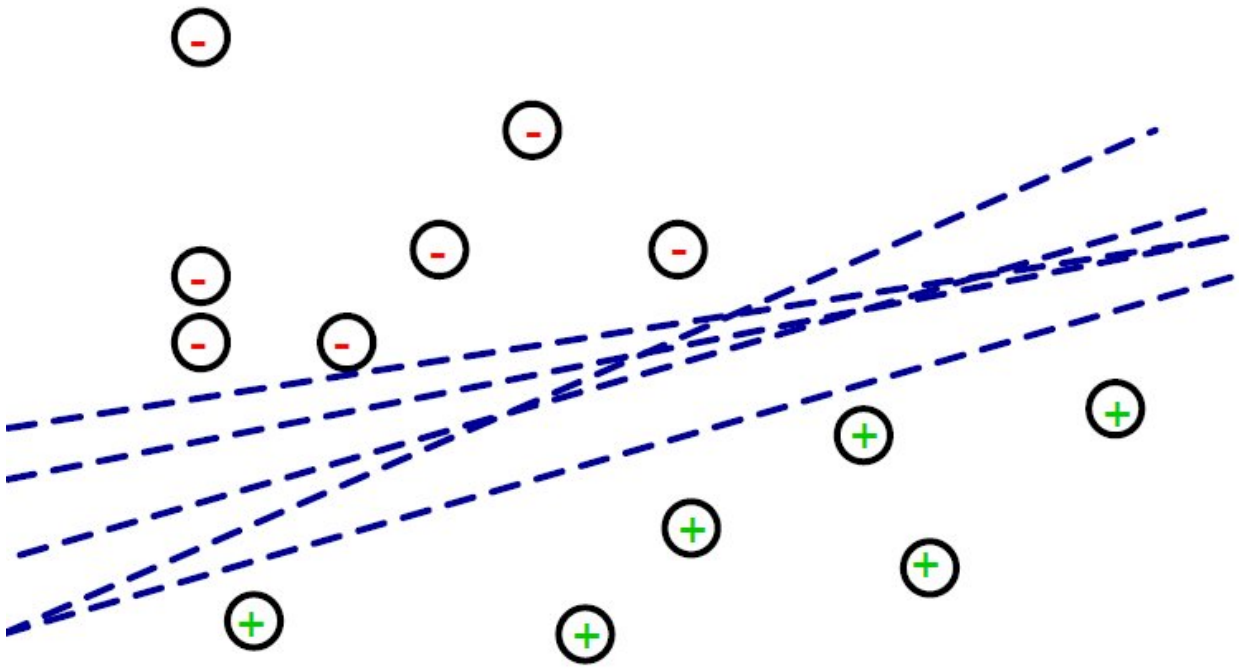
Para el caso binario, la regla de decisión es:

$$Class(x) = \begin{cases} C1, & \text{if } g(x) > 0 \\ C2, & \text{if } g(x) \leq 0 \end{cases}$$

Por tanto, la frontera de decisión es :

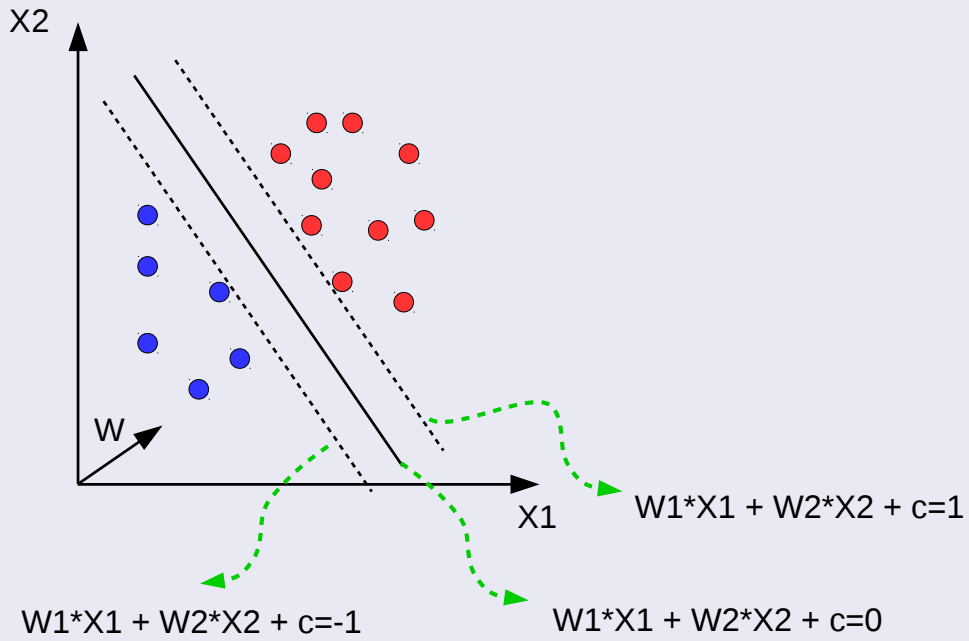
$$g(x) = 0.$$

¿Cuál es el mejor hiperplano para separar las clases?

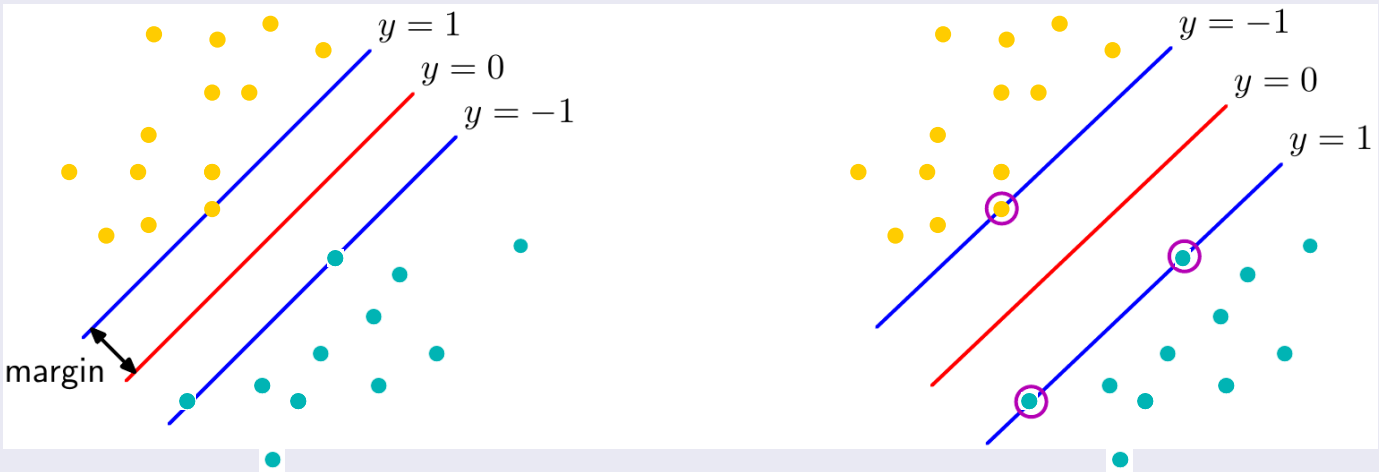


Estrategia de clasificación de máximo margen

- Si se toma un plano muy cerca a las clases, esto puede resultar en una buena clasificación en el conjunto de entrenamiento pero en una mala generalización. El mejor plano para clasificar las 2 clases es el que maximiza el margen de separación entre ellas.
- Donde el margen de separación es definido como la distancia perpendicular entre el plano (superficie de decisión) y los registros más cercanos a cada uno de sus lados.
- Como veremos más adelante, para facilitar los cálculos matemáticos el mínimo margen es escalado al valor 1.



Estrategia de clasificación de máximo margen



- La posición del plano es definida por un reducido conjunto de los datos de entrenamiento. Estos registros o vectores son los que dan soporte a la superficie de decisión, de ahí el nombre del método.
- Dado que estos registros son los más cercanos a la superficie de decisión, son también los más difíciles de clasificar, por qué?.
- La definición de este plano implica 2 condiciones fundamentales:
  1. Los datos de entrenamiento deben ser bien clasificados (¿por qué sólo los datos de entrenamiento?),
  2. Se debe maximizar el margen respecto a los registros más cercanos al plano (¿por qué los más cercanos?).

Estrategia de clasificación binaria de máximo margen

Condición 1: hiperplano $g(x)$ que clasifique correctamente los registros de cada clase [1]:

$$g(x_k) = w\ x_k + c \begin{cases} > 0, & \text{if } x_k \in C_1 \\ \leq 0, & \text{if } x_k \in C_2 \end{cases}$$

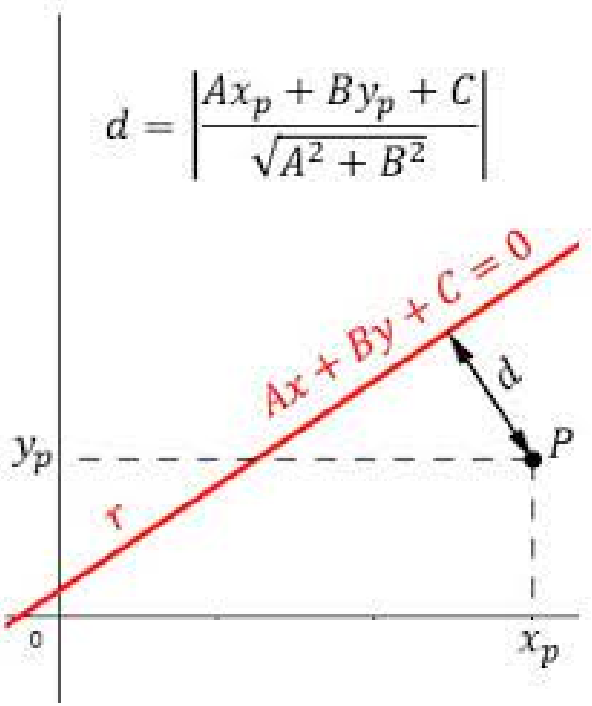
- Esta expresión es compleja de tratar matemáticamente (¿por qué?). Sin embargo, con un pequeño truco podemos juntar las desigualdades.
- Definamos la variable auxiliar $z_k = \pm 1$, dependiendo si la instancia x_k pertenece a C_1 or C_2, respectivamente.
- El uso de z_k nos permite unir las desigualdades anteriores en una sola expresión:

$$z_k(w\ x_k + c) > 0, k = 1 \ldots n$$

[1] Por ahora vamos a asumir caso binario y que los registros son linealmente separables

Estrategia de clasificación binaria de máximo margen

Condición 2: hiperplano $g(x)$ que maximiza margen a registros más cercanos a superficie de decisión. Necesitamos calcular distancia de un punto P a un plano:



Considerando el plano $Ax + By + C = 0$ y el punto $P(x_p, y_p)$, la distancia entre ellos es:

$$d = \frac{|Ax_p + By_p + C|}{\sqrt{A^2 + B^2}}$$

Si $g(x, y) = Ax + By + C$ y $w = [A, B]$,

$$d = \frac{|g(x_p, y_p)|}{\|w\|}$$

Estrategia de clasificación binaria de máximo margen

Condición 2: hiperplano $g(x)$ que maximize margen a registros más cercanos a superficie de decisión.

- La distancia de un punto x_k al hiperplano $g(x) = w\ x + c$ está dada por: $|g(x_k)|/||w||$
- Esta distancia corresponde al margen que queremos maximizar, pero no respecto a todos los registros. ¿A cuáles?
- Considerando que la correcta clasificación de cada instancia x_k garantiza que $z_k g(x_k) > 0$, nuestro problema se reduce al siguiente:

$$\underset{w,c}{\operatorname{argmax}} \left\{ \frac{1}{||w||} \min_k \{ z_k g(x_k) \} \right\}$$

Estrategia de clasificación binaria de máximo margen

Finalmente, las 2 condiciones anteriores se traducen en el siguiente problema de optimización:

$$\underset{w,c}{\operatorname{argmax}} \left\{ \frac{1}{||w||} \min_k \{z_k g(x_k)\} \right\}$$
$$\text{sujeto a: } z_k(w\ x_k + c) > 0, \quad k = 1 \ldots n$$

- Condición 1: El set de desigualdades garantiza que cada registro es bien clasificado, i.e., está en el lado correcto del hiperplano (esto asume que los datos son linealmente separables ¿por qué?).
- Condición 2: La función objetivo garantiza que obtenemos el hiperplano que maximiza el margen deseado.

Estrategia de clasificación binaria de máximo margen

- Es posible escalar el margen mínimo a un valor de 1, con lo cual se obtiene una expresión más sencilla de optimizar.
- La observación clave es que re-escalar la ecuación del plano, $g(x) = wx + c$, por una constante no afecta la distancia de cada punto al plano (recordar que distancia es $|g(x_k)|/||w||$).
- Así, un SVM re-escala w y c para satisfacer mínimo margen equal to 1, i.e., $\min_k\{z_k g(x_k)\} = 1$
- Consecuentemente, el margen para cualquier otro punto debe ser mayor o igual a 1 (por qué?). Es decir, $z_k g(x_k) = z_k(w\ x_k + c) \geq 1$
- Por tanto, todas las instancias de entrenamiento deben satisfacer:

$$z_k g(x_k) \geq 1;\ k = 1, \ldots, n.$$

- Con lo cual nuestro problema de optimización se transforma en :

$$\underset{w,c}{\text{argmax}} \left\{ \frac{1}{||w||} \min_k\{z_k g(x_k)\} \right\} \longrightarrow \underset{w,c}{\text{argmax}} \frac{1}{||w||}$$

sujeto a: $z_k(w\ x_k + c) > 0, \quad k = 1 \ldots n$ sujeto a: $z_k(wx_k + c) > 1, \quad \forall k \in TS$

Estrategia de clasificación binaria de máximo margen

Finalmente, las 2 condiciones anteriores se traducen en el siguiente problema de optimización:

$$\underset{w,c}{\text{argmax}} \ \frac{1}{||w||}$$

$$\text{sujeto a: } z_k(wx_k + c) > 1, \quad \forall k \in TS$$

- Condición 1: El set de desigualdades garantiza que cada registro es bien clasificado, i.e., está en el lado correcto del hiperplano (esto asume que los datos son linealmente separables ¿por qué?).
- Condición 2: La función objetivo garantiza que obtenemos el hiperplano que maximiza el margen deseado.

Estrategia de clasificación binaria de máximo margen

$$\underset{w,c}{\text{argmax}} \frac{1}{||w||}$$

$$\text{sujeto a: } z_k(w\ x_k + c) \geq 1, k = 1 \ldots n.$$

En el contexto de SVMs esto se suele escribir como:

$$\underset{w,c}{\text{argmin}} \frac{1}{2}||w||^2$$

$$\text{sujeto a: } z_k(w\ x_k + c) \geq 1, k = 1 \ldots n.$$

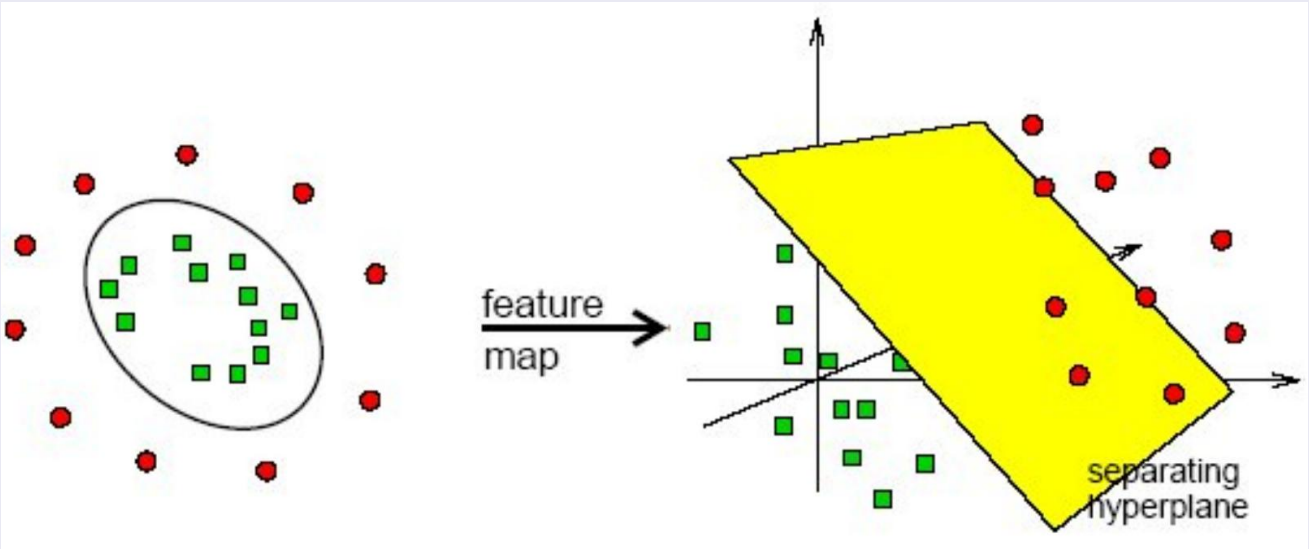
¿Cómo podemos resolver este problema de optimización?

- Problema de optimización trivial, función objetivo cuadrática con restricciones lineales.
- Solución utilizando el método de Lagrange y condiciones de Karush-Kuhn-Tucker (KKT). Los interesados en los detalles del método pueden consultar el texto de Bishop (ver bibliografía en programa del curso).

SVM: Caso no lineal

Idea

- Mapear datos a espacio de mayor dimensionalidad donde los datos son separables linealmente.



Ejemplo: Aumento dimensionalidad

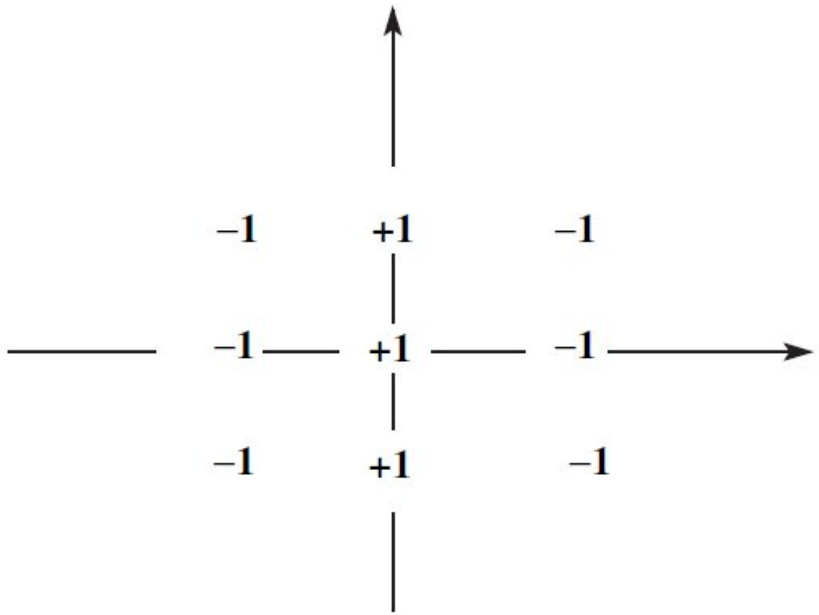


Figure: Datos en espacio original no son linealmente separables.

Ejemplo: Aumento dimensionalidad

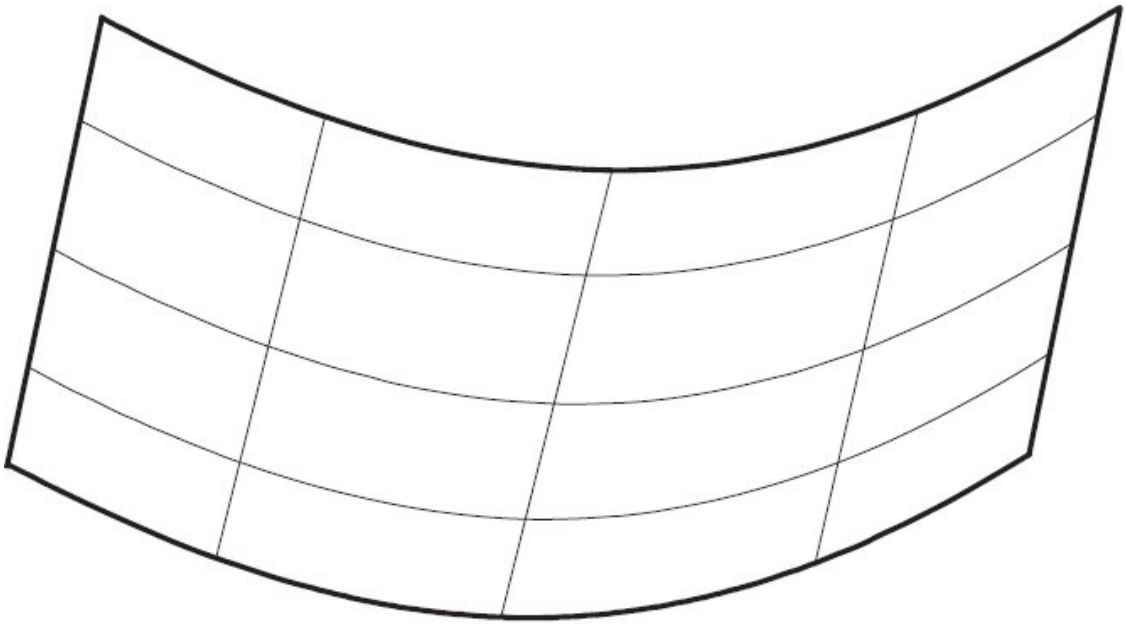


Figure: Espacio de características $f(x_1, x_2) = x_1^2$

Ejemplo: Aumento dimensionalidad

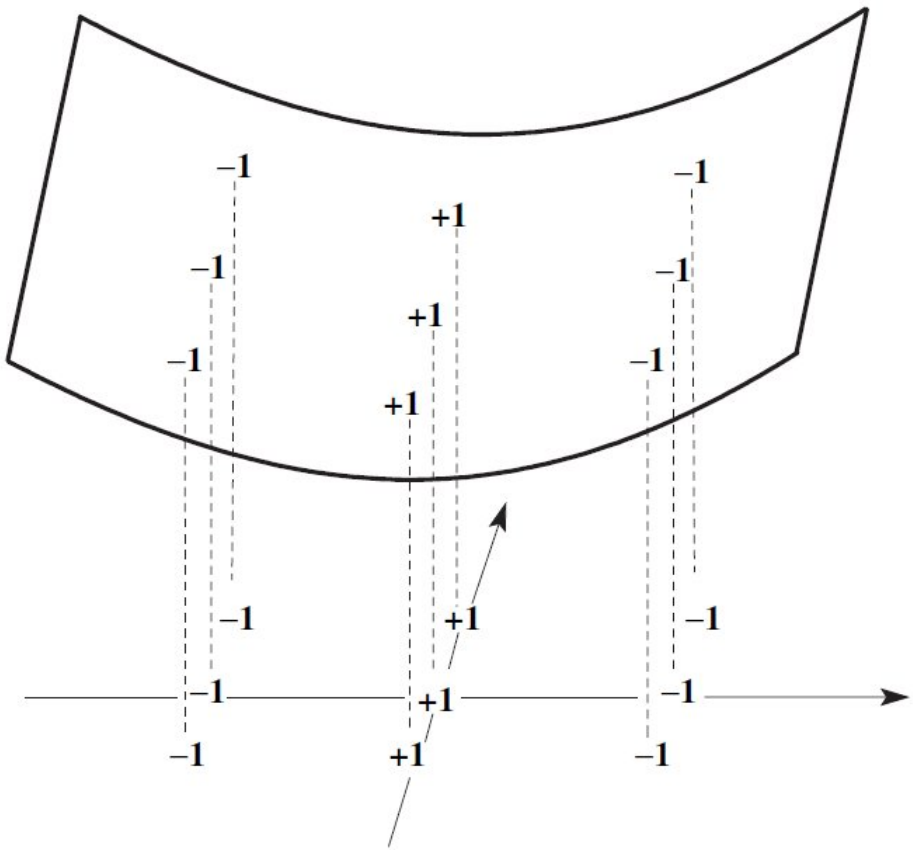


Figure: Al projectar los registros al nuevo espacio de características (x_1, x_2, x_1^2) el problema queda linealmente separable.

Ejemplo: Aumento dimensionalidad

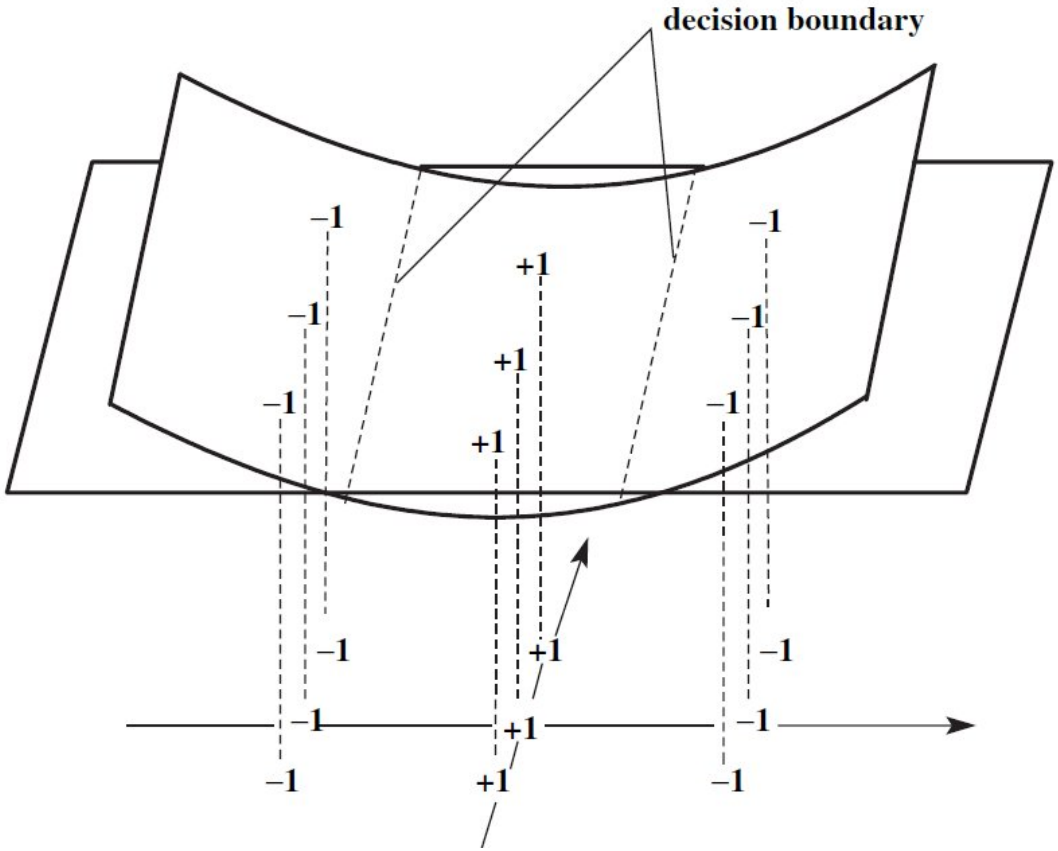


Figure: Plano separa en forma perfecta los registros de clase +1 (abajo del plano) de los registros de clase -1 (sobre el plano).

Nuevo problema en feature space

$$g(x_k) = w_1\phi(x_k) + w_0 \begin{cases} > 0, & \text{if } \phi(x_k) \in C_1 \\ \leq 0, & \text{if } \phi(x_k) \in C_2 \end{cases} \quad (1)$$

Lo cual genera un problema de optimización similar al visto anteriormente pero en nuevo espacio de características:

$$\arg\min_{w_1, w_0} \frac{1}{2}||w_1||^2;$$

$$\text{subject to: } z_k(w_1\phi(x_k) + w_0) \geq 1, k = 1 \ldots n.$$

Aún tenemos 3 problemas o limitaciones importantes

- Cuales serían ?

Aún tenemos 3 problemas o limitaciones importantes

- Cuales serían ?

1. ¿Cómo garantizamos que los datos sean linealmente separables en nuevo espacio?
2. ¿Cómo encontramos una buena función de Kernel para realizar la transformación?
3. ¿Cómo podemos aplicar SVMs a problemas de más de 2 clases?

SVM: Datos de entrenamiento no son linealmente separables en el espacio original o bajo alguna transformación

- En este caso el problema de optimización no tiene solución (¿por qué?).
- Afortunadamente podemos hacer un truco para permitir una solución de compromiso.
- Introducimos las variables auxiliares (slack variables) ξ_k que permiten que un registro este en el lado equivocado de la superficie de decisión.
- Estas variables ξ_k absorven la diferencia para que la desigualdad correspondiente siga respetanto la condición de mínimo margen: $\xi_k = |z_k - g(x_k)|$
- La nueva formulación es:

$$\arg\min_{w_1, w_0} \frac{1}{2}||w_1||^2 + C\sum_{k=1}^{K} \xi_k;$$

$$\text{sujeto a: } z_k(w_1\phi(x_k) + w_0) \geq 1 - \xi_k,\ \xi_k \geq 0,\ k = 1 \ldots n.$$

donde $C > 0$ controla el trade-off entre penalización y margen.

$$\arg \min_{w_1, w_0, \xi} \frac{1}{2}||w_1||^2 + C\sum_{k=1}^{K} \xi_k; \text{ with } \xi_k = |z_k - g(x_k)|$$

$$\text{subject to: } z_k(w_1\phi(x_k) + w_0) \geq 1 - \xi_k,\ \xi_k \geq 0,\ k = 1 \ldots n.$$

- Registro correctamente clasificado y fuera de la frontera de mínimo margen tiene asociado slag $\xi_k = 0$, ¿por qué?
- Registro en la frontera de mínimo margen tiene asociado slag $\xi_k = 1$, ¿por qué?
- Registro dentro de la frontera de mínimo margen pero correctamente clasificado tiene asociado slag $0 < \xi_k \leq 1$,¿por qué?
- Registros en el lado incorrecto de la superficie de decisión tiene asociado slag $\xi_k > 1$, ¿por qué?

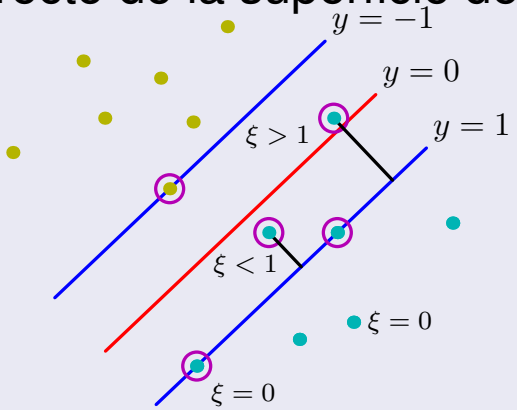


Figure: Registros encerrados con círculo corresponden a vectores de soporte.

1. ¿Cómo garantizamos que los datos sean linealmente separables en nuevo espacio? No podemos garatizarlo pero podemos usar variables auxiliares para permitir que el problema de optimización tenga solución.
2. ¿Cómo encontramos una buena función de Kernel para realizar la transformación? Vamos a ver algunas funciones de Kernel pero esta temática la dejaremos como opcional.
3. ¿Cómo podemos aplicar SVMs a problemas de más de 2 clases? Para problemas con múltiples clases se aplica el mismo principio binario, este se puede descomponer en múltiples casos de clasificación binaria. Este básicamente consiste en primero dividir la clase C_0 del resto, y así sucesivamente con las demás.
4. NOTA: AQUÍ puedes encontrar un ejemplo de cómo programar esto:

Material Opcional

What conditions should a kernel meet?

- A kernel functions should provide a direct mapping to inner products in feature space (Kernel Trick).
- Formally, a kernel is a function k that $\forall x, x' \in X$ satisfies:
  $k(x, x') = \langle \phi(x), \phi(x') \rangle$
  where $\langle \cdot, \cdot \rangle$ denotes an inner product
  $\phi : x \rightarrow \phi(x) \in F$,
  and F is an inner product feature space.
- An important fact is that the previous definition allow us to use kernel evaluations to implicitly calculate inner product between projections of points to feature space.
- This avoids paying the penalty of transforming to a high dimensionality space.

Kernels and a similarity metric

- The definition of a kernel in terms of an inner product highlights its relation to a given similarity metric.
- In general, finding a similiarity measure that can be implied by a kernel function is in general more natural than performing an explicit construction of a feature space.
- Actually, the concept of a similarity metric is key to the operation of a kernel method. A kernel is like an oracle guessing the similarity of two data points.

Kernels: Material Opcional

When a kernel is valid?

- A kernel is valid if its associated Gram matrix is positive semi-definite

Gran matrix

- A Gram matrix is defined as the matrix G with entries $Gij = \langle z_i, z_j \rangle$, where z_i, z_j are vectors in a inner product space.

Recall: positive semi-definite matrix

- A matrix G is positive semi-definite is its eigenvalues are all non-negatives.
- This is equivalent to state that: $x^T Gx > 0$ for all non-zero vectors $x \in \mathbb{R}$.

Kernel matrix

If we have a feature space where the inner product can be defined in terms of points provided by a kernel function $\phi(\cdot)$, the entries of the associated Gram matrix, a.k.a. kernel matrix κ, are given by:

$$G_{i,j} = \langle \phi(x_i), \phi(x_j) \rangle = \kappa(x_i, x_j),$$

where x_i, x_j are points in the input space, and $\phi(x_i), \phi(x_j)$ are the corresponding images in feature space.

Some examples

- Polynomial kernel:

$$K(x_i, x_j) = (\alpha\langle x_i, x_j\rangle + c)^d,$$

where $\langle x_i, x_j\rangle$ denotes the traditional linear dot product. Adjustable parameters are the slope α, the constant term c and the polynomial degree d.

- Gaussian radial basis functions (RBFs):

$$K(x_i, x_j) = e^{-\frac{||x_i - x_j||}{2\sigma^2}},$$

where $||x_i - x_j||$ denotes norm. σ is the main parameter. If overestimated, the kernel behaves almost linearly. If underestimated, the kernel lacks regularization and the decision boundary is highly sensitive to noise in training data.

Some examples

- Exponential RBFs:

$$K(x_i, x_j) = e^{-\frac{||x_i - x_j||}{2\sigma^2}}$$

  Similar to the Gaussian RBF but without the square in the norm.

- Exponential ANOVA kernel:

$$K(x_i, x_j) = \left(\sum_{k=1}^{n} e^{-\sigma(x_i^k - x_j^k)^2} \right)^d$$

  NOVA kernel is also a RBF. It has been shown to perform well in multidimensional regression problems (Hofmann, 2008).

Many more kernels

- Fourier series kernels
- Laplacian Kernel
- Hyperbolic Tangent (Sigmoid) Kernel
- Rational Quadratic Kernel
- Multiquadric Kernel
- Inverse Multiquadric Kernel
- Circular Kernel
- Spherical Kernel
- Wave Kernel
- Power Kernel
- Log Kernel
- Spline Kernel
- B-Spline Kernel
- Bessel Kernel
- Cauchy Kernel
- Chi-Square Kernel
- Histogram Intersection Kernel
- Generalized Histogram Intersection Kernel
- Generalized T-Student Kernel
- Bayesian Kernel
- ...

Check:
http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html