SUPPORT VECTOR MACHINES

Biblio:
- Bishop, "Pattern Recognition and Machine Learning", Cap 7.
- Schölkopf & Smola, "Learning with Kernels", cap 7.

- Limitante de los métodos de Kernel: (i) si se requiere evaluar $k(\underline{x}_n, \underline{x}_m)$ para todo par de muestras de entrenamiento, puede volverse irrealizable; (ii) testing o clasificación de $\underline{x}$: costo computacional de evaluar $k(\underline{x}, \underline{x}_n)$ para todo $\underline{x}_n$ puede ser excesivo.

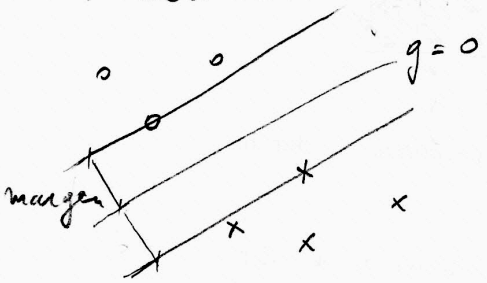
- SVM: Veremos que
  - la clasificación de nuevas muestras depende sólo de la evaluación de funciones kernel en un subconjunto extremadamente reducido del conjunto de entrenamiento.
  - los parámetros del clasificador se determinan fácilmente, resolviendo un problema de optimización convexa.

1. PROBLEMA DE DOS CLASES

- $\{\underline{x}_1, \underline{x}_2, \dots, \underline{x}_N\}$ muestras de entrenamiento, $\underline{x}_n \in \mathcal{X}$
  $\underline{x}_n \leftrightarrow$ etiqueta $t_n \in \{-1, 1\}$, $n = 1, \dots, N$
- $\phi(\cdot)$ mapeo no lineal a un espacio de dimensión mayor: $\phi : \mathcal{X} \longrightarrow \mathcal{Y}$, $\dim \mathcal{Y} > \dim \mathcal{X}$
  Ec. de superficie de decisión $\mathcal{S}$:
  $$g(\underline{x}) = \underline{w}^T \phi(\underline{x}) + b \quad , \quad \mathcal{S} = \{\underline{x} : g(\underline{x}) = 0\}$$
  En el espacio transformado $\mathcal{Y}$, la superficie de decisión es un hiperplano de vector normal $\frac{\underline{w}}{\|\underline{w}\|}$ y desp. $b/\|w\|$
- Una nueva muestra $\underline{x}$ se clasifica según signo$(g(\underline{x}))$

1.1. CLASES LINEALMENTE SEPARABLES (EN APRENDIZAJE)

- En este caso existe al menos un par $(\underline{w}, b)$ para el cual todas las muestras de entrenamiento están bien clasificadas, i.e. $t_n\, g(\underline{x}_n) > 0 \quad \forall\ i = 1, .., N$
- Entre los $(\underline{w}, b)$ posibles, SVM elige aquél que corresponde al hiperplano separador con margen máximo



Es intuitivo (y se puede demostrar [Vapnik]) que maximizar el margen minimiza el error de generalización del clasificador.

- Distancia de un punto $\underline{x}$ al plano $\{g(\underline{x}) = 0\}$: $\dfrac{|g(x)|}{\|w\|}$

Para cualquier hiperplano separador, puesto que estamos en el caso separable:

$$|g(\underline{x}_n)| = t_n\, g(\underline{x}_n), \quad n = 1, 2, .., N$$

Solución de máximo margen:

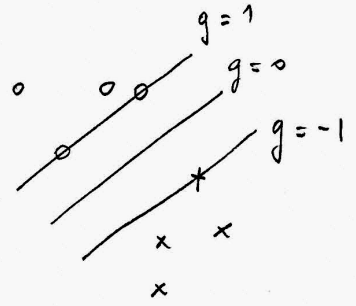
$$\underset{\underline{w},\, b}{\arg\max} \left\{ \frac{1}{\|\underline{w}\|} \min_{n \in \{1, .., N\}} \left[t_n \left(\underline{w}^T \phi(\underline{x}_n) + b \right) \right] \right\}$$

Esta optimización es complicada, pero podemos encontrar una formulación equivalente sencilla de optimizar:

Obs: el escalado de $\underline{w}$ y b por una misma constante no altera el cociente $\dfrac{t_n\, g(\underline{x}_n)}{\|\underline{w}\|}$

$\Rightarrow$ Podemos definir la normalización tal que $t_k \left(\underline{w}^T \phi(\underline{x}_k) + b \right) = 1$ para el (o los) puntos x_k que realizan el margen (el más cercano a $\{g(\underline{x}) = 0\}$)

Luego, tenemos que $\forall\ n = 1, 2, .., N$, $t_n \left(\underline{w}^T \phi(\underline{x}_n) + b \right) \geq 1$.



. Se llaman:

- puntos activos: aquellos para los cuales $t_n(w^T\phi(x_n)+b)=1$
- puntos inactivos: el resto (>1)

• Siempre habrá al menos un punto activo, y al maximizar el margen habrá al menos dos

Tenemos entonces la siguiente formulación equivalente del problema de maximización del margen con clasificación (de entrenamiento) perfecta:

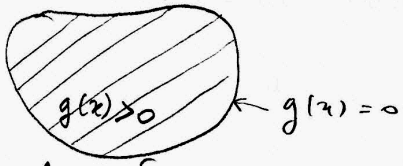
$$\begin{cases} \min_{\underline{w},b} \ \frac{1}{2}\|\underline{w}\|^2 \\ \text{sujeto a} \quad t_n(\underline{w}^T\phi(\underline{x}_n)+b) \geq 1 \ , \quad n=1,\dots,N \end{cases}$$

Es un problema de programación cuadrática, fácil de resolver. Hay "solvers" que vienen en el toolbox de optimización de Matlab (ver también el toolbox no propietario CVX).

Pero el problema anterior (llamado primal) se puede llevar a una formulación aún más sencilla mediante las condiciones de Karush-Kuhn-Tucker (KKT). (llamado problema dual).

BREVE PARENTESIS: CONDICIONES KKT

$$\begin{cases} \min f(x) \\ \text{s.t.} \ \ g(x) \geq 0 \end{cases}$$

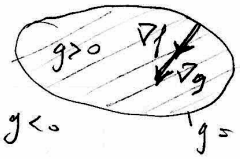


2 posibilidades para la solución:

(i) $x \in \{g(x)>0\}$: en este caso x es mínimo local de $f \Rightarrow \nabla f(x)=0$

(ii) $x \in \{g(x)=0\}$: en este caso debe ser $\nabla f(x) \perp \{g=0\}$, ya que de lo contrario existiría un vecino x' de x, $x' \in \{g=0\}$, con $f(x')<f(x)$

Como $\nabla g \perp \{g=0\}$, $\exists\, \lambda \neq 0$ t.q. $\nabla f(x) = \lambda \nabla g(x)$. Luego, debe ser $\lambda > 0$, ya que si $\lambda < 0$, existiría un vecino x' de x, $x' \in \{g>0\}$, para el cual $f(x') < f(x)$.



Definiendo el Lagrangeano

$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x)$, las 2 situaciones se resumen en una única, mediante las condiciones:

$$\frac{\partial \mathcal{L}}{\partial \underline{x}} = 0 \quad , \quad \lambda \geq 0 \, , \quad \lambda g(x) = 0$$

(a esta última se le llama condición kkT)

Obs 1: La solución (x^*, λ^*) es punto silla de $\mathcal{L}(x, \lambda)$:

$$\mathcal{L}(x^*, \lambda) \leq \mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x, \lambda^*) \, , \quad \forall \ (x, \lambda)$$

En efecto,

. $\mathcal{L}(x^*, \lambda) \leq \mathcal{L}(x^*, \lambda^*)$

$$\Leftrightarrow \cancel{f(x^*)} - \underbrace{\lambda g(x^*)}_{\leq 0} \leq \cancel{f(x^*)} - \underbrace{\lambda^* g(x^*)}_{=0} \quad \checkmark$$

. $\mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x, \lambda^*)$

$$\Leftrightarrow f(x^*) - \underbrace{\lambda^* g(x^*)}_{=0} \leq f(x) - \lambda^* g(x)$$

$$\Leftrightarrow \underbrace{\lambda^* g(x)}_{\geq 0} \leq f(x) - f(x^*) \quad \checkmark$$

Obs 2: De obs 1; la optimización correspondiente es

$$\min_{x} \max_{\lambda} \mathcal{L}(x, \lambda) \, .$$

Volviendo a nuestro problema:

$$\mathcal{L}(\underline{w}, b, \underline{a}) = \frac{1}{2} \|w\|^2 - \sum_{n=1}^{N} a_n \underbrace{\left(t_n \left(\underline{w}^T \phi(\underline{x}_n) + b \right) - 1 \right)}_{\geq 0}$$

$\phi^T(\underline{x}_n) w$

son las restricciones de positividad

$$(P) \quad \begin{cases} \min\limits_{\underline{w}, b} \max\limits_{\underline{a}} \mathcal{L}(\underline{w}, b, \underline{a}) \\ \text{con } a_n \geq 0, \quad n = 1, 2, \ldots, N \end{cases}$$

$$\frac{\partial \mathcal{L}}{\partial \underline{w}} = \underline{0} \Leftrightarrow \underline{w} = \sum_{n=1}^{N} a_n t_n \phi(\underline{x}_n) \qquad (*)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Leftrightarrow \sum_{n=1}^{N} a_n t_n = 0 \qquad (**)$$

Reemplazamos (*) en $\mathcal{L}(\underline{w}, b, \underline{a})$:

$$\mathcal{L}(\underline{w}, b, \underline{a}) = \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m \phi^T(\underline{x}_n) \phi(\underline{x}_m)$$
$$- \sum_{n=1}^{N} a_n t_n \phi^T(\underline{x}_n) \sum_{m=1}^{N} a_m t_m \phi(\underline{x}_m)$$
$$- b \underbrace{\sum_{n=1}^{N} a_n t_n}_{=0 \ (ec\ **)} + \sum_{n=1}^{N} a_n$$

$$\mathcal{L}(\underline{w}, b, \underline{a}) = \tilde{\mathcal{L}}(\underline{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m \underbrace{\phi^T(\underline{x}_n) \phi(\underline{x}_m)}_{k(\underline{x}_n, \underline{x}_m)}$$

El nuevo problema, llamado problema dual, queda:

$$(P') \begin{cases} \max\limits_{\underline{a}} \ \tilde{\mathcal{L}}(\underline{a}) \\ \text{con} \ \ \underline{a} \geq \underline{0} \end{cases}$$

De nuevo, este es un problema de programación cuadrática, pero todavía más sencillo de resolver ya que involucra únicamente a los multiplicadores de Lagrange.

- Clasificación de una nueva muestra $\underline{x}$:

Se evalúa el signo de $g(\underline{x}) = \underline{w}^T \phi(\underline{x}) + b$

$$= \sum_{n=1}^{N} a_n t_n k(\underline{x}_n, \underline{x}) + b$$

Veremos como calcular b más adelante.

- Vectores de soporte

De las condiciones KKT;
$$a_n \geq 0$$
$$t_n g(\underline{x}_n) - 1 \geq 0$$
$$a_n (t_n g(\underline{x}_n) - 1) = 0$$

Tenemos dos tipos de muestras de entrenamiento, que verifican de una forma u otra la última condición:

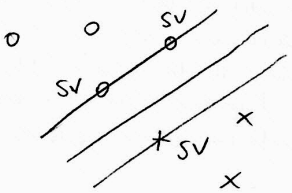
(i) $a_n = 0$

(ii) $t_n g(\underline{x}_n) = 1$

Las primeras no influyen en la clasificación de una nueva muestra:

$$g(\underline{x}_n) = \sum_{\substack{a_n \neq 0 \\ n=1,\dots,N}} a_n t_n k(\underline{x}_n, \underline{x}) + b$$

Las restantes se llaman vectores de soporte (SV), como verifican $t_n g(\underline{x}_n) = 1$, son muestras que caen sobre los hiperplanos de máximo margen.



Conclusión: en la etapa de testing, una vez que se ha entrenado el modelo, el proceso es rápido porque sólo se consideran los vectores de soporte ($\#SV \ll N$ en general)

¿Cuanto vale el margen?

$$SV: \; t_n g(\underline{x}_n) = 1 \Rightarrow t_n \left(\sum_{\underline{x}_m \, SV} a_m t_m k(\underline{x}_m, \underline{x}_n) + b \right) = 1$$

$$\Rightarrow \sum_{\underline{x}_n \, SV} \underbrace{t_n^2}_{=1} \left(\sum_{\underline{x}_m \, SV} a_m t_m k(\underline{x}_m, \underline{x}_n) + b \right) = \sum_{\underline{x}_n \, SV} t_n$$

$\times t_n$ y sumo en los SV

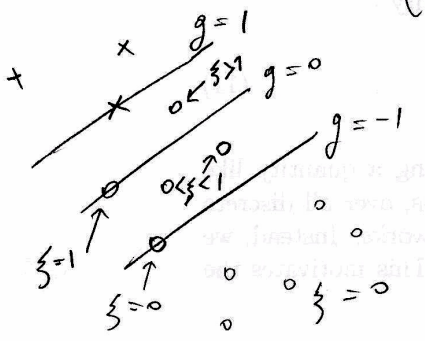
$$\Rightarrow b = \frac{1}{\#SV} \sum_{\underline{x}_n \, SV} \left(t_n - \sum_{\underline{x}_m \, SV} a_m t_m k(\underline{x}_m, \underline{x}_n) \right)$$

1.2. CASO NO SEPARABLE LINEALMENTE (EN APRENDIZAJE)

la idea es modificar la formulación anterior para autorizar que algunos puntos caigan del lado equivocado del hiperplano, pero penalizando esos errores, por ejemplo con una penalización que crezca con la distancia al hiperplano de decisión.

Tomaremos la penalización siguiente:

$$\xi_n = \begin{cases} 0 & \text{si } \underline{x}_n \text{ cae del lado correcto} \\ |t_n - g(\underline{x}_n)| & \text{si no.} \end{cases}$$



. Puntos en la frontera: $g(\underline{x}_n) = 0 \Rightarrow \xi_n = 1$

. Punto mal clasificados: $\xi_n > 1$

Remplazamos la condición de clasificación perfecta

$t_n(\underline{w}^T \phi(\underline{x}_n) + b) \geq 1$ ("Hard Margin")

por la condición relajada o "Soft Margin":

$$\begin{cases} t_n(\underline{w}^T \phi(x_n) + b) \geq 1 - \xi_n, & n = 1, \dots, N \\ \text{Con } \xi_n \geq 0 \end{cases}$$

OBJETIVO: maximizar el margen, penalizando errores:

$$\begin{cases} \min_{\underline{w}, b} \left\{ C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \|\underline{w}\|^2 \right\}, & C > 0. \\ \text{sujeto a} \begin{cases} t_n(\underline{w}^T \phi(\underline{x}_n) + b) \geq 1 - \xi_n, & n = 1, \dots, N. \\ \xi_n \geq 0 \end{cases} \end{cases}$$

C: constante de trade-off errores vs. margen

$C \to \infty$: no autorizo errores, Hard Margin.

$C \downarrow$: más errores, margen mayor.

El problema se resuelve de forma análoga:

$$\mathcal{L}(\underline{w}, b, \underline{a}, \underline{\mu}) = \frac{1}{2}\|w\|^2 + C\sum_{n=1}^{N}\xi_n - \sum a_n\left(t_n g(\underline{x}_n) - 1 + \xi_n\right) - \sum_{n=1}^{N}\mu_n\xi_n$$

$a_n \geq 0$, $\mu_n \geq 0$ mult. de Lagrange

KKT:

$$\begin{cases} a_n \geq 0 \\ t_n g(\underline{x}_n) - 1 + \xi_n \geq 0 \\ a_n\left(t_n g(\underline{x}_n) - 1 + \xi_n\right) = 0 \\ \mu_n \geq 0 \\ \xi_n \geq 0 \\ \mu_n \xi_n \geq 0 \end{cases}$$

$$\frac{\partial \mathcal{L}}{\partial \underline{w}} = 0 \iff \underline{w} = \sum_{n=1}^{N} a_n t_n \phi(x_n)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \iff \sum_{n=1}^{N} a_n t_n = 0$$

$$\frac{\partial \mathcal{L}}{\partial \xi_n} = 0 \iff a_n = C - \mu_n \iff \mu_n = C - a_n$$

Remplazando $\underline{w}$, μ_n en $\mathcal{L}(\underline{w}, b, \underline{a}, \underline{\mu})$ con las ec. anteriores, los ξ_n y b se cancelan y obtenemos el Lagrangeano dual

$$\tilde{\mathcal{L}}(\underline{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\underline{x}_n, \underline{x}_m)$$

Con $\left.\begin{array}{l} a_n \geq 0 \\ \mu_n = C - a_n \geq 0 \end{array}\right\} \Rightarrow 0 \leq a_n \leq C$

- Clasificación de nueva muestra $\underline{x}$:

según signo de $g(\underline{x}) = \sum_{n=1}^{N} a_n t_n k(\underline{x}_n, \underline{x}) + b$

- Support vectors ?

$a_n > 0$

$t_n g(\underline{x}_n) = 1 - \xi_n$

- Si $a_n < C \Rightarrow \mu_n > 0 \Rightarrow \xi_n = 0$: son SV dentro del lado correcto, fuera del margen.

($\mu_n \xi_n = 0$)

- Si $a_n = C \Rightarrow$ puede ser $\mu_n = 0 \Rightarrow \begin{cases} \xi_n \leq 1 & \text{bien clasif.} \\ \xi_n > 1 & \text{mal clasificado} \end{cases}$

- Margen b ?

Support vectors: $0 < a_n < C \Rightarrow \xi_n = 0 \Rightarrow t_n g(\underline{x}_n) = 1$

Exactamente de la misma forma que para el SVM Hard Margin, obtenemos:

$$b = \frac{1}{\#S} \sum_{\underline{x}_n \in S} \left(t_n - \sum_{\underline{x}_m \in S} a_m t_m k(\underline{x}_m, \underline{x}_n) \right)$$

donde $S = \{ \underline{x}_m : 0 < a_n < C, \; n = 1, 2, \ldots, N \}$

Conjunto de SVs.

- El método Soft-Margin presentado aquí se conoce como C-SVM. Existen otras versiones soft margin (e.g. ν-SVM, ver Schölkopf & Smola).

- Usualmente C y el o los parámetros del kernel se estiman conjuntamente usando Grid Search y validación cruzada sobre el conjunto de entrenamiento.

2. EXTENSIÓN A M CLASES

2.1. Uno contra el resto

- Para cada una de las M clases, se construye un clasificador lineal usando como conjunto de entrenamiento la clase en cuestión por un lado, y todo el resto por el otro lado:

$$g^j(\underline{x}) = \sum_{n=1}^{N} a_n^j t_n^j k(\underline{x}_n, \underline{x}) + b^j \quad , \quad j = 1, \dots, M$$

- Una muestra no etiquetada $\underline{x}$, se asigna a la clase $\underset{j=1,\dots,M}{\operatorname{argmax}}\ g^j(\underline{x})$
- Desventaja: poco simétrico.

2.2. Clasificación por pares

- Para cada par de clases, se consideran únicamente los datos de esas dos clases y se entrena un clasificador lineal para separarlas $\longrightarrow$ $M(M-1)/2$ clasificadores lineales
- Costo computacional: hay que resolver más problemas que para "uno contra todos", pero con menos muestras.
- Clasificación de una nueva muestra $\underline{x}$:
  - Se evalúan los $M(M-1)/2$ clasificadores, y $\underline{x}$ se asigna a la clase más votada.

2.3. Error correcting output coding

- Se generan L particiones lineales de clases. Por ejemplo, para la base de dígitos 0, 1, 2, ..., 9 USPS:

$$P_1 = (\{0,1,\dots,4\}, \{5,\dots,9\}), \quad P_2 = (\{0,1,4,5,8\}, \{2,3,6,7,9\})$$
$$P_2 = (\{0,2,4,6,8\}, \{1,3,5,7,9\}) \quad \dots$$

La idea es que a cada clase le corresponda un único código $\underline{d} = (d_1, d_2, \dots, d_L)$, con $d_i \in \{-1, 1\}$ según en qué sub-clase de la partición P_i esté la clase. Luego se entrena un clasificador lineal para cada partición, $f_1, \dots, f_L$.

Clasificación: $\cdot\ \underline{x} \to (f_1(\underline{x}), f_2(\underline{x}), \dots, f_L(\underline{x})) = F(\underline{x})$
$\cdot\ \underline{x}$ se asigna a $\underset{j=1,\dots,M}{\operatorname{argmin}}\ D(F(\underline{x}), \underline{d}_j)$

D: Hamming u otra distancia que además contemple cuán grande es cada error.