UCZENIE MASZYNOWE

MASZYNA WEKTORÓW NOŚNYCH - REGRESJA

Dr hab. inż. Grzegorz Dudek
Wydział Elektryczny
Politechnika Częstochowska

Projekt finansowany w ramach programu Ministra Nauki i Szkolnictwa Wyższego pod nazwą „Regionalna Inicjatywa Doskonałości" w latach 2019 - 2022 nr projektu 020/RID/2018/19 kwota finansowania 12 000 000 PLN

FUNKCJE JĄDROWE WYKORZYSTYWANE W SVM

Najpopularniejsze funkcje jądrowe:

- wielomian stopnia *q*:

$$K(\mathbf{x}_i, \mathbf{x}) = (\mathbf{x}^T \mathbf{x}_i + 1)^q$$

Dla $q = 2$ i $n = 2$:

$$K(\mathbf{x}_a, \mathbf{x}_b) = (\mathbf{x}_a^T \mathbf{x}_b + 1)^2 = (x_{a,1}x_{b,1} + x_{a,2}x_{b,2} + 1)^2 = 1 + 2x_{a,1}x_{b,1} + 2x_{a,2}x_{b,2} + 2x_{a,1}x_{b,1}x_{a,2}x_{b,2} + x_{a,1}^2 x_{a,2}^2 + x_{b,1}^2 x_{b,2}^2$$

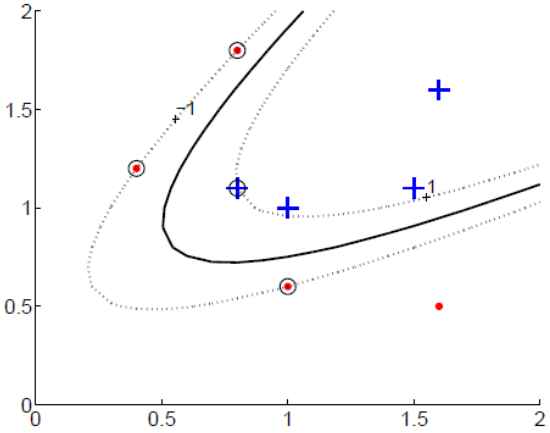
Odpowiada to iloczynowi skalarnemu funkcji bazowych postaci:

$$\boldsymbol{\varphi}(\mathbf{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2]^T$$

(Ale te funkcje nie muszą być znane, wystarczy znać jądro!)

FUNKCJE JĄDROWE WYKORZYSTYWANE W SVM

W takim przypadku powierzchnia decyzyjna w przestrzeni *X* ma postać:



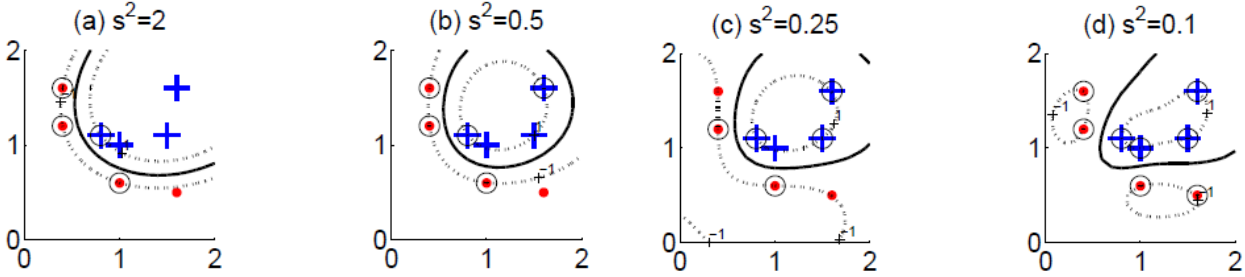
- radialna funkcja bazowa:

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|}{2s^2}\right)$$

Parametr *s* - szerokość funkcji radialnej dobieramy w kroswalidacji.

FUNKCJE JĄDROWE WYKORZYSTYWANE W SVM

Różne powierzchnie decyzyjne w zależności od parametru *s*:



- jądro sigmoidalne:

$$K(\mathbf{x}_i, \mathbf{x}) = \tanh(2\mathbf{x}^T\mathbf{x}_i + 1)$$

Funkcje jądrowe mierzą podobieństwo pomiędzy przykładami. Im przykłady są do siebie bardziej podobne tym wartość funkcji jądrowej jest większa (maksymalna dla identycznych przykładów).

FUNKCJE JĄDROWE WYKORZYSTYWANE W SVM

Możemy definiować jądra specyficzne dla danego problemu. Poprzez odpowiednio zdefiniowane jądra możemy wprowadzać dodatkową wiedzę o problemie (*kernel engineering*).

Zależnie od sposobu reprezentacji danych możemy tworzyć jądra łańcuchowe (*string kernels*), drzewiaste (*tree kernels*), grafowe (*graph kernels*).

Na przykład, gdy analizujemy dwa dokumenty jądrem może być liczba jednakowych słów pojawiająca się w tych dokumentach.

Typowo dla dwóch dokumentów D_1 i D_2 określa się listę M słów i definiuje funkcję $\varphi(D)$ jako M-wymiarowy wektor binarny. Jedynka na pozycji i-tej w tym wektorze oznacza, że i-te słowo z listy występuje w dokumencie. Iloczyn skalarny $\varphi(D_1)^T\varphi(D_2)$ wyznacza liczbę słów jednakowych w obu dokumentach. Jeśli bezpośrednio zdefiniujemy jądro $K(D_1, D_2)$ jako liczbę słów wspólnych nie musimy wyznaczać listy M słów.

Rozważmy model regresji liniowej:

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

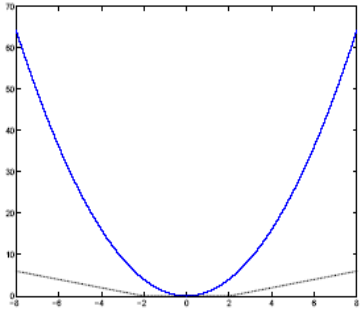
W regresji używamy błędu kwadratowego:

$$e_2(y_i, f(\mathbf{x}_i)) = (y_i - f(\mathbf{x}_i))^2$$

Natomiast w SVM błąd definiujemy następująco:

$$e_\varepsilon(y_i, f(\mathbf{x}_i)) = \begin{cases} 0, & \text{jeżeli } |y_i - f(\mathbf{x}_i)| < \varepsilon \\ |y_i - f(\mathbf{x}_i)| - \varepsilon, & \text{w przeciwnym przypadku} \end{cases}$$

co oznacza, że tolerujemy błędy mniejsze od ε i błąd jest liniowy, nie kwadratowy.



Analogicznie do klasyfikatora SVM dla danych nieseparowanych definiujemy "miękki" błąd i dodajemy go do funkcji celu:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_{i+} + \xi_{i-})$$

Ograniczenia:

$$y_i - (\mathbf{w}^T\mathbf{x}_i + w_0) \le \varepsilon + \xi_{i+}$$
$$(\mathbf{w}^T\mathbf{x}_i + w_0) - y_i \le \varepsilon + \xi_{i-}$$
$$\xi_{i+}, \xi_{i-} \ge 0$$

gdzie zmienne ξ_{i+} i ξ_{i-} dotyczą dodatnich i ujemnych odchyłek, odpowiednio.

Lagrangian ma postać:

$$L_P(\mathbf{w}, w_0, \boldsymbol{\alpha}_+, \boldsymbol{\alpha}_-, \boldsymbol{\mu}_+, \boldsymbol{\mu}_-) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_{i+} + \xi_{i-}) - \sum_{i=1}^{N}\alpha_{i+}[\varepsilon + \xi_{i+} - y_i + (\mathbf{w}^T\mathbf{x}_i + w_0)]$$
$$- \sum_{i=1}^{N}\alpha_{i+}[\varepsilon + \xi_{i-} + (\mathbf{w}^T\mathbf{x}_i + w_0) - y_i] - \sum_{i=1}^{N}(\mu_{i+}\xi_{i+} + \mu_{i-}\xi_{i-})$$

Przyrównując pochodne L_P po $\mathbf{w}$, w_0 i ξ_i do zera otrzymujemy:

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \quad \rightarrow \quad \mathbf{w} = \sum_{i=1}^{N}(\alpha_{i+} + \alpha_{i-})\mathbf{x}_i\,, \qquad \frac{\partial L_P}{\partial w_0} = 0 \quad \rightarrow \quad \sum_{i=1}^{N}(\alpha_{i+} + \alpha_{i-})\mathbf{x}_i = 0$$

$$\frac{\partial L_P}{\partial \xi_{i+}} = 0 \quad \rightarrow \quad C - \alpha_{i+} - \mu_{i+} = 0\,, \qquad \frac{\partial L_P}{\partial \xi_{i-}} = 0 \quad \rightarrow \quad C - \alpha_{i-} - \mu_{i-} = 0$$

Podstawiając powyższe do L_P otrzymamy postać dualną:

$$L_D = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_{i+} + \alpha_{i-})(\alpha_{j+} + \alpha_{j-})\mathbf{x}_i^T\mathbf{x}_j - \varepsilon\sum_{i=1}^{N}(\alpha_{i+} + \alpha_{i-}) - \sum_{i=1}^{N}y_i(\alpha_{i+} + \alpha_{i-})$$

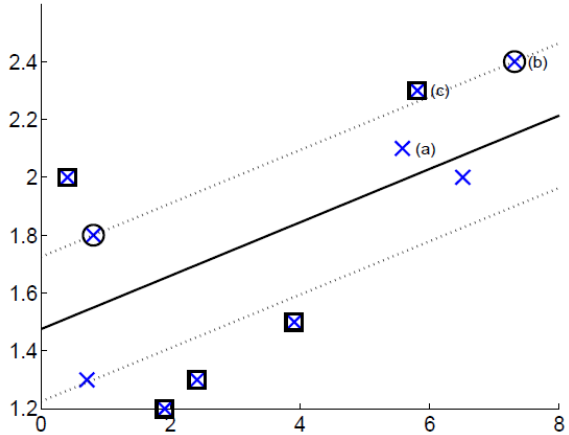
przy ograniczeniach: $0 \le \alpha_{i+}, \alpha_{i-} \le C$ i $\sum_{i=1}^{N}(\alpha_{i+} - \alpha_{i-}) = 0$

Maksymalizujemy L_D ze względu na mnożniki α_i i w wyniku otrzymujemy:

a) $\alpha_{i+} = \alpha_{i-} = 0$ dla punktów leżących w paśmie pomiędzy płaszczyznami granicznymi marginesu; są to punkty aproksymowane z akceptowalnym błędem,

b) $0 < \alpha_{i+} < C$ lub $0 < \alpha_{i-} < C$ dla punktów leżących na płaszczyznach granicznych marginesu,

c) $\alpha_{i+} = C$ lub $\alpha_{i-} = C$ dla punktów leżących poza pasmem marginesu; są to punkty aproksymowane z nieakceptowalnym błędem ($>\varepsilon$).



Wektorami nośnymi nazywa się punkty, dla których $\alpha_i > 0$. Punkty te definiują wektor normalny **w** płaszczyzny aproksymacyjnej. Wartości w_0 wyznaczamy na podstawie punktów leżących na granicach marginesu z równania $y_i = \mathbf{w}^T\mathbf{x}_i + w_0 \pm \varepsilon$.

Funkcja aproksymująca zależy jedynie od wektorów nośnych:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = \sum_{i=1}^{N}(\alpha_{i+} + \alpha_{i-})\mathbf{x}_i^T\mathbf{x} + w_0$$

Iloczyn skalarny $\mathbf{x}_i^T\mathbf{x}$, podobnie jak w przypadku klasyfikatora, zastępujemy jądrem $K(\mathbf{x}_i, \mathbf{x})$, co umożliwi aproksymację nieliniową: