Data Science
Metody Sztucznej Inteligencji

Paweł Wawrzyński

Uczenie maszynowe
Klasyfikacja i regresja

Plan na dziś

- Uczenie maszynowe
- Aproksymacja funkcji / regresja
  - Aproksymatory parametryczne
- Klasyfikacja i klasyfikatory
  - Maszyny wektorów nośnych
  - Gradient boosting

Uczenie maszynowe

- Problemy
  - sterowniki dla systemów o nieznanej dynamice
  - budowa modeli na podstawie napływających danych
- Techniki:
  - aproksymacja funkcji
  - klasyfikacja
  - grupowanie
  - uczenie się ze wzmocnieniem
- Narzędzia
  - sieci neuronowe
  - drzewa decyzyjne

Problem aproksymacji funkcji

- X – przestrzeń wejść
- Y – przestrzeń wyjść
- dostępne są próbki $\left(x_t, f\left(x_t\right)+\xi_t\right) \in X \times Y$
- należy określić przybliżoną reprezentację funkcji f

Problem regresji

- X – przestrzeń wejść
- Y – przestrzeń wyjść
- dostępne są próbki $(x_t, y_t) \in X \times Y$
- próbki pochodzą z pewnego rozkładu $P_{x,y}$
- należy określić przybliżoną reprezentację

$$E(y|x)$$

Problem klasyfikacji

- X – przestrzeń wejść
- Y – **dyskretna** przestrzeń klas
- dostępne są próbki $(x_t, y_t) \in X \times Y$
- pochodzące z rozkładu $P_{x,y}$
- należy zbudować klasyfikator, który:
  dla danego x
  wkskazuje y najczęściej towarzyszący temu x
  w rozkładzie $P_{x,y}$

Uogólnienie: problem decyzji statystycznych

- X – przestrzeń wejść
- Y – przestrzeń decyzji
- dostępne są próbki (x_t, q_t), t.że $x_t \in X$, $q_t : Y \to R$
- pochodzące z rozkładu $P_{x,q}$
- q określa **stratę** za decyzję na podstawie x
- należy zbudować funkcję decyzyjną $d : X \to Y$, która:
  - → dla danego x
  - → wskazuje decyzję minimalizującą $q(d(x))$

Uogólnienie: problem decyzji statystycznych

- globalny wskaźnik jakości dla funkcji decyzyjnej:

$$J(d) = E\, q(d(x))$$

- klasyfikacja:

$$q(d(x)) = (y == d(x))\,?\,0:1$$

- regresja:

  błąd średniokwadratowy, *MSE - mean square error*

$$q(d(x)) = \| d(x) - y \|^2$$

- aproksymacja funkcji: j.w.

Aproksymatory parametryczne

- wejścia z przestrzeni X
- wyjścia z przestrzeni $Y = \mathbb{R}^{n_y}$
- parametry z przestrzeni $\Theta = \mathbb{R}^{n_\theta}$
- aproksymator $\bar{f} \;:\; X \times \Theta \to Y$
- cel 1: aproksymacja funkcji

$$f : X \to Y$$

- cel 2: znalezienie najlepszej funkcji wg pewnego kryterium

Przykłady prostych aproksymatorów

- wielomian

$$\bar{f}(x;\theta) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$$

- szereg trygonometryczny

$$\bar{f}(x;\theta) = \theta_1 + \sum_{k=1}^{n} \big(\theta_{2k-1}\cos(kx) + \theta_{2k}\sin(kx)\big)$$

- aproksymator liniowy

$$\bar{f}(x;\theta) = \sum_{i=1}^{d} \theta_i x_i + \theta_{d+1}$$

Przykłady prostych aproksymatorów

- tablica

$$\mathcal{X} = \bigcup_{i=1}^{n} \mathcal{X}_i$$

$$\Phi_i(x) = \begin{cases} 1 \text{ jeśli } x \in \mathcal{X}_i \\ 0 \text{ jeśli } x \notin \mathcal{X}_i \end{cases}$$

$$\bar{f}(x;\theta) = \sum_{i=1}^{n} \theta_i \Phi_i(x)$$

Zagadnienie aproksymacji funkcji na zbiorze skończonym - uczenie off-line

- Dany jest skończony zbiór elementów

$$\langle x_i, y_i \rangle, \quad i \in \{1, \ldots, N\}$$

- Należy znaleźć wektor parametrów aproksymatora, który minimalizuje wskaźnik jakości

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|y_i - \bar{f}(x_i;\theta)\|^2$$

Przykład: aproksymator liniowy

- Funkcja

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \|y_i - \bar{f}(x_i;\theta)\|^2$$

- Gradient

$$\nabla J(\theta) = \frac{1}{N}\sum_{t=1}^{N} \frac{\mathrm{d}}{\mathrm{d}\theta^T}\|\bar{f}(x_t;\theta) - y_t\|^2$$

- Działają wszystkie gradientowe metody optymalizacji:
  - w szczególności: metoda gradientu prostego

Metoda gradientu prostego

- dziedzina $\Theta = R^d$
- funkcja $J : \Theta \rightarrow R$
- ciąg parametrów kroku $\{\beta_t, t = 1,2,3,\ldots\}$
- ciąg punktów $\{\theta_t, t = 1,2,3,\ldots\}$
- obliczany wg formuły

$$\theta_{t+1} = \theta_t - \beta_t \nabla J(\theta_t)$$

Metoda gradientu prostego

Warunki zbieżności

1. $\sum_{t\geq 1} \beta_t = +\infty$
2. $\beta_t < 1/\lambda_{max}\left(\nabla^2 J\left(\theta\right)\right)$
3. Funkcja J jest ciągła i różniczkowalna
4. Hesjan $\nabla^2 J$ jest ograniczony
5. Funkcja J osiąga swoje suprema

Metoda gradientu prostego

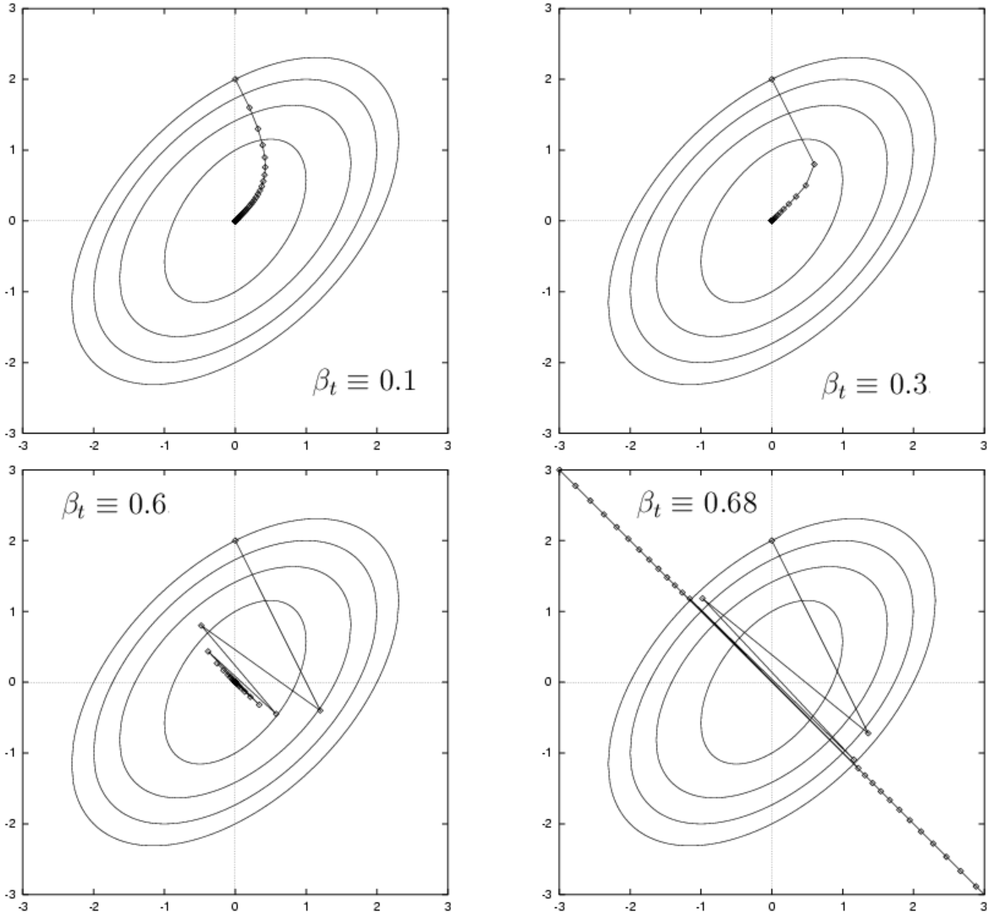
Przykład

$$\Theta = R^2$$

$$J(\theta) = J(\theta_1, \theta_2) = \theta_1^2 + \theta_2^2 - \theta_1\theta_2$$

$$J(\theta_1, \theta_2) = \frac{1}{2}[\theta_1^2 + \theta_2^2 + (\theta_1 - \theta_2)^2]$$

$$\nabla J(\theta_1, \theta_2) = \begin{bmatrix} 2\theta_1 - \theta_2 \\ 2\theta_2 - \theta_1 \end{bmatrix}$$



Przykład: aproksymator liniowy

$$\bar{f}(x;\theta) = \sum_{i=1}^{d} \theta^i x^i + \theta^{d+1}$$

$$\bar{f}(x;\theta) = [x^T 1]\theta$$

$$\frac{\mathrm{d}}{\mathrm{d}\theta^i} \|\bar{f}(x;\theta) - y\|^2 = 2(\bar{f}(x;\theta) - y)x^i,\ 1 \leq i \leq d$$

$$\frac{\mathrm{d}}{\mathrm{d}\theta^{d+1}} \|\bar{f}(x;\theta) - y\|^2 = 2(\bar{f}(x;\theta) - y).$$

$$\frac{\mathrm{d}}{\mathrm{d}\theta^T} \|\bar{f}(x;\theta) - y\|^2 = 2(\bar{f}(x;\theta) - y)\begin{bmatrix} x \\ 1 \end{bmatrix}$$

Zagadnienie aproksymacji funkcji na zbiorze nieskończonym - uczenie on-line

- Dany jest generator losowych par $\langle x, y \rangle \sim P_{x,y}$ który generuje kolejne próbki $\langle x_t, y_t \rangle, t = 1, 2, \ldots$
- Po *t*-tej próbce parametr aproksymatora jest aktualizowany (na jej podstawie) do wartości θ_t
- Ciąg parametrów $\theta_t, t = 1, 2, \ldots$ powinien zbiegać do minimum wskaźnika jakości

$$\begin{aligned} J(\theta) &= \mathcal{E} \| y - \bar{f}(x;\theta) \|^2 \\ &= \iint \| y - \bar{f}(x;\theta) \|^2 P_{x,y}(x, y) \mathrm{d}y \mathrm{d}x \end{aligned}$$

Metoda stochastycznego najszybszego spadku

- dziedzina $\Theta = R^d$
- funkcja $J : \Theta \to R$
- ciąg parametrów kroku $\{\beta_t, t = 1,2,3,\ldots\}$
- ciąg punktów $\{\theta_t, t = 1,2,3,\ldots\}$
- obliczany wg formuły

$$\theta_{t+1} = \theta_t - \beta_t g_t$$

gdzie

$$E g_t = \nabla J(\theta_t)$$

Dodatkowe warunki zbieżności

1. $\sum_{t\geq 1} \beta_t^2 < +\infty$
2. $P\left(g_t | \theta_t, g_{t-k}\right) = P\left(g_t | \theta_t\right)$
3. Wariancja g_t jest jednostajnie ograniczona

Uczenie aproksymatora przykład-po-przykładzie

- Chcemy zminimalizować

$$J(\theta) = \mathcal{E}\,\|Y - \bar{f}(X;\theta)\|^2$$

- Wykorzystujemy stochastyczny najszybszy spadek uwzględniając fakt, że przy spełnieniu pewnych warunków regularności

$$\begin{aligned}\nabla J(\theta) &= \frac{\mathrm{d}}{\mathrm{d}\theta^T}\iint \|y - \bar{f}(x;\theta)\|^2 P_{x,y}(x,y)\mathrm{d}y\mathrm{d}x \\ &= \iint \frac{\mathrm{d}}{\mathrm{d}\theta^T}\|\bar{f}(x;\theta) - y\|^2 P_{x,y}(x,y)\mathrm{d}y\mathrm{d}x \\ &= \mathcal{E}\left(\frac{\mathrm{d}}{\mathrm{d}\theta^T}\|\bar{f}(X;\theta) - Y\|^2\right)\end{aligned}$$

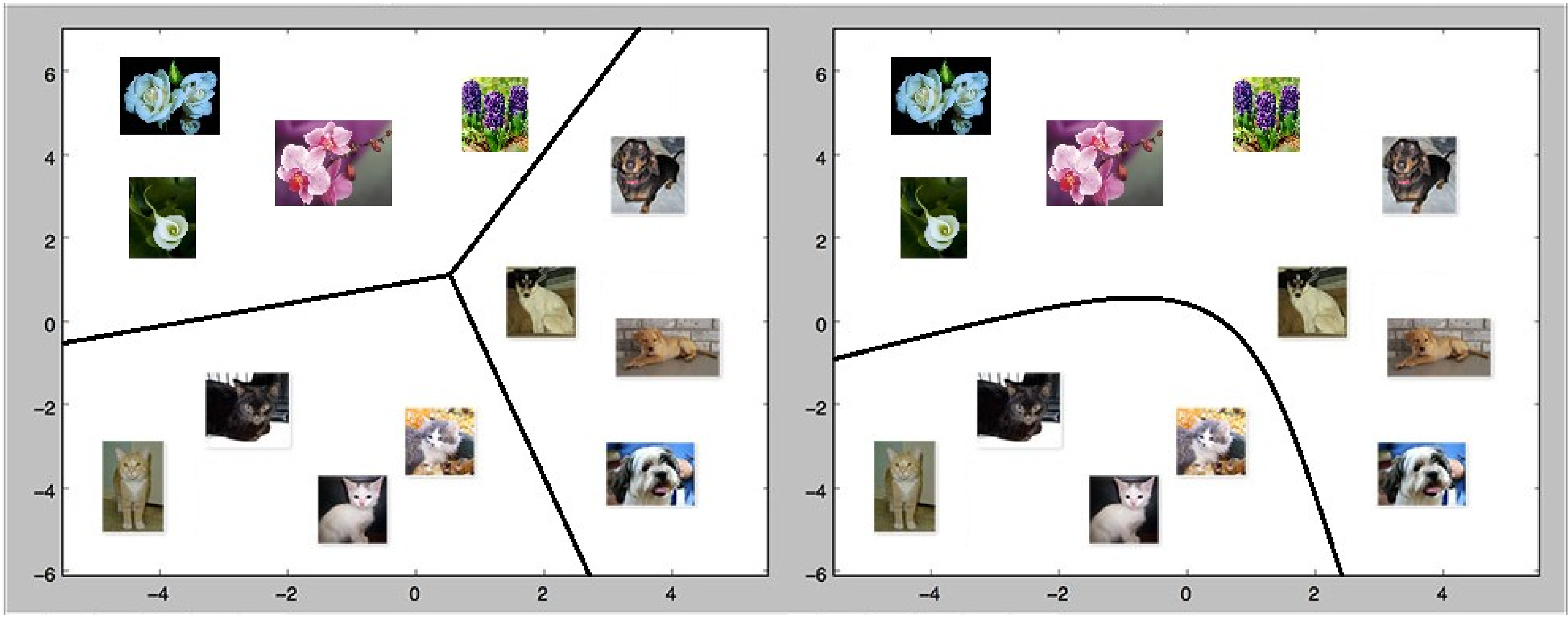
Algorytm uczenia aproksymatora

0. Dane: θ_1 – początkowe oszacowanie optymalnego parametru θ, być może całkowicie losowe.
   Przypisz $t := 1$.
1. Wylosuj parę $\langle x_t, y_t \rangle \sim P_{x,y}$.
2. Oblicz kolejne przybliżenie θ_t:

$$\theta_{t+1} := \theta_t - \beta_t \frac{\mathrm{d}}{\mathrm{d}\theta_t^T} \|\bar{f}(x_t; \theta_t) - y_t\|^2.$$

3. Jeśli spełnione są warunki zakończenia (np. dotyczące t lub osiągniętej jakości aproksymacji), zakończ. W przeciwnym razie przypisz $t := t + 1$ i przejdź do punktu 1.

Maszyny wektorów nośnych
Support Vector Machines – SVM



- Klasyfikator
- Na podstawie danych buduje funkcję

$$f(x) > 0 \rightarrow x \in \text{Klasa}$$
$$f(x) \leq 0 \rightarrow x \notin \text{Klasa}$$

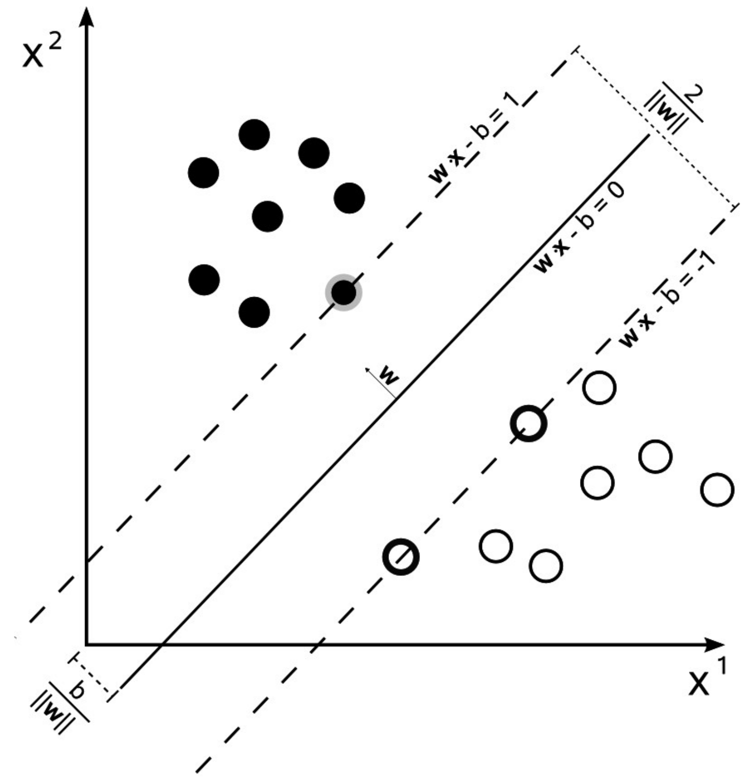
SVM – przypadek liniowo separowalny

x_i - i -ty obraz
$y_i = 1$ jeśli $x_i \in$ Klasa
$y_i = -1$ jeśli $x_i \notin$ Klasa

Funkcja rozgraniczająca
$f(x) = w^T x - b$

$(w, b) = \arg\min_{w,b} \|w\|^2$

przy ograniczeniach
$w^T x_i - b \geq 1$ dla $x_i \in$ Klasa
$w^T x_i - b \leq -1$ dla $x_i \notin$ Klasa
inaczej, przy ograniczeniach
$(w^T x_i - b) y_i \geq 1$



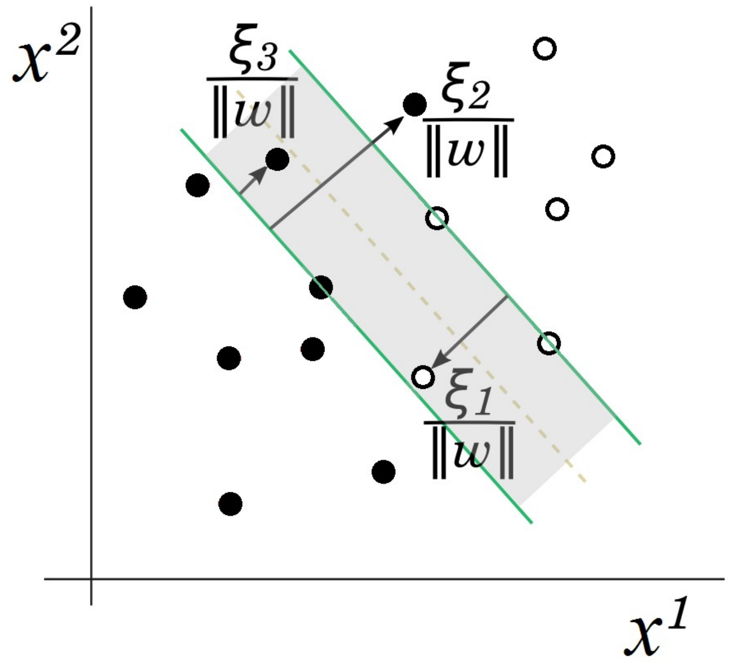
SVM – przypadek nieseparowalny liniowo

$f(x) = w^T x - b$

$(w, b) = \arg\min_{w,b} \sum_i \xi_i + \lambda \|w\|^2$

$\lambda > 0$

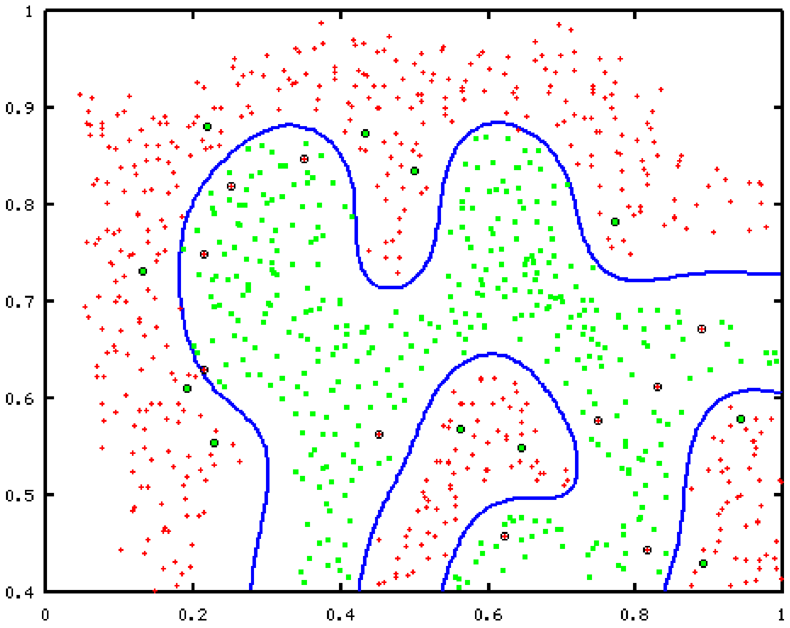
przy ograniczeniach
dla każdego i:
$\xi_i \geq 0$
$(w^T x_i - b) y_i \geq 1 - \xi_i$



Wniosek z powyższego
$\xi_i = \max\{1 - f(x_i) y_i, 0\}$

Twierdzenie o reprezentacji
$w = \sum_i \alpha_i y_i x_i$
$\alpha_i \neq 0$ tylko dla $i \in SVs$

SVM – postać nieliniowa



- Zasada taka sama, ale nowa przestrzeń

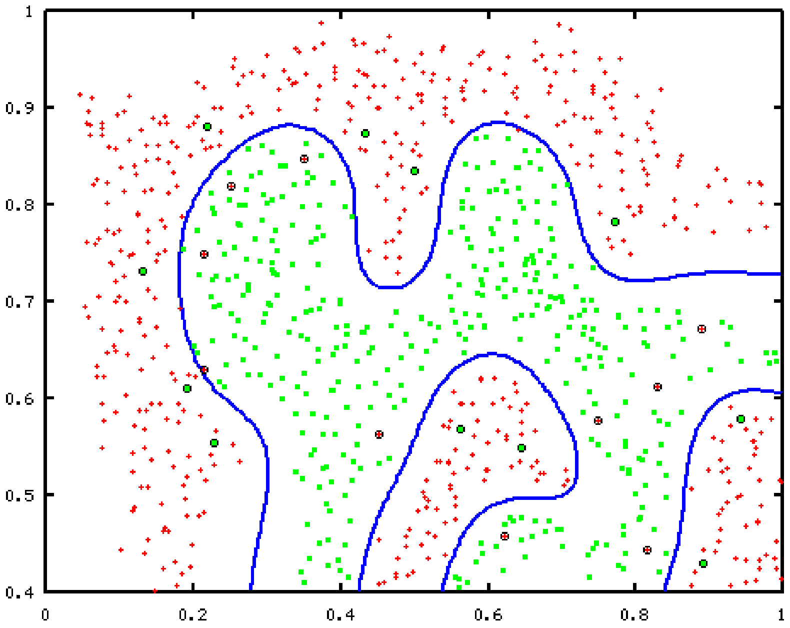
$$z = \phi(x)$$

$$w = \sum_i \alpha_i y_i \phi(x_i)$$

$$f(x) = w^T \phi(x) - b$$

$$(\alpha_{1..N}, b) = \arg\min_{\alpha_{1..N}, b} \sum_i \max\{1 - f(x_i) y_i, 0\} + \lambda \|w\|^2$$

SVM – postać nieliniowa



$$f(x) = w^T \phi(x) - b$$
$$w = \sum_i \alpha_i y_i \phi(x_i)$$
$$f(x) = \sum_i \alpha_i y_i \phi(x_i)^T \phi(x) - b$$

Jądra (*kernels*) SVM

$$\phi(x)^T \phi(y) = k(x, y)$$

liniowe: $k(x, y) = x^T y$

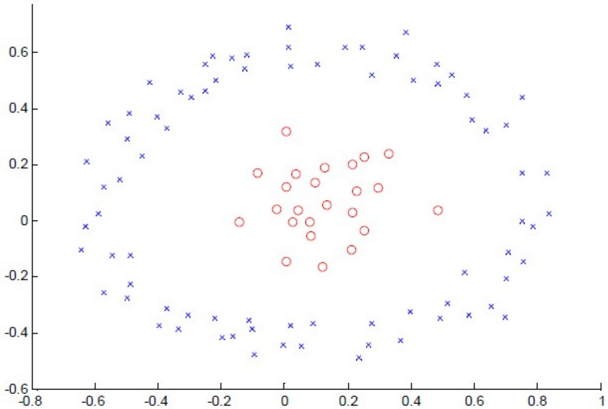
wielomianowe: $k(x, y) = (1 + x^T y)^d, d > 0$

gaussowskie (RBF): $k(x, y) = \exp\left(-\|x - y\|^2 / 2\sigma^2\right)$

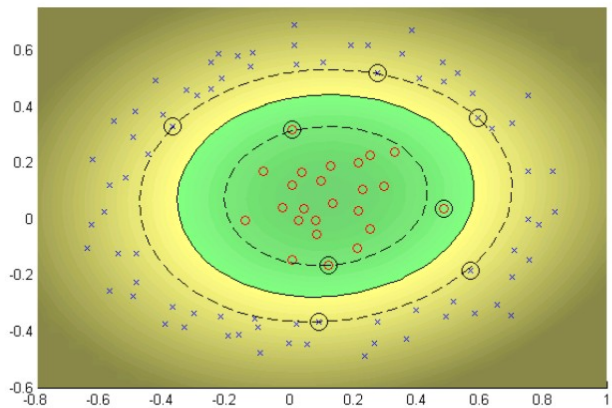
SVM – jądro RBF

$$(\alpha_{1..N}, b) = \underset{\alpha_{1..N}, b}{\arg\min} \sum_i max\{1 - f(x_i)\, y_i, 0\} + \lambda \|w\|^2$$

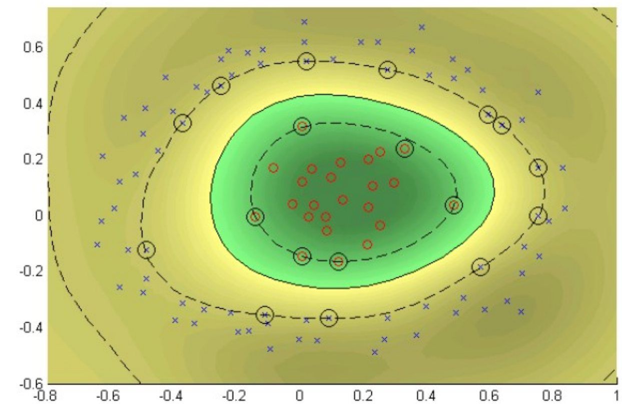
$$f(x) = \sum_i \alpha_i y_i \exp(-\|x_i - x\|^2 / 2\sigma^2)$$



$\sigma = 1, \lambda = 10$

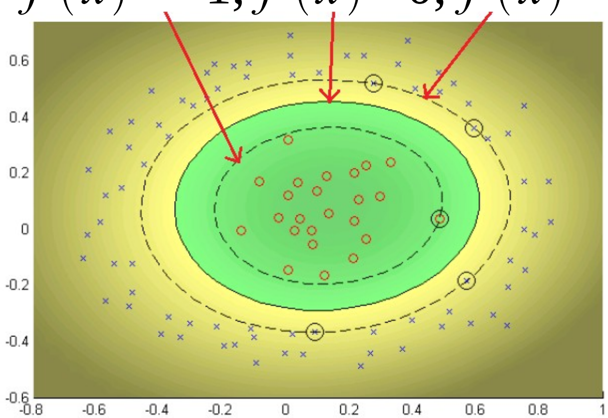


$\sigma = 0.25, \lambda \approx \infty$

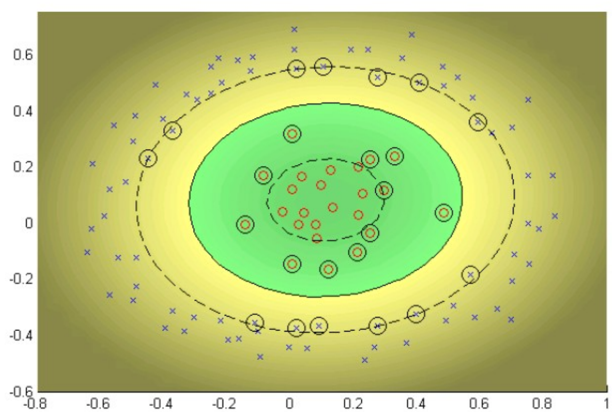


$\sigma = 1, \lambda = 1$

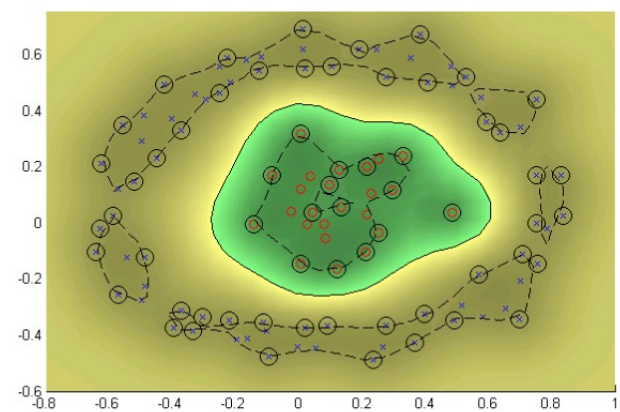
$f(x) = -1, f(x) = 0, f(x) = 1$



$\sigma = 1, \lambda = 100$



$\sigma = 0.1, \lambda \approx \infty$



Inne ważne klasyfikatory

- Drzewa decyzyjne
- Lasy losowe
  ← modele zespołowe/komitetowe

Gradient Boosting – idea

- Zadanie minimalizacji straty

  $\langle x_i, q_i \rangle, \;\; i \in \{1, \ldots, N\}, \;\; q_i : R^{n_y} \to R, \;$ np. $\; q_i(y) = \| y - y_i \|^2$
- Modele

  $$\bar{f}_m : R^{n_x} \to R^{n_y}, \;\; \gamma_m \in R, \;\; F_m(x_i) = \sum_m \gamma_m \bar{f}_m(x_i)$$
- Szukamy modelu minimalizującego stratę

  $$\frac{1}{N} \sum_{i=1}^{N} q_i(F_m(x_i))$$
- W pętli:
  - Kolejne $\bar{f}_m$ poprawia błędy dotychczasowego modelu

Gradient Boosting – algorytm

1: Inicjalizacja wartością stałą

$$F_0(x) \equiv \arg\min_y \sum_{i=1}^{N} q_i(y).$$

2: Dla $m=1$ do M:

2.1. Oblicz pseudo-rezidua:

$$r_{i,m} = -\left[\frac{\partial q_i(F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}\right], \ i=1,\ldots,n, \ \text{np. } r_{i,m} = 2(y_i - F_{m-1}(x_i))$$

2.2. Naucz $\bar{f}_m$ używając $\langle x_i, r_{i,m}\rangle, i=1,\ldots,N$

jako zbioru treningowego

2.3. Oblicz γ_m

$$\gamma_m = \arg\min_\gamma \sum_{i=1}^{N} q_i(F_{m-1}(x_i) + \gamma \bar{f}_m(x_i)).$$

2.4. $F_m(x) = F_{m-1}(x) + \gamma_m \bar{f}_m(x)$

3. Zwróć $F \equiv F_M$

XGBoost – biblioteka

- eXtreme Gradient Boosting
- Algorytm: Gradient Boosting
- $\bar{f}_m$ mają postać drzew
- Do ściągnięcia z github-a
- Projekt rozpoczęty przez Tianqi Chen'a z Distributed Machine Learning Community
- Często wygrywa konkursy na Kaggle.com
- „When in doubt, use xgboost”

Jak wybrać najlepszy model?
k-krotna walidacja krzyżowa

1. Sortujemy zbiór T w losowej kolejności
2. Dzielimy T na k równych części: $T_1 \cup \ldots \cup T_k = T$
3. Dla $i = 1, \ldots, k$:

3.1. Uczymy model na zbiorze $T \setminus T_i$

3.2. Rejestrujemy średnią stratę $\bar{q}_{T_i}$ na zbiorze T_i

4. Uczymy model na zbiorze T
5. Przyjmujemy średnią stratę tego modelu $\bar{q} \approx (1/k) \sum_{i=1}^{k} \bar{q}_{T_i}$