Politechnika Warszawska

MSI

10. Uczenie się aproksymacji funkcji

Włodzimierz Kasprzak

Treść

1. Zadanie aproksymacji / regresji
2. Metodyka uczenia się aproksymacji
3. Aproksymacja parametryczna
4. Regresja liniowa
5. Przykład: regresja w klasyfikacji
6. Regresja nieliniowa
7. Regresja wykładnicza i logarytmiczna
8. Model pamięciowy aproksymacji
9. SVR

1. Zadanie aproksymacji / regresji

Uczenie aproksymacji funkcji jest kolejną, po uczeniu klasyfikacji i uczeniu pojęć, odmianą uczenia indukcyjnego.

Zadanie aproksymacji

Poszukujemy reprezentacji nieznanej funkcji,

$$f(\mathbf{x}): \mathbf{X} \rightarrow \Re,$$

o wartościach rzeczywistych. Wyznaczyć należy taką funkcję,

$$h(\mathbf{x}; \mathbf{p}), \text{ z parametrami } \mathbf{p} = [p_1 , p_2 , \dots , p_n]^{\mathrm{T}},$$

której wartości w badanej dziedzinie $\mathbf{X}$ dostatecznie mało różnią się, w sensie przyjętego kryterium, od odpowiednich wartości funkcji $f(\mathbf{x})$.

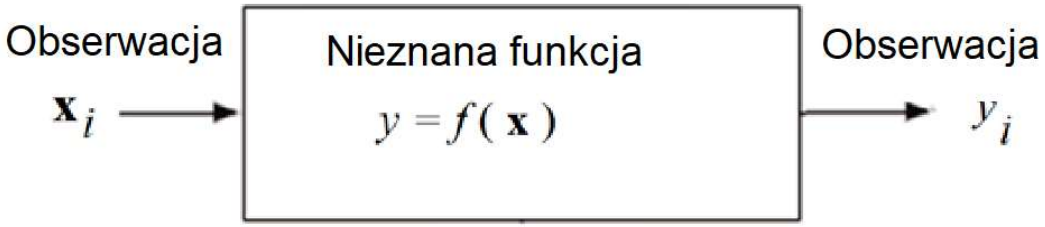
Informacja trenująca

Próbka ucząca to para $(\mathbf{x}, f(\mathbf{x}))$ dla $\mathbf{x} \in \mathbf{X}$.

Aproksymacja funkcji metodą regresji

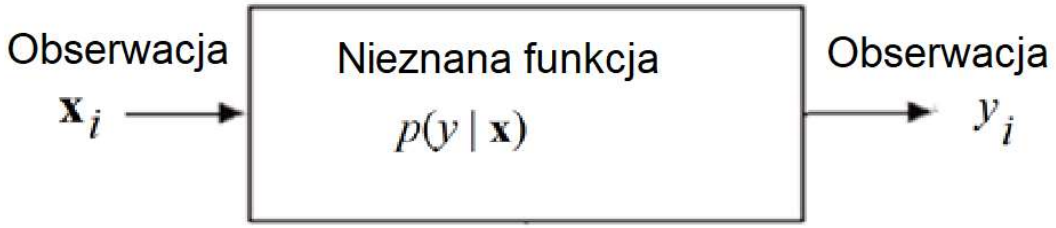
1) Metodyka deterministyczna: aproksymacja funkcji wiążącej dwie **obserwowane** zmienne y i $\mathbf{x}$:

$y = f(\mathbf{x})$, gdy danych jest N obserwacji, $(\mathbf{x}_i , y_i)$, i=1, 2, …, N .



W metodach **regresji** - rodzina funkcji jest znana, np. funkcja liniowa, wielomian, sigmoid, itp.; należy wyznaczyć parametry funkcji dla uzyskania konkretnej funkcji.

2) Metodyka stochastyczna: aproksymacja funkcji gęstości prawdopodobieństwa dwóch obserwowanych zmiennych losowych y i $\mathbf{x}$: $p(y \mid \mathbf{x})$, gdy danych jest N obserwacji, $(\mathbf{x}_i , y_i)$, i=1, 2, …, N .



Ocena aproksymacji

Kryterium oceny aproksymacji

Dla większości algorytmów jako kryterium oceny wygodne jest stosowanie błędu średniokwadratowego zdefiniowanego jako:

$$e_p = \frac{1}{N} \sum_{x_i \in \mathrm{T}} (f(\mathbf{x}_i) - h(\mathbf{x}_i, \mathbf{p}))^2$$

gdzie N oznacza dalej liczbę próbek uczących w zbiorze $\mathbf{T}$.

Rola aproksymowanej funkcji

Najczęściej aproksymowana funkcja pełni następnie rolę funkcji decyzyjnej lub funkcji potencjału w zagadnieniach klasyfikacji i charakteryzuje ona stopień przynależności danej próbki do zadanej klasy.

2. Metodyka uczenia się aproksymacji funkcji

Niech f oznacza nieznaną nam docelową funkcję.

Znane są **przykłady** (obserwacje) tej funkcji dla zadanego wejścia x – czyli pojedyncza obserwacja (**próbka ucząca**) to para $(x, f(x))$.

Zadanie aproksymacji w uczeniu przez indukcję jest:

mając dany zbiór próbek uczących znaleźć funkcję **(hipotezę)** h, aproksymującą nieznaną funkcję, tzn. $h \approx f$.

Jak zmierzyć jakość hipotezy, tzn. bliskość hipotezy i rzeczywistej funkcji, tzn. czy $h \approx f$?

Ocena jakości hipotezy

Jakość hipotezy h można wyrazić poprzez ocenę jej zdolności do generalizacji, tzn. tego, jak dobrze przewiduje ona wartości funkcji f dla nieznanych dotąd obserwacji.

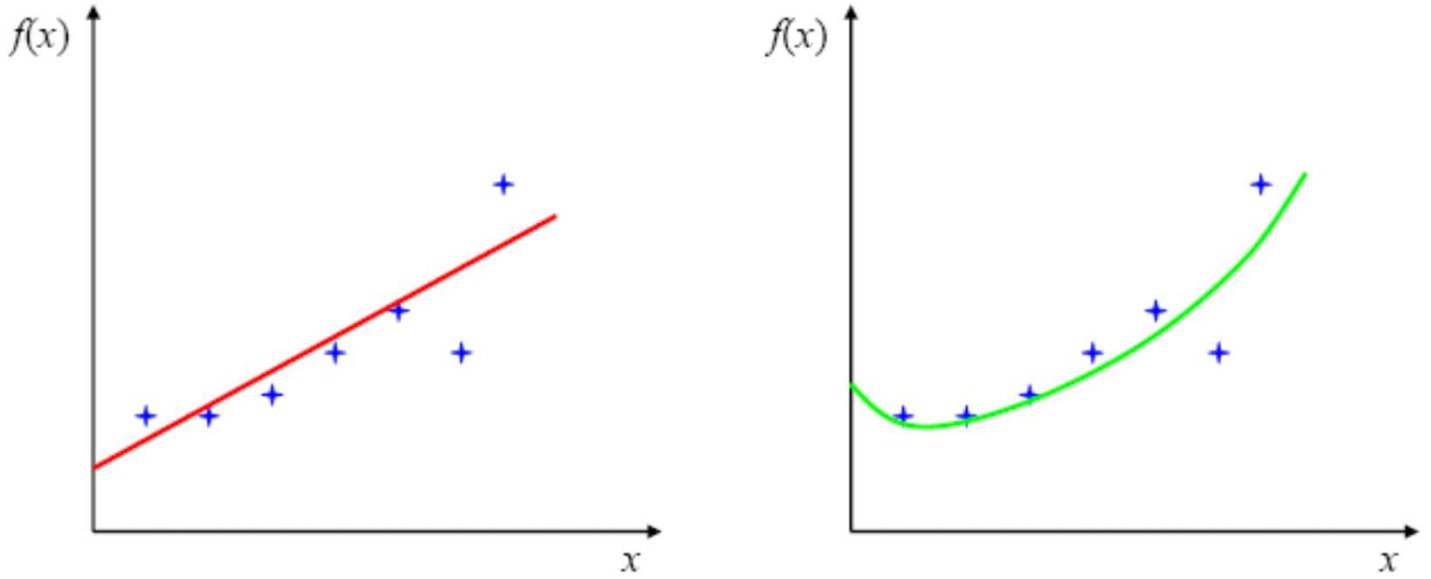
Ocena jakości hipotezy

Wykonując testy zgodności hipotez uzyskanych dzięki uczeniu na zbiorach próbek uczących o różnych rozmiarach możemy przedstawić jakość hipotez w postaci krzywej.

Krzywa zgodności hipotez: wykres przedstawia przykładową procentową zgodność hipotezy jako funkcję rozmiaru zbioru próbek uczących.

Przykład aproksymacji

Aproksymacja zbioru punktów na płaszczyźnie linią prostą lub krzywą (funkcja 1-wymiarowa).

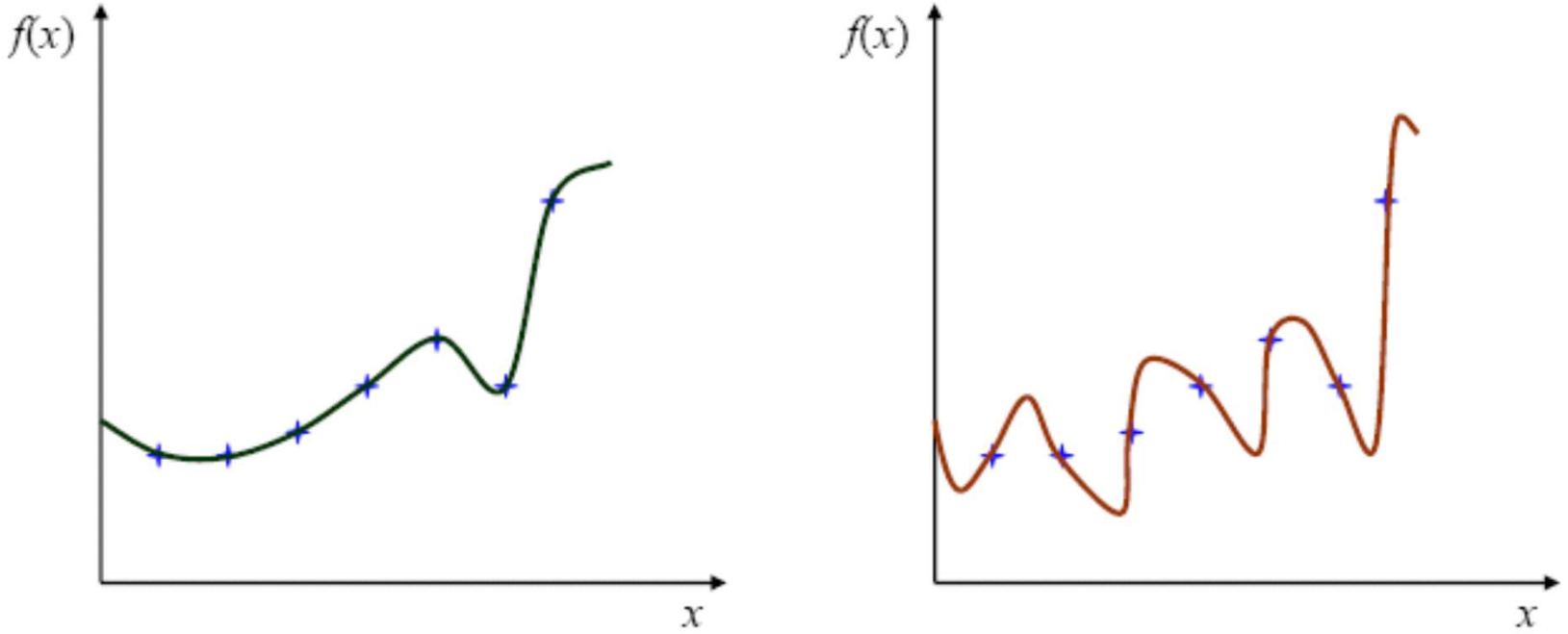


Hipotezy: funkcja liniowa i wielomian 2-go stopnia

Hipoteza jest **zgodna z próbkami** uczącymi, jeśli wszystkie je spełnia. Powyższe hipotezy nie są zgodne z próbkami. Dlatego kontynuujemy szukanie hipotezy o postaci wielomianu wyższego rzędu.

Wybór hipotezy

Może być wiele funkcji zgodnych z próbkami uczącymi. Kierujemy się wtedy zasadą wyboru funkcji najprostszej spośród zgodnych funkcji:



Dwie funkcje wielomianowe zgodne z próbkami uczącymi. Lewa funkcja jest prostsza i powinna być wybrana jako hipoteza.

3. Aproksymacja parametryczna (regresja)

W klasycznym zadaniu **aproksymacji parametrycznej** poszukiwana funkcja ma postać parametryczną, zdefiniowaną nad przestrzenią cech jako:

$$h(\mathbf{x},\mathbf{p}) = \mathbf{p}^T \varphi(\mathbf{x}), \quad \mathbf{p} \in \Re^m, \varphi(\mathbf{x}) \in \Re^m$$

Każdy przykład (próbka ucząca), $x \in X$, jest reprezentowany przez wektor cech, $c = \varphi(x)$, gdzie $\varphi = [\varphi_0, \ldots, \varphi_m]^T$ jest wektorem funkcji wyznaczających wartości cech.

Dla każdej klasy funkcja $h()$ ma tę samą postać, a różni się jedynie unikalnym wektorem wartości parametrów $\mathbf{p}$. W istocie jest to funkcja liniowa nad przestrzenią cech, gdzie jednak tzw. **funkcje bazowe**, $\varphi_i(x)$, są potencjalnie nieliniowe.

Zadanie aproksymacji funkcji parametrycznej:

1. poszukiwanie zestawu funkcji bazowych $\varphi(x)$,
2. minimalizacja błędu e_p względem wektora parametrów $\mathbf{p}$.

Liniowe i nieliniowe funkcje bazowe

Liniowa funkcja bazowa przedstawia bezpośrednią zależność wektora cech od próbki:

$$\mathbf{c} = \varphi(\mathbf{x}) = [1, x_1, x_2, \cdots, x_m]^T$$

gdzie $\mathbf{c}$ jest wektorem o $(m+1)$ elementach.

Przykład nieliniowej funkcji bazowej to **funkcja kwadratowa** (wielomian rzędu 2) o m zmiennych (składowych $\mathbf{x}$):

$$\varphi(\mathbf{x}) = [1, x_1, x_2, \cdots, x_n, x_1x_1, x_2x_1, \cdots, x_nx_n]^T$$

Teraz $\mathbf{c}$ będzie wektorem $(1+m + m(m+1)/2)$-elementowym i tyle też potrzebnych będzie parametrów w wektorze $\mathbf{p}$.

Nieliniowe funkcje bazowe wyższego rzędu w ogólności również mogą zależeć od zbioru parametrów, tak jak poszukiwana aproksymacja funkcji $f(\mathbf{x})$.

4. Regresja liniowa

Błąd średniokwadratowy e_p to funkcja kwadratowa względem nieznanych parametrów, więc dla **liniowej funkcji** $h(\boldsymbol{x}, \boldsymbol{p})$

$$h(\boldsymbol{x}, \boldsymbol{p}) = p_0 + \sum_{i=1}^{m} p_i \cdot x_i$$

problem znalezienia optymalnych parametrów ma jedno rozwiązanie – błąd osiąga wartość minimalną w miejscu zerowania się pochodnych cząstkowych względem poszczególnych parametrów.

Dla jednego parametru mamy: $\dfrac{\mathrm{d}\, e_p}{\mathrm{d}\, p} = 0$

i stąd wyznaczamy p.

Dla aproksymacji **liniowej** funkcji **wielowymiarowej** h rozwiążemy układ m równań o m niewiadomych p_i :

$$\frac{\mathrm{d}\, e_p}{\mathrm{d}\, p_i} = 0\,,\ i = 0,1,\ldots,m$$

Regresja liniowa (2)

Przykłady trenujące nie musza leżeć na jednej prostej (płaszczyźnie, hiperpłaszczyźnie) w przestrzeni $\mathbf{X}$, ale mimo to próbujemy jednak aproksymować je funkcją liniową.

Niech danych jest N próbek uczących w zbiorze uczącym $\mathbf{T}$.

Można ułożyć układ równań o (N) wierszach i $(m+1)$ kolumnach (dla $m+1$ parametrów w wektorze $\mathbf{p}$).

Uzupełniamy każdą próbkę $\mathbf{x}$ o zerową składową $(x_0 = 1)$ i otrzymujemy macierz współczynników próbek:

$$\mathbf{T} = \begin{pmatrix} 1 & \mathbf{x}_1^T \\ 1 & \mathbf{x}_2^T \\ \vdots & \vdots \\ 1 & \mathbf{x}_{N-1}^T \\ 1 & \mathbf{x}_N^T \end{pmatrix} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ 1 & x_{21} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & & \vdots \\ 1 & x_{N-1,1} & x_{N-1,2} & \cdots & x_{N-1,m} \\ 1 & x_{N,1} & x_{N,2} & \cdots & x_{N,m} \end{pmatrix}$$

Regresja liniowa (3)

Wektor oczekiwanych wyników funkcji $h(\mathbf{x})$ dla próbek uczących oznaczmy przez $\mathbf{d}$.

Wektor błędów składowych (liczony po parametrach) dla aproksymacji funkcji wyznaczamy jako:

$$\boldsymbol{\varepsilon} = \mathbf{d} - \mathbf{T}\,\mathbf{p}$$

Średni błąd kwadratowy wynosi: $\| e \|^2 = \frac{1}{N} \sum_{i=1}^{N} \varepsilon_i^2$

Teraz należy znaleźć minimum powyższej funkcji błędu, stosując metodę najmniejszych kwadratów (MNK), przyrównując pochodne cząstkowe do zera i rozwiązując tak wyznaczony układ równań.

MNK (*ang. LSE*)

Metoda Najmniejszych Kwadratów (ang. LSE, *least square error*)

Rozpatrzmy najpierw przypadek 2-wymiarowy – celem jest znalezienie liniowego odwzorowania pomiędzy dwiema zmiennymi y i x, o postaci: $y = a\,x + b$ lub $a\ x\ +\ b\ -\ y\ =\ 0,$ gdy istnieje N obserwacji (próbek): (x_i, y_i), i=1, 2,…, N

Tworzymy układ N równań:

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

a dokładnie:

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{bmatrix}$$

Optymalizowany błąd: $U(a,b) = \varepsilon_1^2 + \varepsilon_2^2 + \ldots + \varepsilon_N^2 = \sum \varepsilon_i^2 = |\,\boldsymbol{\varepsilon}\,|^2$

Szukane parametry to $\boldsymbol{p} = [a,\ b]$.

MNK (2 parametry)

Minimum funkcji błędu $U(a, b)$:

$$\frac{\partial U}{\partial a} = 0 \quad , \quad \frac{\partial U}{\partial b} = 0 \quad .$$

Ogólna postać rozwiązania układu równań:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{N\sum x_i^2 - (\sum x_i)^2} \cdot \begin{bmatrix} N & -\sum x_i \\ -\sum x_i & \sum x_i^2 \end{bmatrix} \cdot \begin{bmatrix} \sum (x_i\, y_i) \\ \sum y_i \end{bmatrix}$$

Znane jest też ogólne, analityczne rozwiązanie układu równań w metodzie MNK dla dowolnej liczby m parametrów liniowej funkcji stanowiącej aproksymację N obserwacji $(m\text{-}1)$ wymiarowej przestrzeni - jest to **pseudo-odwrotność Moore-Penrose.**

Macierz pseudo-odwrotna Moore-Penrose

Ogólna postać rozwiązania układu N równań o n parametrach (niewiadomych) tworzących wektor $\mathbf{p}$:

$$\mathbf{b} = \mathbf{A}\,\mathbf{p}$$

gdzie

$$\mathbf{b} \in \mathrm{R}^{N},\ \mathbf{A} \in \mathrm{R}^{N \times n},\ \mathbf{p} \in \mathrm{R}^{n}$$

Ogólną postacią rozwiązania jest: $\mathbf{p} = \mathbf{A}^{\dagger}\,\mathbf{b}$

gdzie $\mathbf{A}^{\dagger}$ jest macierzą pseudo-odwrotną Moore-Penrose.

Jeśli $\mathbf{A}$ jest nieosobliwa to $\mathbf{A}^{\dagger}$ może zostać wyznaczona jako:

- gdy $N = n$: $\mathbf{A}^{\dagger} = \mathbf{A}^{-1}$ - jest zwykłą macierzą odwrotną;
- gdy $N > n$: stanowi rozwiązanie **problemu MNK** (minimalizacji błędu $|\mathbf{b} - \mathbf{A}\,\mathbf{p}|^2$) i jest postaci $\mathbf{A}^{\dagger} = (\mathbf{A}^{\mathrm{T}}\,\mathbf{A})^{-1}\,\mathbf{A}^{\mathrm{T}}$
- gdy $N < n$: wymaga dodatkowych ograniczeń, np. dla minimalizacji normy $|\mathbf{p}|^2$: $\mathbf{A}^{\dagger} = \mathbf{A}^{\mathrm{T}}\,(\mathbf{A}\,\mathbf{A}^{\mathrm{T}})^{-1}$

5. Przykład: regresja w klasyfikacji

Dana jest **rodzina parametrycznych funkcji** reprezentujących rozkład ***energii*** *każdej klasy (pojęcia)* w przestrzeni cech:

$$d_i(\boldsymbol{c},\boldsymbol{a}) = \{\boldsymbol{a}_i^T \varphi(\boldsymbol{c}) \,|\, \boldsymbol{a}_i \in \Re^{(n+1)}\}, (i = 1,\cdots,m)$$

Gdzie: m - liczba klas, $\boldsymbol{c}$ – wektor cech, $\boldsymbol{a}_i$ – wektor parametrów funkcji potencjału dla „i”-tej klasy, $\varphi(\ \boldsymbol{c}\ )$ – odwzorowanie wektora cech zależne od typu funkcji potencjału.

Liniowa funkcja energii (potencjału) klasy – w tym przypadku odwzorowanie wektora cech ma na celu jedynie zwiększenie wymiaru wektora o wyraz wolny 1: $\varphi(c) = (1, c_1, c_2, \cdots, c_n)^T$

Klasyfikator według liniowych funkcji potencjału, przypadek 2 klas

Funkcje: $d_1(\mathbf{c},\mathbf{a}^{(1)}) = a_0^{(1)} + \sum_{i=1}^{n} a_i^{(1)} \cdot c_i \qquad d_2(\mathbf{c},\mathbf{a}^{(2)}) = a_0^{(2)} + \sum_{i=1}^{n} a_i^{(2)} \cdot c_i$

Reguła decyzyjna **klasyfikatora według funkcji potencjału**:

$$d(\ \boldsymbol{c}\ ) = d_1(\boldsymbol{c}, \boldsymbol{a}^{(1)}) - d_2(\boldsymbol{c}, \boldsymbol{a}^{(2)}),$$

$$\text{IF} \quad d(\ \boldsymbol{c}\ ) \geq \mathbf{0} \ \text{ THEN klasa 1 ELSE klasa 2}$$

Funkcja wielomianowa

Gdy funkcja potencjału ma postać wielomianu zmiennych składowych wektora cech, rozmiar wektora a wzrasta odpowiednio.

Np. dla funkcji kwadratowej (wielomian rzędu 2) n zmiennych (składowych wektora cech) $c_1, c_2, \cdots, c_n$:

$$\varphi(c) = (1, c_1, c_2, \cdots, c_n, c_1c_1, c_2c_1, \cdots, c_nc_n)^T$$

tzn. $\boldsymbol{a}$ jest wtedy wektorem $(1+n + n(n+1)/2)$-elementowym.

Np. dla n=2 wielomian jest postaci:

$$d(c, \boldsymbol{a}) = a_0 + a_1c_1 + a_2c_2 + a_3c_1c_2 + a_4{c_1}^2 + a_5{c_2}^2$$

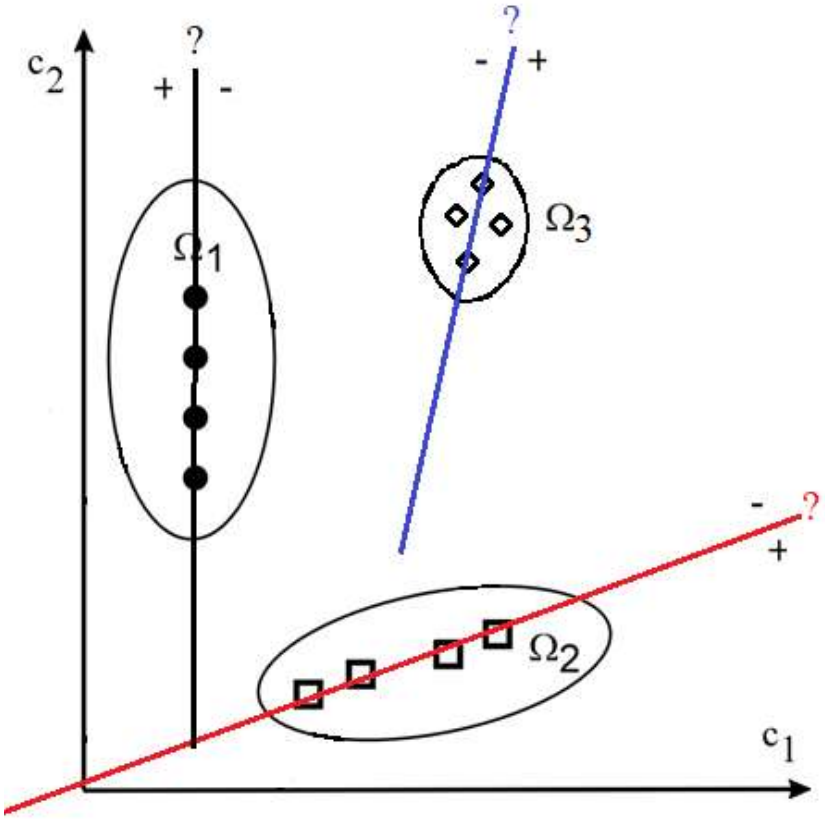
Jakie zbiory uczące? (1)

Przykład: 3 klasy w przestrzeni cech 2-D.

Strategia uczenia 1:

Parametry funkcji każdej klasy uczone są niezależnie od siebie (brane są pod uwagę jedynie próbki pozytywne dla każdej klasy).

Niedokładna separacja obszarów klas:

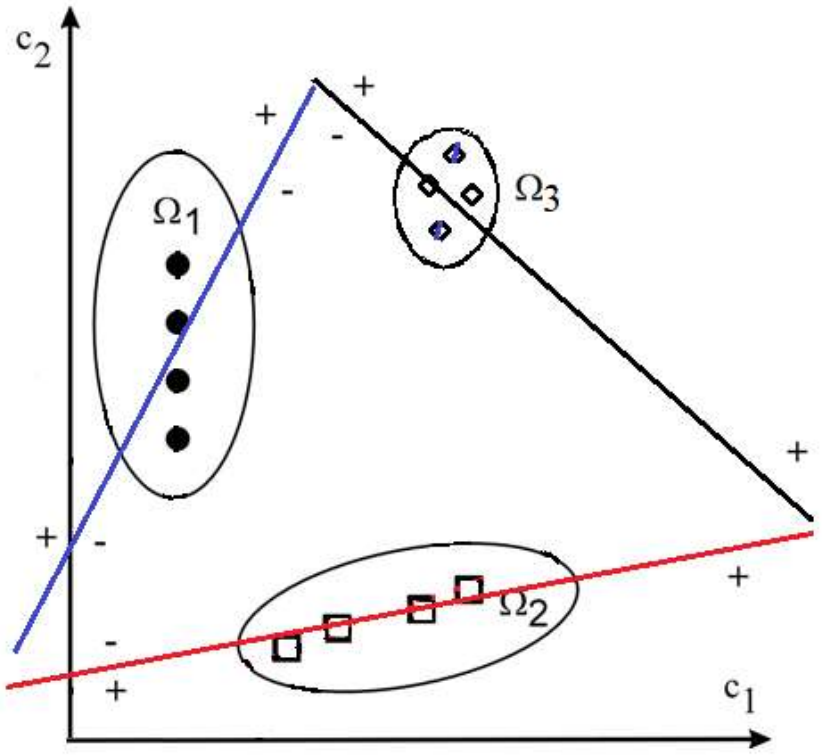


Jakie zbiory uczące ? (2)

Strategia uczenia 2

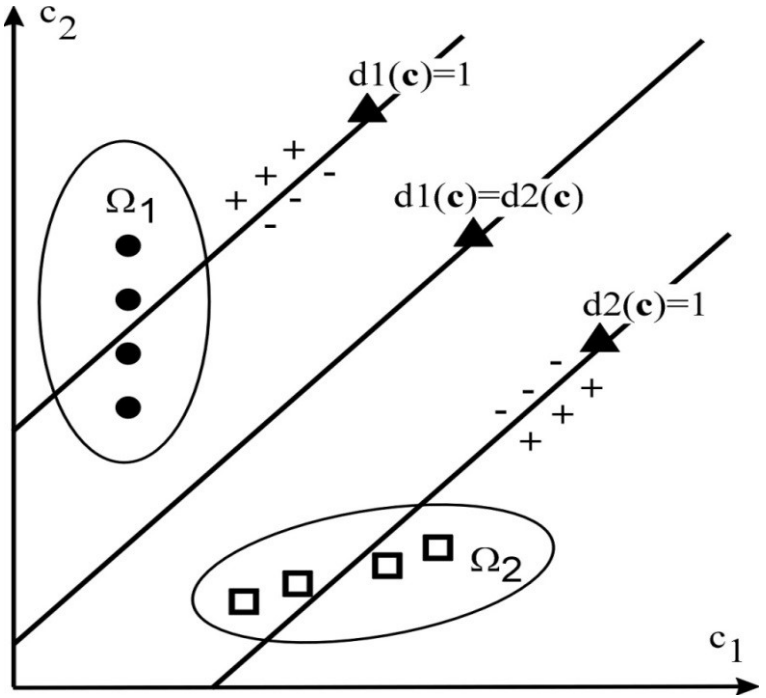
Uczymy osobno dla każdej klasy ale bierzemy pod uwagę zarówno próbki pozytywne (tej klasy) jak i negatywne – czyli próbki pozostałych klas.

Przykład: proste $d_k(c) = 1$ (k=1,2, …,K) w ogólności nie będą równoległe do siebie.



Przykład: 2 klasy, strategia 2

Przykład liniowych funkcji potencjału klasyfikatora binarnego (2 klasy) w przestrzeni cech 2D. Każdy rozkład potencjału ma postać płaszczyzny odpowiednio nachylonej względem przestrzeni cech – pokazano jedynie kilka prostych dla punktów o identycznym potencjale. Dla przypadku 2 klas proste obu klas są do siebie równoległe i mamy linię prostą dla której $d_1(\boldsymbol{c}) = d_2(\boldsymbol{c})$. Dla większej liczby klas zarówno proste równego potencjału nie będą równoległe jak i linia podziału nie będzie linią prostą.



Przykład: *K* klas, strategia 2

Klasyfikator z liniowymi funkcjami potencjału dla K klas:

$$\varsigma(c) = \arg\max_k d_k(c, a^{(k)}) = \arg\max_k (a_0^{(k)} + \sum_{i=1}^{n} a_i^{(k)} \cdot c_i)$$

Uczenie

1. Niech $\mathbf{C}$ będzie macierzą obserwowanych cech (próbek uczących) o rozmiarze $N \times (n+1)$, tzn. złożoną z N próbek o wymiarze n . Niech N_k próbek jest z klasy k.
2. Niech $\mathbf{d}$ będzie wektorem N-elementowym, którego elementy wskazują na przynależność odpowiadających im próbek (wierszy macierzy C) do klas.
3. Dla każdej klasy Ω_k definiujemy układ równań:

$$\boldsymbol{\varepsilon} + \mathbf{d}^{(k)} = \mathbf{C}\, \boldsymbol{a}^{(k)},$$

gdzie $\mathbf{d}^{(k)} = [d^{(k)}_1, d^{(k)}_2, \ldots, d^{(k)}_N]^{\mathrm{T}}$,

$d_k(\boldsymbol{c}, \boldsymbol{a}^{(\mathrm{k})}) = 1$, gdy $\boldsymbol{c} \in \Omega_k$ lub $d_k(\boldsymbol{c}, \boldsymbol{a}^{(\mathrm{k})}) = -1$, gdy $\boldsymbol{c} \notin \Omega_k$;

$\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N]^{\mathrm{T}}$ jest wektorem błędów.

Przykład: regresja MNK (*LSE*)

4. Mamy do rozwiązania problem optymalizacji „liniowego LSE” (*least square error* – minimalizacja błędu kwadratowego) – metodą najmniejszych kwadratów (MNK) dla każdej klasy:

$$\mathbf{a}_{opt} = \arg\min_{\mathbf{a}} \sum_{i=1}^{N} \varepsilon_i^2(\mathbf{a})$$

Jak wiemy rozwiązanie powyższego problemu prowadzi do nowego układu równań:

$$\mathbf{X}^{(k)} = \mathbf{B}^{(k)} \; \boldsymbol{a}_{opt}^{(k)},$$

gdzie

$$\mathbf{X}^{(k)} = \mathbf{C}^{(k)\mathrm{T}} \, \mathbf{d}^{(k)}, \qquad \mathbf{B}^{(k)} = \mathbf{C}^{(k)\mathrm{T}} \, \mathbf{C}^{(k)} \; .$$

5. Powyższy układ rozwiązywany jest analitycznie stosując pseudo-odwrotną macierz Moore-Penrose:

$$\boldsymbol{a}_{opt}^{(k)} = (\mathbf{C}^{(k)\mathrm{T}} \, \mathbf{C}^{(k)})^{-1} \, \mathbf{C}^{(k)\mathrm{T}} \, \mathbf{d}^{(k)} \; .$$

6. Regresja nieliniowa

Jeśli funkcje f, h **nie są liniowe** to metoda MNK bezpośrednio nie prowadzi do układu równań liniowych. Wtedy stosujemy metody iteracyjnego poprawiania oszacowania parametrów $\mathbf{p}(\mathrm{k})$ przesuwając się wzdłuż kierunku ujemnego gradientu funkcji błędu (przesuwamy się w przestrzeni parametrów).

Jest to tzw. **uogólniona reguła delta**: $\mathbf{p}(k+1) = \mathbf{p}(k) + \Delta\mathbf{p}$,

gdzie $\Delta\mathbf{p} = -\beta \cdot \frac{1}{2}\nabla_p e_p$ jest wektorem pochodnych cząstkowych funkcji błędu: $\nabla_p e_p = [\frac{\partial e_p}{\partial p_0}, \frac{\partial e_p}{\partial p_1}, \cdots, \frac{\partial e_p}{\partial p_m}]$, $\beta \in (0,1]$.

W k-tej iteracji wyznaczamy wektor modyfikacji $\Delta\mathbf{p}$ jako:

$\Delta\mathbf{p} = \beta \frac{1}{N} \sum_{x_i \in \mathrm{T}} (f(\mathbf{x}_i) - h(\mathbf{x}_i, \mathbf{p}(k))) \cdot \nabla_p h\big(x_i, p(k)\big)$ (dla zbioru N próbek)

lub: $\Delta\mathbf{p}(k+1) = \beta\, [f(\mathbf{x}_i) - h\big(\mathbf{x}_i, \mathbf{p}(k)\big)] \cdot \nabla_p h(x_i, p(k))$ (gdy modyfikacja następuje po każdej próbce)

Nieliniowa MNK (*LSE*)

Danych jest N próbek : $(x_i, y_i) \in \mathbf{R}^2$

Relację pomiędzy zmiennymi x i y chcemy aproksymować parametryczną nieliniową funkcją,

$y = f(x \mid \mathbf{p})$, gdzie $\mathbf{p} \in \mathbf{R}^n$ $(N \geq n)$ jest wektorem parametrów.

Celem jest znalezienie parametrów zapewniających minimalizację **błędu średniokwadratowego :**

$$\Phi(\mathbf{p}) = \frac{1}{2}\mathbf{r}^2(\mathbf{p}) = \frac{1}{2}\sum_{i=1}^{N} r_i{}^2(\mathbf{p}) = \frac{1}{2}\sum_{i=1}^{N}[y_i - f(x_i \mid \mathbf{p})]^2$$

Gradient funkcji błędu względem nieznanych parametrów:

$$\nabla\Phi(\mathbf{p}) = \sum_{i=1}^{N} r_i(\mathbf{p})\,\nabla r_i(\mathbf{p}) = \mathrm{J}(\mathbf{p})^T\,\mathbf{r}(\mathbf{p})$$

gdzie $\mathrm{J}(\mathbf{p})$ jest macierzą Jakobianu: $\left[\,\mathrm{J}(\mathbf{p})\,\right]_{ij} = \left[\,\dfrac{\partial r_i(\mathbf{p})}{\partial p_j}\,\right]$

Nieliniowa MNK (2)

Macierz Hesianu tworzą pochodne drugiego rzędu:

$$\mathbf{H}(\mathbf{p}) = \nabla^2 \Phi(\mathbf{p}) = \mathrm{J}(\mathbf{p})^T \, \mathrm{J}(\mathbf{p}) + \sum_{i=1}^{N} r_i(\mathbf{p}) \, \nabla^2 r_i(\mathbf{p})$$

Dla małych błędów $\mathbf{r}$ może być ona aproksymowana jako:

$$\nabla^2 \Phi(\mathbf{p}) = \mathrm{J}(\mathbf{p})^T \, \mathrm{J}(\mathbf{p})$$

Iteracyjne rozwiązania problemu nieliniowej regresji (nieliniowa metoda MNK):

1) Gradient descent search – reguła modyfikacji $\mathbf{p}$ korzysta jedynie z pochodnych 1-szego rzędu:

$$\mathbf{p}_{i+1} = \mathbf{p}_i - \lambda \, \nabla \Phi(\mathbf{p}_i)$$

2) Metoda Gaussa-Newtona - uwzględnia pochodne 2-ego rzędu:

$$\mathbf{p}_{i+1} = \mathbf{p}_i - (\nabla^2 \Phi(\mathbf{p}_i))^{-1} \, \nabla \Phi(\mathbf{p}_i)$$

3) Metoda Levenberga-Marquardta – reguła L-M modyfikacji $\mathbf{p}$

$$\mathbf{p}_{i+1} = \mathbf{p}_i - (\mathbf{H}(\mathbf{p}_i) + \lambda \, diag(\mathbf{H}(\mathbf{p}_i)))^{-1} \, \nabla \Phi(\mathbf{p}_i)$$

Levenberg-Marquardt

Pojedyncza iteracja w algorytmie Levenberga-Marquardta:

a) Wyznacz $\boldsymbol{p}_{i+1}$ stosując regułę modyfikacji L-M,

b) Wyznacz aktualny błąd $\mathbf{r}_{i+1} = f(\boldsymbol{p}_{i+1})$;

c) IF aktualny błąd $\mathbf{r}_{i+1}$ jest większy niż poprzedni $\mathbf{r}_i$,

THEN wykonaj krok *wstecz* do $\mathbf{p}_i$ i zwiększ λ , k krotnie (np. k=10)

ELSE kontynuuj z $\mathbf{p}_{i+1}$ i zmniejsz λ , k krotnie .

7. Regresja wykładnicza i logarytmiczna

Regresja wykładnicza służy do modelowania sytuacji, w których wzrost zaczyna się powoli, a następnie gwałtownie przyspiesza bez ograniczeń, lub w których spadek zaczyna się szybko, a następnie zwalnia, by coraz bardziej zbliżać się do zera: $y(\mathrm{x}) = ab^x$, gdzie b musi być nieujemne.

W szczególności, gdy:

- $b>1$: następuje wykładniczy wzrost;
- $0<b<1$: następuje wykładniczy spadek.

Podobnie regresja logarytmiczna jest odpowiednia w sytuacjach, w których wzrost lub spadek na początku szybko przyspiesza, a następnie zwalnia z czasem, $y(x) = a + b\ln(x)$

x musi być nieujemne. Gdy $b>0$, mamy model wzrostu, a gdy $b<0$, to model spadku.

Regresja logistyczna

Wartość funkcji logistycznej wzrasta z czasem. W pewnym momencie wzrost stopniowo zwalnia, a funkcja zbliża się do górnej granicy (wartości granicznej). Regresja logistyczna najlepiej nadaje się do modelowania zjawisk, w których istnieją ograniczenia wzrostu.

$$y(x) = \frac{c}{1 + ae^{-bx}}$$

W szczególności, gdy $b>0$, to począwszy od

$y(0)=c/(1+a)$

funkcja szybko rośnie i osiąga maksymalne tempo wzrostu dla

$y(\ln(a)/b)=c/2$.

Następnie tempo wzrostu zwalnia i funkcja dochodzi asymptotycznie do wartości $y(\infty)=c$.

8. Model pamięciowy aproksymacji

Przedstawimy teraz zupełnie odmienne podejście do uczenia się aproksymacji funkcji. Tzw. **pamięciowa metoda** uczenia się nie przetwarza przykładów trenujących a tylko te przykłady zapamiętuje.

Załóżmy istnienie zbioru próbek uczących, $\mathbf{T} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_N\}$, dla których znane są wartości nieznanej funkcji $f(\mathbf{c}_i)$.

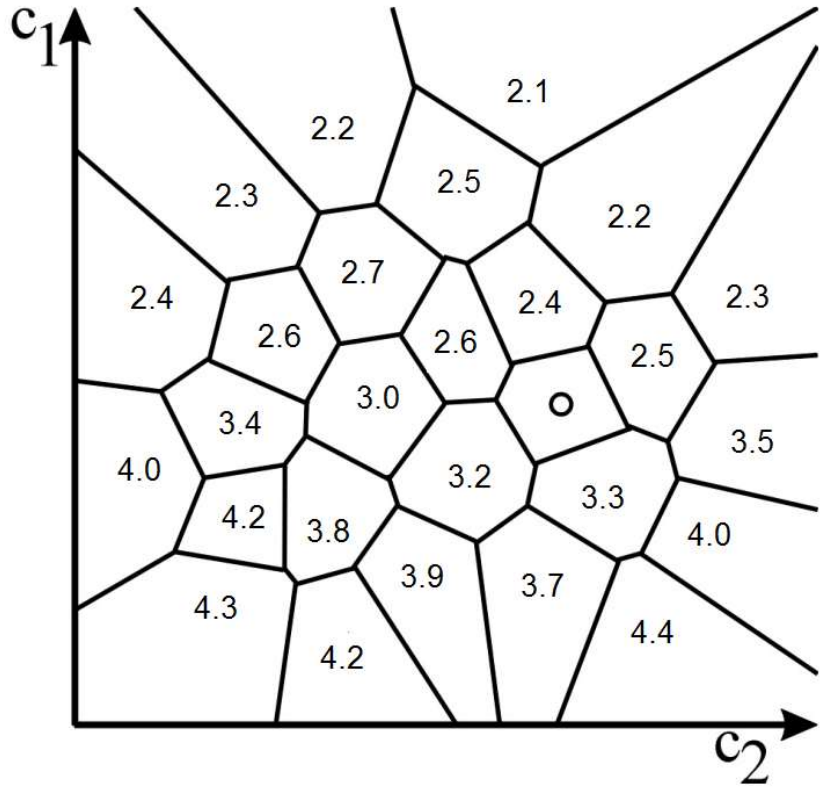
W pamięciowym modelu uczenia każda próbka zostaje zapamiętana, staje się reprezentantem funkcji a wartość jej etykiety stanowi aproksymację wartości funkcji dla „*zajmowanego przez nią*" obszaru w przestrzeni reprezentacji

Zbiór punktów przestrzeni cech, dla których najbliższym sąsiadem spośród istniejących cech jest $\mathbf{c}_i$ nazywamy ***komórką Voronoia*** dla $\mathbf{c}_i$.

Podział Voronoia

Zbiór przykładów uczących $\mathbf{T}$ wyznacza *podział Voronoia* całej przestrzeni cech.

Ilustracja podziału Voronoia dla 2-wymarowych cech.



Aproksymacja *według k sąsiadów*

Aproksymacja według *najbliższego sąsiada*

Aproksymuj wartość funkcji w punkcie $\mathbf{x}$ przez wartość pamiętanej próbki $\mathbf{c}$ położonej najbliżej zadanej wartości $\mathbf{x}$.

gdzie

$$h(\mathbf{x}) = f(\mathbf{c})$$

$$\mathbf{c} = \arg\min_{\mathbf{c}_i \in \mathrm{T}} \|\mathbf{x} - \mathbf{c}_i\|$$

Aproksymacja według *k sąsiadów*

Aproksymuj wartość funkcji w punkcie $\mathbf{x}$ przez średnią wartość k pamiętanych próbek $\mathbf{c}_i$ położonych najbliżej zadanemu punktowi $\mathbf{x}$ spośród próbek zbioru $\mathbf{T}$:

$$h(\mathbf{x}) = \frac{1}{k}\sum_{i=1}^{k} f(\mathbf{c}_i)$$

9. SVR

Zadaniem regresji **SVR** (ang. *support vector regression*) jest aproksymacja zbioru danych pomiarowych $(\boldsymbol{x}_i, y_i)$ określonych w dziedzinie liczb rzeczywistych, tzn. poza $\{\boldsymbol{x}_i\}$ również $\{y_i\}$ mogą przyjmować dowolne wartości rzeczywiste a nie tylko $+/-1$ jak w klasyfikatorze SVM.

Przyjmuje się, że aproksymowana będzie funkcja liniowa:

$y = f_{\boldsymbol{a}}(\boldsymbol{x}) = \boldsymbol{a}^T \cdot \phi(\boldsymbol{x}) + b$, gdzie $\phi(\boldsymbol{x})$ jest pewną transformacją

Zadaniem uczenia jest taki dobór wektora wag $\boldsymbol{a}$, aby zminimalizować wartość funkcji błędu o tolerancji ε :

$$\boldsymbol{a} = \arg\min E(\boldsymbol{a})$$

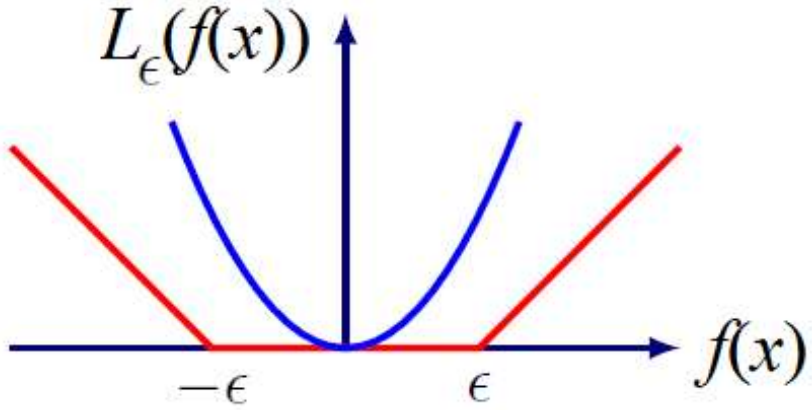
$$E(\boldsymbol{a}) = \frac{1}{2}\sum_{i=1}^{N} L_\varepsilon\left(y_i, f_{\boldsymbol{a}}(x_i)\right)$$

$$L_\varepsilon\left(y_i, f_{\boldsymbol{a}}(\boldsymbol{x}_i)\right) = max(0, |y_i - f_{\boldsymbol{a}}(\boldsymbol{x}_i)| - \varepsilon)$$

ε –tolerancja

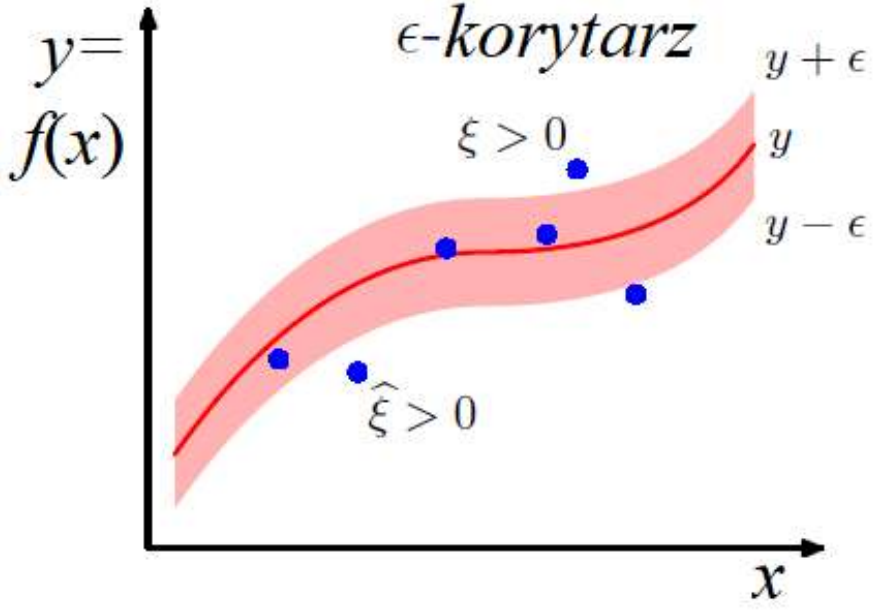
Funkcja błędu z tolerancją ε :

$$L_\varepsilon\left(y_i, f_{\boldsymbol{a}}(\boldsymbol{x}_i)\right) = max(0, |y_i - f_{\boldsymbol{a}}(\boldsymbol{x}_i)| - \varepsilon)$$



Celem SVR jest umieszczenie „prawie wszystkich” próbek uczących w korytarzu o tolerancji wokół funkcji krzywej $y = f_{\boldsymbol{a}}(\boldsymbol{x}_i)$.

Próbki położone poza korytarzem potraktujemy jako „szum” i uwzględnimy w składowej „kary” w optymalizowanej funkcji celu.



Zadanie uczenia SVR

Funkcja celu (zadanie) uczenia w dziedzinie pierwotnej:

$$\hat{\boldsymbol{a}} = \arg\min_{a} \frac{1}{2}\boldsymbol{a}^T\boldsymbol{a} + C\left[\sum_{i=1}^{N}(\xi_i + \xi_i')\right]$$

przy liniowych ograniczeniach, dla $i = 1,2,\cdots,N$:

- $(y_i - (\boldsymbol{a}^T \cdot \phi(\boldsymbol{x}) + b) \leq \varepsilon + \xi_i)$
- $\left((\boldsymbol{a}^T \cdot \phi(\boldsymbol{x}) + b) - y_i \leq \varepsilon + \xi_i'\right)$
- $\xi_i \geq 0,\ \xi_i' \geq 0.$

Rozwiązanie przebiega podobnie jak dla klasyfikatora SVM. Definiuje się funkcję Lagrange'a z nieujemnymi mnożnikami $(\alpha_i\ ,\ \alpha'_i)$ dla każdej próbki. Następnie przechodzi się do postaci problemu dualnego minimalizując Lagrange'ian względem wag $\boldsymbol{a}$ i zmiennych dopełniających $(\xi_i\ ,\ \xi_i')$. Rozwiązanie tego problemu stanowią wektory nośne $(j = 1,2,\cdots,N_{SV})$ i ich mnożniki $(\alpha_i\ ,\ \alpha'_i)$.

Rozwiązanie SVR

Parametry funkcji SVR obliczane są na podstawie wyznaczonych uprzednio wektorów nośnych i ich mnożników Lagrange'a:

$$\boldsymbol{a} = \sum_{j=1}^{N_{SV}} (\alpha_j - \alpha_j') \phi(\boldsymbol{x}_j)$$

Funkcja SVR jest wtedy postaci:

$$y = f_{\boldsymbol{a}}(\boldsymbol{x}) = \sum_{j=1}^{N_{SV}} (\alpha_j - \alpha_j') K(\boldsymbol{x}, \boldsymbol{x}_j) + b$$

gdzie funkcja „jądra" to: $K(\boldsymbol{x}, \boldsymbol{x}_j) = \phi^T(\boldsymbol{x}) \phi(\boldsymbol{x}_j)$

Pytania

1. Na czym polega zadanie aproksymacji parametrycznej (regresji)?
2. Jak wyznaczamy parametry aproksymacji funkcji?
3. Przedstawić pojęcie macierzy pseudo-odwrotnej Moore-Penrose.
4. Omówić metody nieliniowej regresji (nieliniowa MNK).
5. Omówić model pamięciowy w uczeniu się aproksymacji funkcji.
6. Przedstawić model SVR.