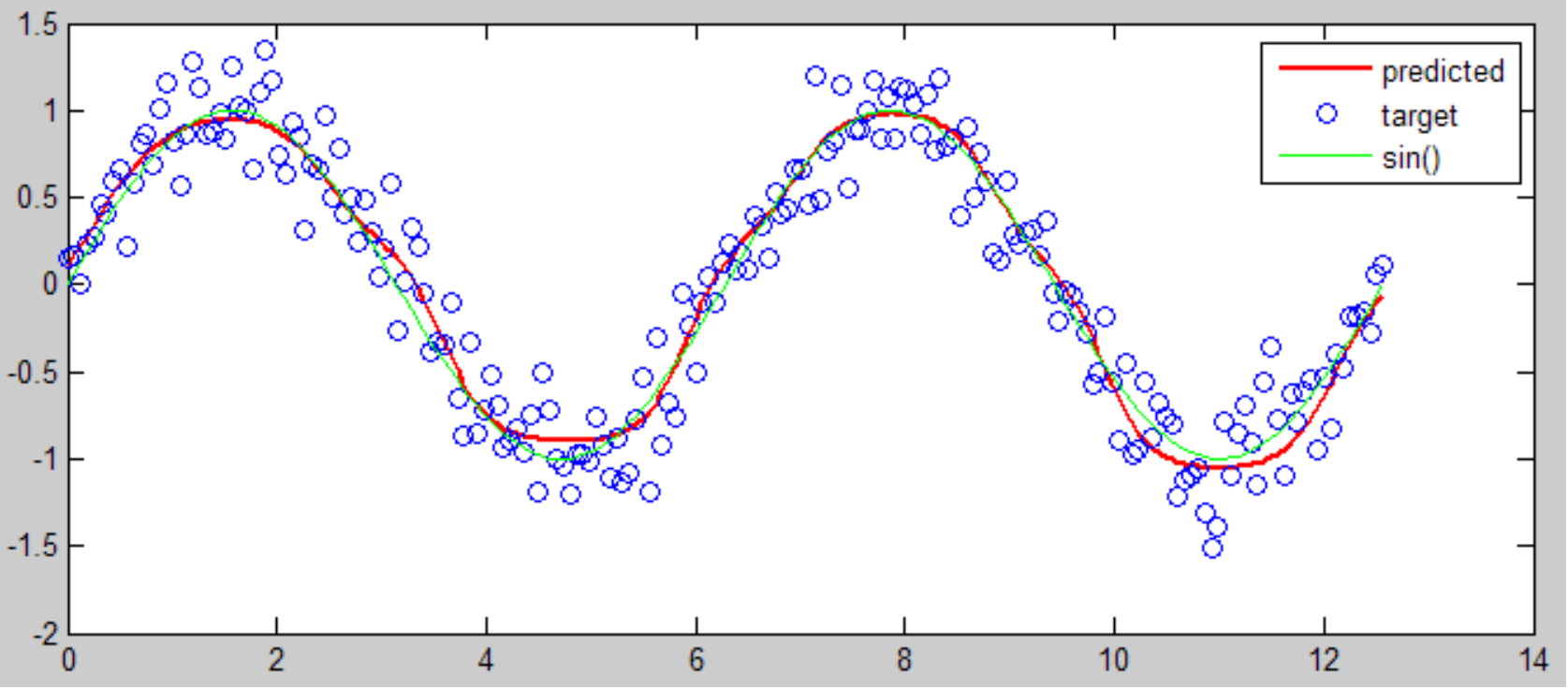
Wprowadzenie do Aproksymacji

Czym jest Aproksymacja?

- Aproksymacja funkcji $f(x)$ oznacza jej przybliżenie za pomocą innej, „prostszej” funkcji $\hat{f}(x)$.



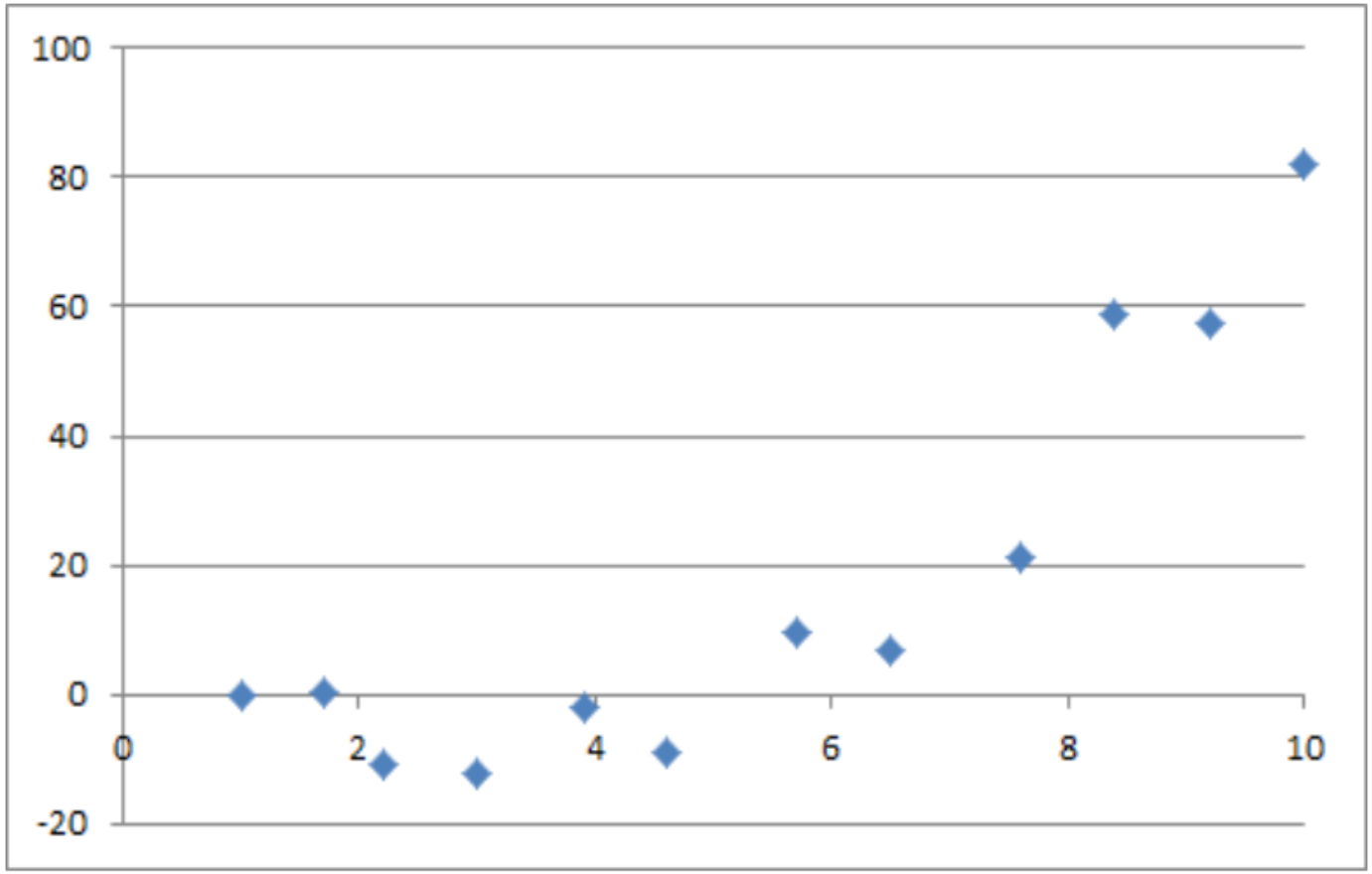
Źródło: https://stackoverflow.com/questions/1565115/approximating-function-with-neural-network

Czym jest Aproksymacja?

- Aproksymacja funkcji $f(x)$ oznacza jej przybliżenie za pomocą innej, „prostszej” funkcji $\hat{f}(x)$.
- Dlaczego?
  - **uproszczenie zagadnienia** - funkcja aproksymowana $f(x)$ jest bardzo skomplikowana;
  - **rozszerzenie zastosowań** - znamy tylko skończony zbiór wartości $f(x)$.
- Co możemy aproksymować:
  - $f(x)\colon \mathbb{R} \to \mathbb{R}$
  - $f(x)\colon \mathbb{R}^n \to \mathbb{R}$
  - $f(x)\colon \mathbb{R}^n \to \mathbb{R}^m$

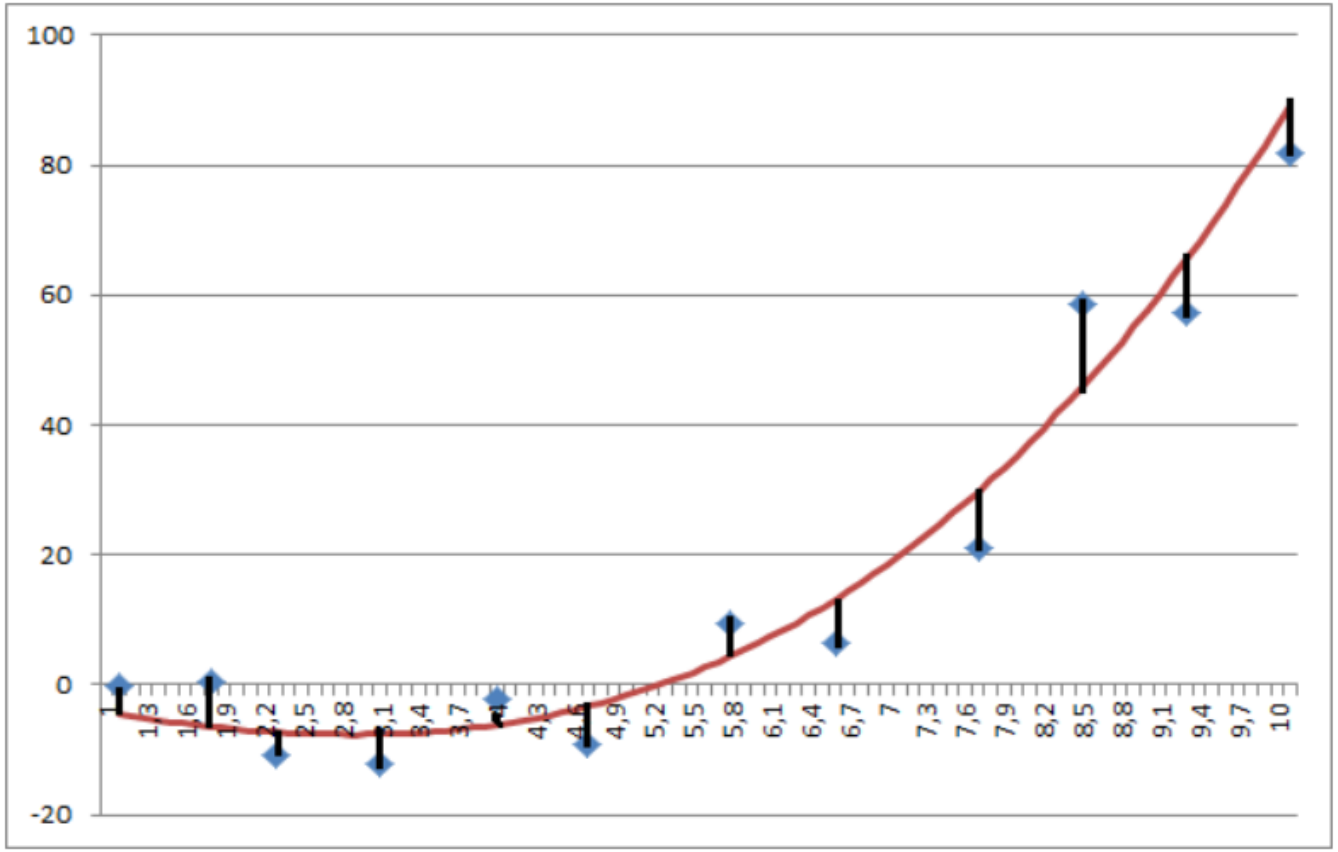
Czym jest Aproksymacja?

- Aproksymacja dyskretna:



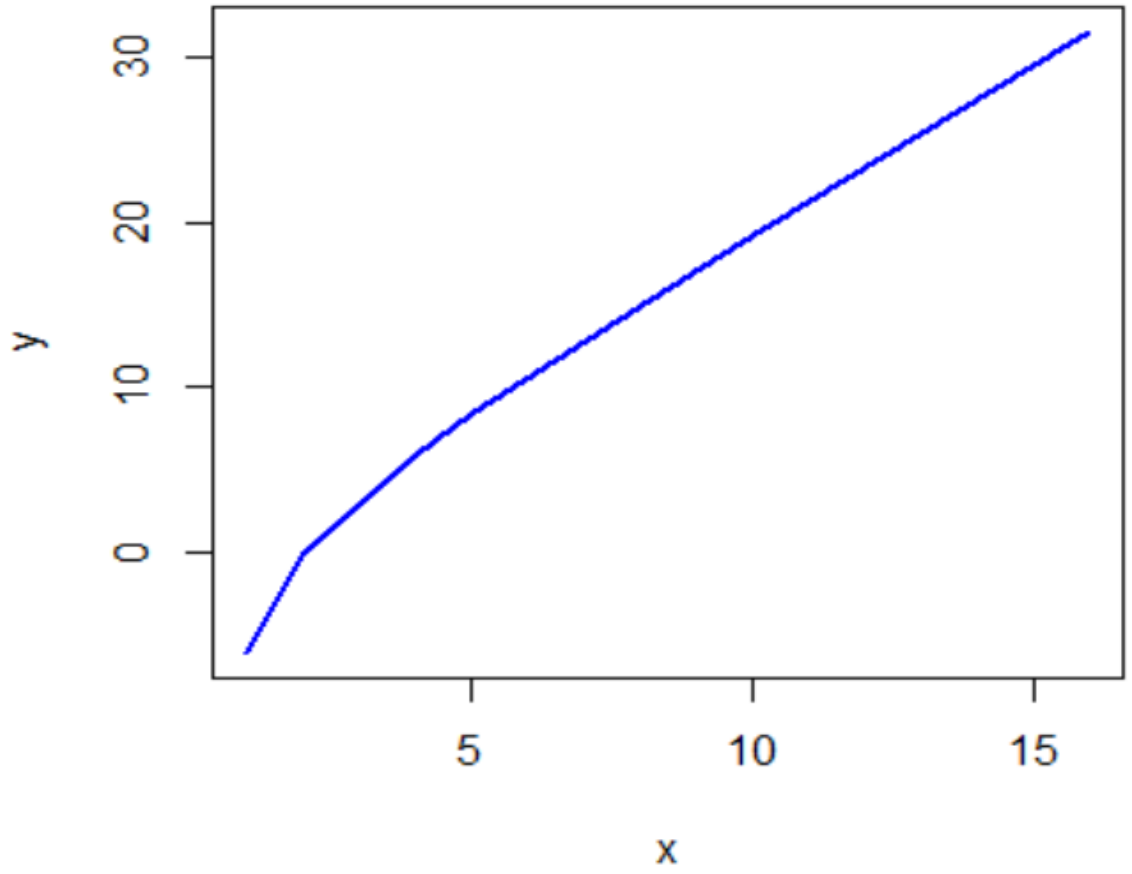
Czym jest Aproksymacja?

- Aproksymacja dyskretna:



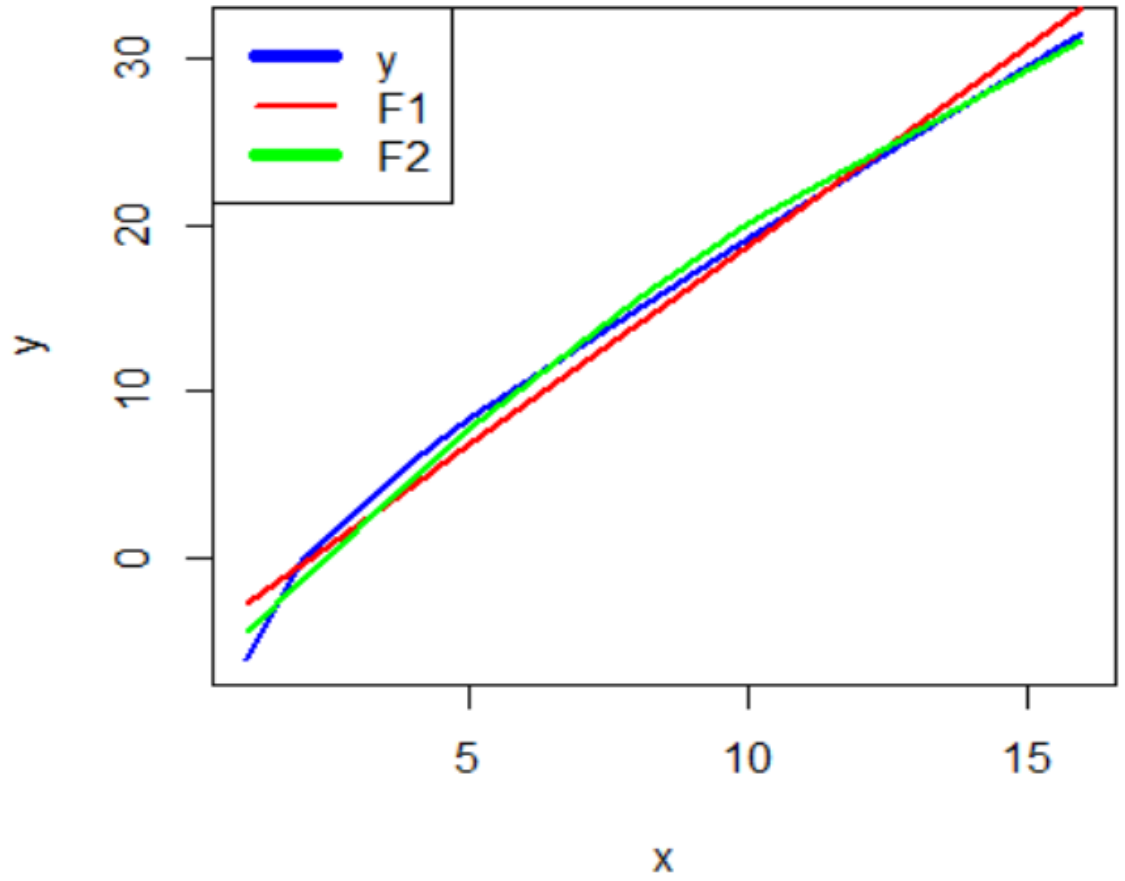
Czym jest Aproksymacja?

- Aproksymacja ciągła:



Czym jest Aproksymacja?

- Aproksymacja ciągła:



Czym jest Aproksymacja?

- Zakładamy, że funkcja aproksymująca należy do ustalonej klasy funkcji np. uogólnionych wielomianów:

$$\hat{f}(x) = a_0\phi_0(x) + a_1\phi_1(x) + \cdots + a_m\phi_m(x)$$

gdzie ϕ_0 to zadana z góry **funkcja bazowa**, która może mieć postać:

- Jednomianu

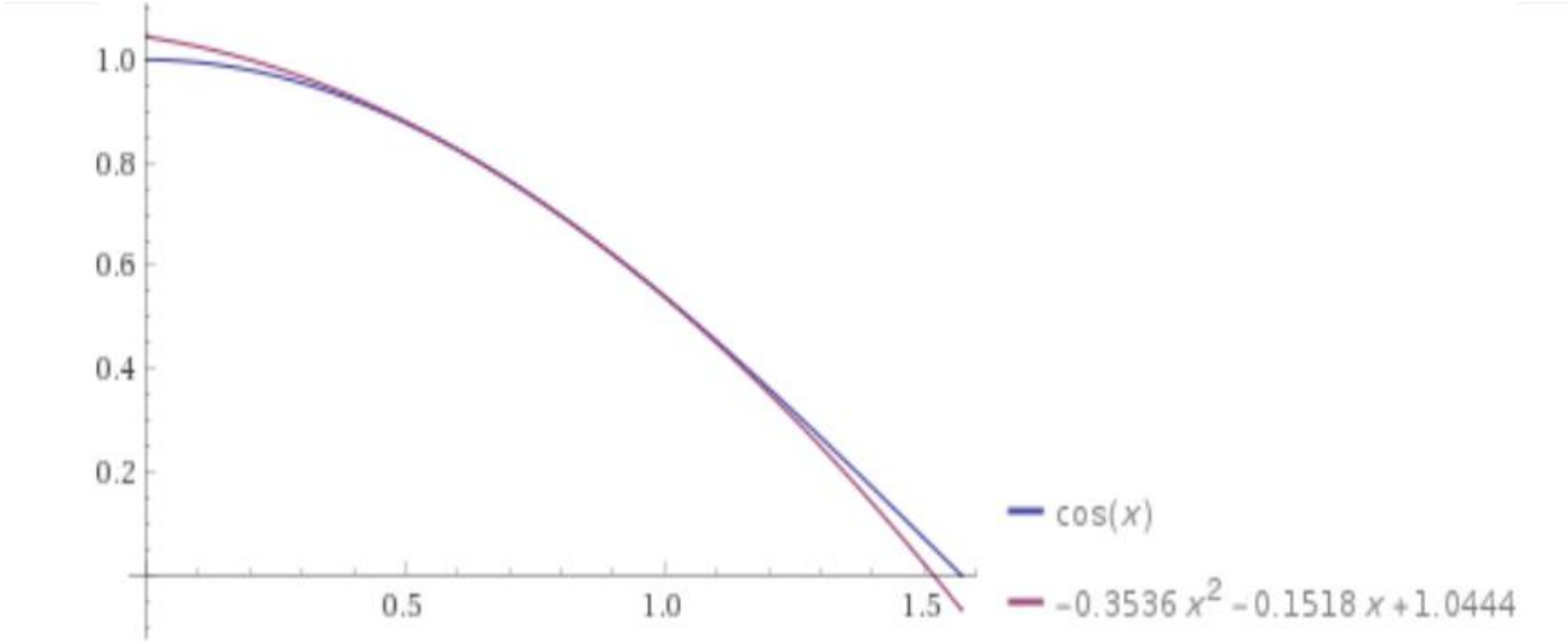
$$1, x, x^2, \ldots, x^m$$

- Wielomianu ortogonalnego
- Funkcji trygonometrycznej

$$1, \sin x\,, \cos x\,, \sin 2x\,, \cos 2x\,, \ldots, \sin mx\,, \cos mx$$

Czym jest Aproksymacja?

- Aproksymacja funkcji inherentnie wiąże się z pojawieniem się błędów, nazywanych **błędami aproksymacji**.



Czym jest Aproksymacja?

- Konieczne jest więc wprowadzenie miary, która pozwoli na mierzenie takiego błędu. Naturalnie chcemy żeby aproksymata $\hat{f}(x)$ minimalizowała taki błąd.
- W zależności od przyjętego błędu możemy mówić o **aproksymacji jednostajnej**:

$$\left\|f(x)-\hat{f}(x)\right\| = \sup_{x\in[a,b]}\left|f(x)-\hat{f}(x)\right|$$

- Lub **średniokwadratowej**:

$$\left\|f(x)-\hat{f}(x)\right\| = \int_a^b w(x)\left|f(x)-\hat{f}(x)\right|^2 dx \text{ (przypadek ciągły)}$$

$$\left\|f(x)-\hat{f}(x)\right\| = \sum_{i=1}^{n} w(x_i)\left|f(x_i)-\hat{f}(x_i)\right|^2 \text{ (przypadek dyskretny)}$$

Czym jest Aproksymacja?

- Możliwe są też inne metody obliczania błędu (np. **Entropia**)
- Wielkość błędów aproksymacji wpływa na wybór metody I funkcji aproksymujacej.

Czym jest Aproksymacja?

Twierdzenie Stone’a-Weierstrassa 1:

Jeżeli funkcja $f(x)$ jest określona i ciągła na skończonym przedziale $[a, b]$, to dla każdego $\epsilon > 0$ istnieje takie n i wielomian $\mathrm{W}_n(x)$ stopnia n, dla którego zachodzi nierówność:

$$|f(x) - W_n(x)| < \epsilon$$

na przedziale $[a, b]$.

Czym jest Aproksymacja?

Twierdzenie Stone’a-Weierstrassa 2:

Jeżeli funkcja $f(x)$ jest funkcją ciągłą i okresową w $\mathbb{R}$, o okresie 2π, to dla każdego $\epsilon > 0$ istnieje taki wielomian trygonometryczny

$$S_n(x) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \sin kx + b_k \cos kx)$$

dla którego zachodzi nierówność:

$$|f(x) - S_n(x)| < \epsilon$$

Aproksymacja a uczenie maszynowe

- Uczenie maszynowe (uczenie nadzorowane) może być traktowane jako specjalny, nietrywialny przypadek aproksymacji.
- Przede wszystkim, minimalizacja błędu musi odbywać się w sposób niejawny – nie znamy rzeczywistej postaci funkcji $f(x)$, tylko jej realizację $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \sim \mathcal{D}$
- Aby rozwiązać takie zadanie, musimy posłużyć się **funkcją kosztu** $J(\alpha)$.

Aproksymacja a uczenie maszynowe

- Funkcje kosztu $J(\alpha)$ definiujemy zazwyczaj jako przeciętną wartość **funkcji straty L**:

$$J(\alpha) = \mathbb{E}_{(x,y)\sim\mathcal{D}} L(\hat{f}(x;\alpha), y)$$

gdzie $\mathcal{D}$ to rozkład zmiennych x i y.

- Naszym celem jest **minimalizacja** funkcji $J(\alpha)$, czyli znalezienie takich parametrów α dla zadanej z góry rodziny funkcji $\mathcal{F}$ dla których błąd będzie możliwie najniższy.

Aproksymacja a uczenie maszynowe

- Funkcje kosztu $J(\alpha)$ definiujemy zazwyczaj jako przeciętną wartość **funkcji straty L**:

$$J(\alpha) = \mathbb{E}_{(x,y)\sim\hat{\mathcal{D}}}\, L(\hat{f}(x;\alpha), y)$$

gdzie $\hat{\mathcal{D}}$ to empiryczny rozkład zmiennych x i y.

- Naszym celem jest **minimalizacja** funkcji $J(\alpha)$, czyli znalezienie takich parametrów α dla zadanej z góry rodziny funkcji $\mathcal{F}$ dla których błąd będzie możliwie najniższy.

Aproksymacja a uczenie maszynowe

Pierwotna definicja uczenia statystycznego (Vapnik, 1999):

Dla zadanej klasy funkcji $\mathcal{F} = \{\alpha \in \Lambda: \hat{f}(x, \alpha)\}$, procesu generującego dane $\mathcal{D} = (X, Y)$ oraz funkcji straty $L(Y, \hat{Y})$ należy rozwiązać problem:

$$\hat{\alpha} = \underset{\alpha \in \Lambda}{\operatorname{argmin}} \left(\mathbb{E} \left(L\left(\hat{f}(x; \alpha), y \right) \right) \right)$$

Na podstawie próby $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Aproksymacja a uczenie maszynowe

- Minimalizacja funkcji $J(\alpha)$ nie jest trywialnym zadaniem; musimy rozpatrzyć dwa potencjalne źródła błędu:
  - Błąd aproksymacji.
  - Błąd estymacji.
- Co więcej, zazwyczaj budujemy modele w celach prognostycznych.
- Musimy więc ocenić jaka jest skuteczność / poprawność aproksymacji na danych, które nie były wykorzystywane do budowy modelu.
- Błędy liczone na zbiorach treningowym i testowym mogą się znacznie różnić.
- Jak więc wybrać najlepszy model?

Twierdzenie Vapnika – problem klasyfikacji

- Zadana klasa funkcji dopuszczalnych $\mathcal{F}$.
- Dla $\mathcal{F}$ można wyznaczyć **wymiar Vapnika-Chervonenkisa** $h(\mathcal{F})$ mierzący jej zdolność dopasowania się do danych.
- Dysponujemy n-elementową próbą estymacyjną.
- Wybieramy funkcję $f \in \mathcal{F}$ minimalizującą błąd na danych estymacyjnych R_e.
- Chcemy oszacować błąd prognozy R_p:

Twierdzenie Vapnika – problem klasyfikacji

Twierdzenie (Vapnik, 1995):

Dla dowolnego rozkładu (X, Y) i klasy funkcji $\mathcal{F}$ z prawdopodobieństwem $1 - q$ zachodzi zależność:

$$R_p \leq R_e + \underbrace{\sqrt{\frac{h(\mathcal{F})\left(1 + \ln\left(2n/h(\mathcal{F})\right)\right) - \ln(q/4)}{n}}}_{\epsilon}$$

Wymiar Vapnika-Chervonenkisa

- Dla zadanego zbioru przykładów $C = \{c_1, c_2, c_3, \dots, c_n\} \in X$ możemy zdefiniować klasyfikator f jako funkcję która dla każdego podzbioru $C' \subseteq C$ (**każdej klasy**) pozwala przypisać mu odpowiednią etykietę:

$$f(c) = \begin{cases} 1 & c \in C' \\ 0 & c \notin C' \end{cases}$$

Wymiar Vapnika-Chervonenkisa

- Dla zadanego zbioru przykładów $C = \{c_1, c_2, c_3, \dots, c_n\} \in X$ możemy zdefiniować klasyfikator f jako funkcję która dla każdego podzbioru $C' \subseteq C$ (**każdej klasy**) pozwala przypisać mu odpowiednią etykietę:

$$f(c) = \begin{cases} 1 & c \in C' \\ 0 & c \notin C' \end{cases}$$

- Mówimy, że klasa funkcji $\mathcal{F}$ **rozdziela** (*shatters*) zbiór C jeżeli istnieje taka funkcja $f \in \mathcal{F}$, że dla przyporządkowania $\mathcal{F}_C = \{(f(c_1), f(c_2), \dots, f(c_n)) | f \in \mathcal{F}\}$ moc zbioru $|\mathcal{F}_C| = 2^{|C|}$.

Wymiar Vapnika-Chervonenkisa

- Dla zadanego zbioru przykładów $C = \{c_1, c_2, c_3, \dots, c_n\} \in X$ możemy zdefiniować klasyfikator f jako funkcję która dla każdego podzbioru $C' \subseteq C$ (**każdej klasy**) pozwala przypisać mu odpowiednią etykietę:

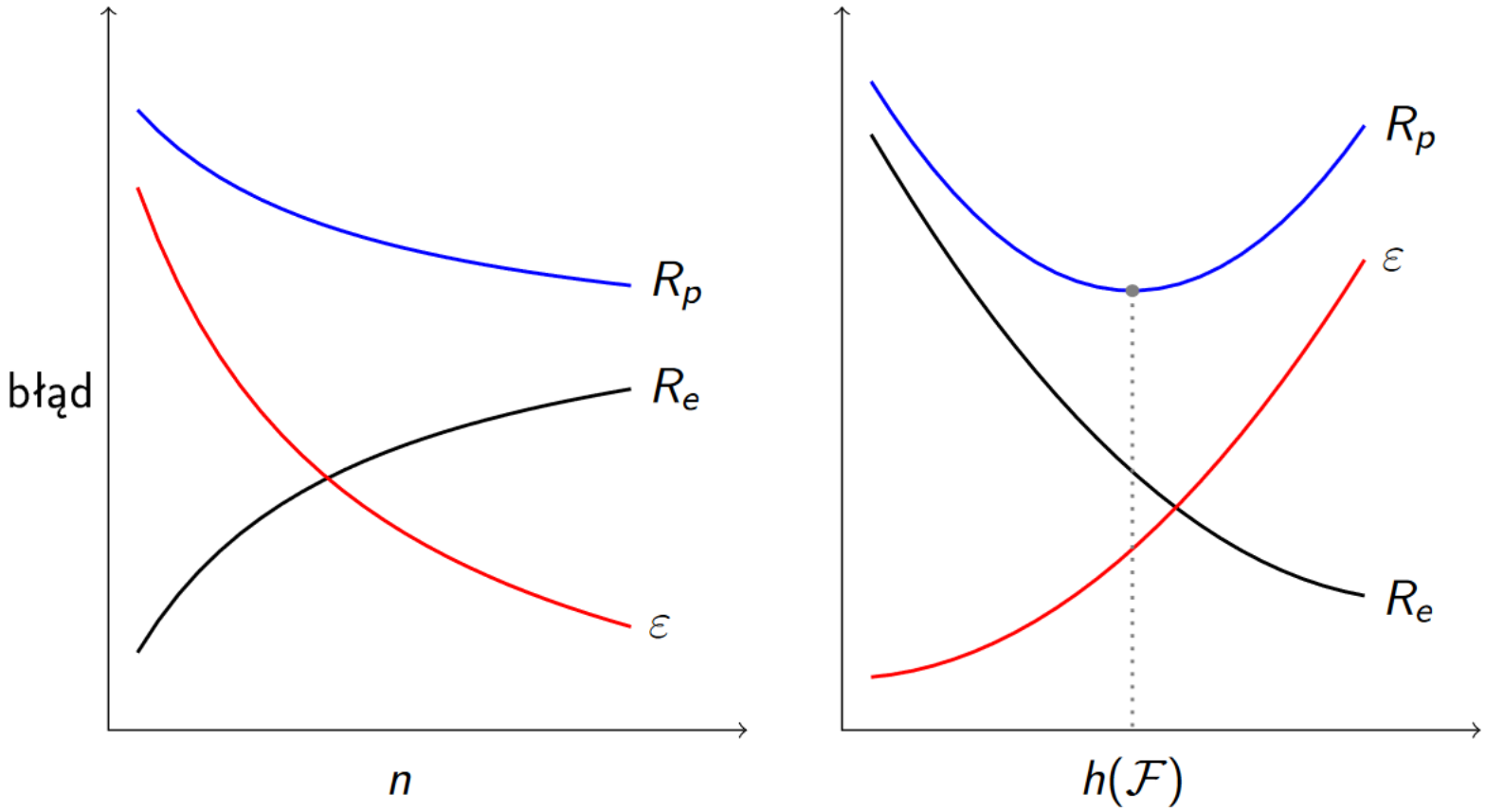
$$f(c) = \begin{cases} 1 & c \in C' \\ 0 & c \notin C' \end{cases}$$

- Mówimy, że klasa funkcji $\mathcal{F}$ **rozdziela** (*shatters*) zbiór C jeżeli istnieje taka funkcja $f \in \mathcal{F}$, że dla przyporządkowania $\mathcal{F}_C = \{(f(c_1), f(c_2), \dots, f(c_n)) | f \in \mathcal{F}\}$ moc zbioru $|\mathcal{F}_C| = 2^{|C|}$.
- Intuicyjnie oznacza to, że funkcja f poprawnie przyporządkowuje etykiety na wszystkie 2^n sposobów niezależnie od tego w jaki sposób są one początkowo przydzielone (jaka próbka C została wylosowana).

Wymiar Vapnika-Chervonenkisa

- Wymiar Vapnika-Chervonenkisa $h(\mathcal{F})$ klasy funkcji $\mathcal{F}$ definiujemy jako rozmiar największego zbioru $C \in X$ jaki jesteśmy w stanie rozdzielić za pomocą $\mathcal{F}$.
- Gdy możemy rozdzielić dowolny zbiór (nasz klasyfikator jest idealny) wtedy $h(\mathcal{F}) = \infty$.

Twierdzenie Vapnika – problem klasyfikacji



Twierdzenie Vapnika – problem klasyfikacji

- **Obciążenie** – błąd aproksymacji wynikający z uproszczenia rzeczywistej zależności. Im większe uproszczenie, tym większe obciążenie.

$$Bias\left(\hat{f}(x)\right) = \mathbb{E}\left(\hat{f}(x)\right) - f(x)$$

- Obciążenie i złożoność funkcji aproksymujących są odwrotnie proporcjonalne.

Twierdzenie Vapnika – problem klasyfikacji

- **Wariancja modelu** – zakres zmian wartości funkcji aproksymującej $\hat{f}$ dla różnych zbiorów treningowych. Duża wariancja oznacza, że nawet małe zmiany w danych treningowych mogą powodować duże zmiany wartości aproksymacji $\hat{f}$.

$$\mathrm{Var}\left(\hat{f}(x)\right) = \mathbb{E}\left(\hat{f}(x)\right)^2 - \left[\mathbb{E}\left(\hat{f}(x)\right)\right]^2$$

- Wariancja i złożoność funkcji aproksymujących są wprost proporcjonalne.

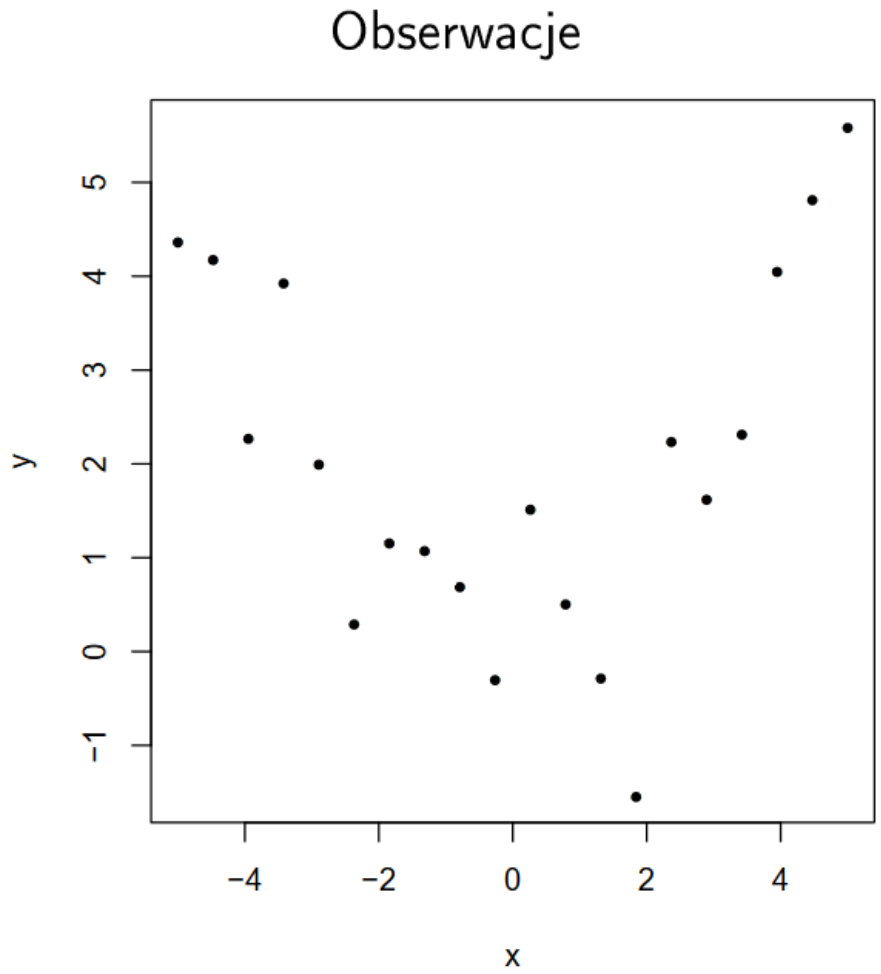
Twierdzenie Vapnika – problem klasyfikacji

- **Błąd aproksymacji** jest równy:

$$\mathbb{E}\left(\hat{f}(x) - f(x)\right)^2 = \left[Bias\left(\hat{f}(x)\right)\right]^2 + \mathrm{Var}\left(\hat{f}(x)\right) + \sigma^2$$

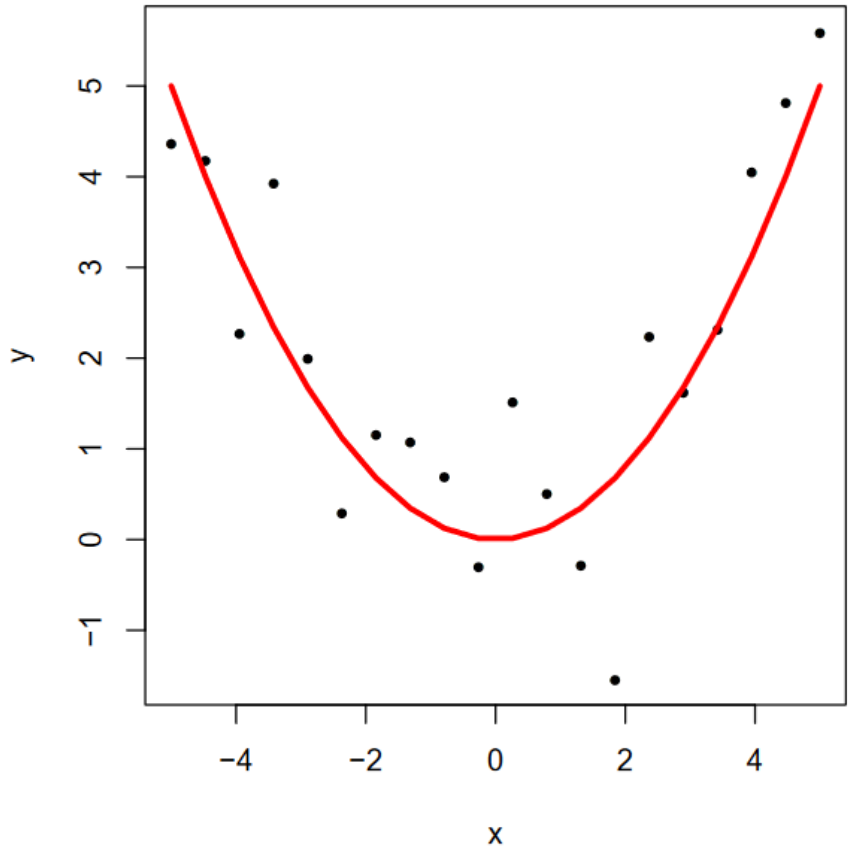
gdzie σ^2 to wariancja składnika losowego.

Twierdzenie Vapnika – problem klasyfikacji



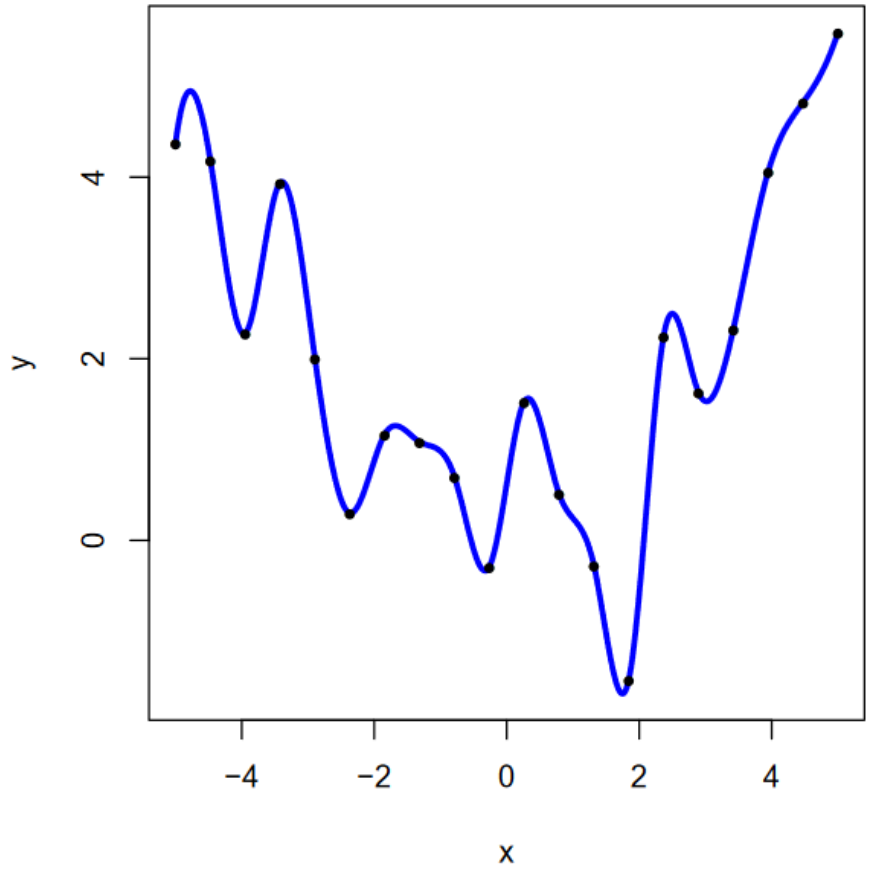
Twierdzenie Vapnika – problem klasyfikacji

proces generujący dane: $y = x^2/5 + \varepsilon$, gdzie $\varepsilon \sim N(0, 1)$



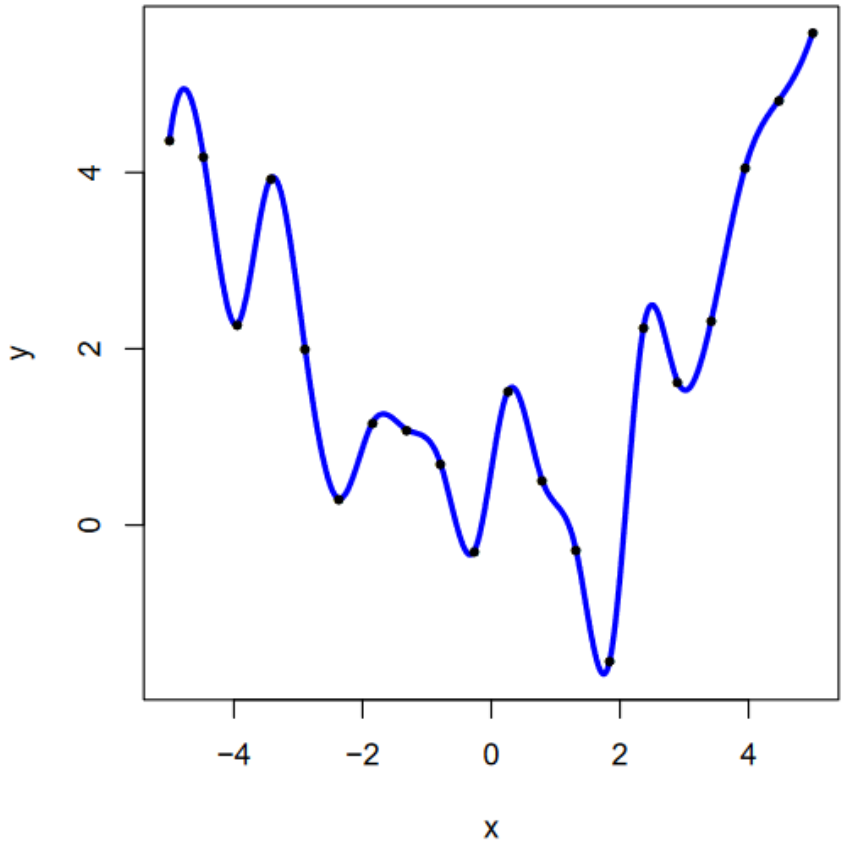
Twierdzenie Vapnika – problem klasyfikacji

Dwukrotnie różniczkowalna funkcja f: $\sum_{i=1}^{n}(f(x_i) - y_i)^2 \to \min$



Twierdzenie Vapnika – problem klasyfikacji

Dwukrotnie różniczkowalna funkcja f: $\sum_{i=1}^{n}(f(x_i) - y_i)^2 \rightarrow \min$

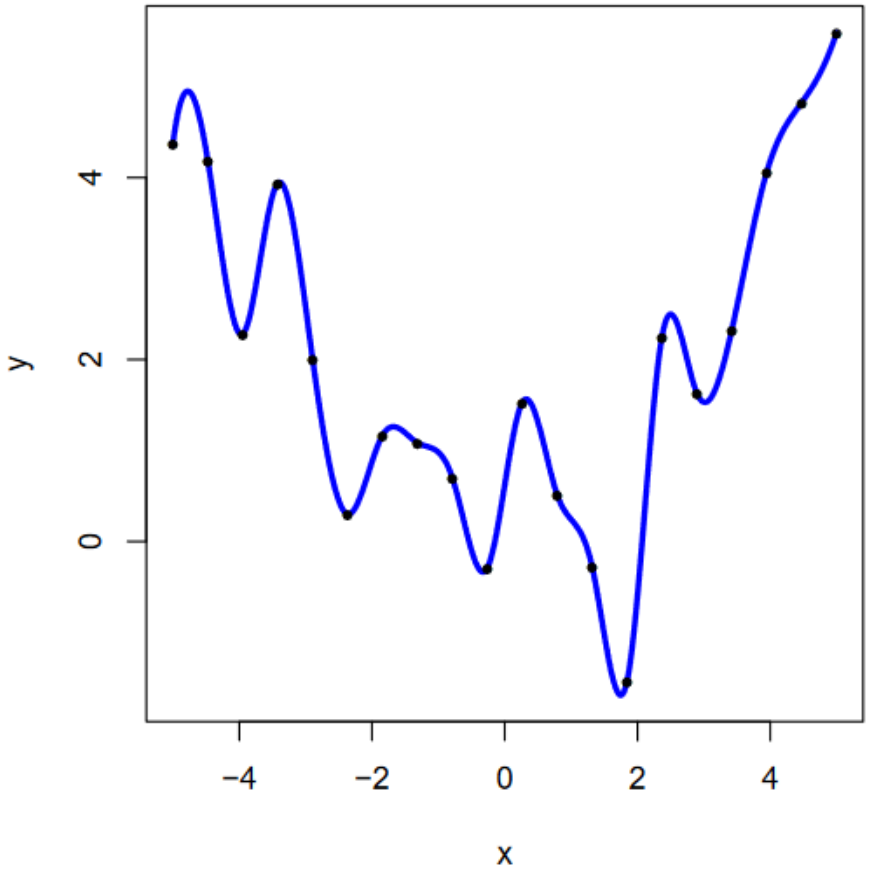


zagnieżdżona klasa funkcji:
wygładzane funkcje sklejane (Hastie et al., 2001)

Twierdzenie Vapnika – problem klasyfikacji

Dwukrotnie różniczkowalna funkcja f:

$\sum_{i=1}^{n}(f(x_i)-y_i)^2 \to \min$, p.w. $\int_D [f''(x)]^2 dx \leq \delta$

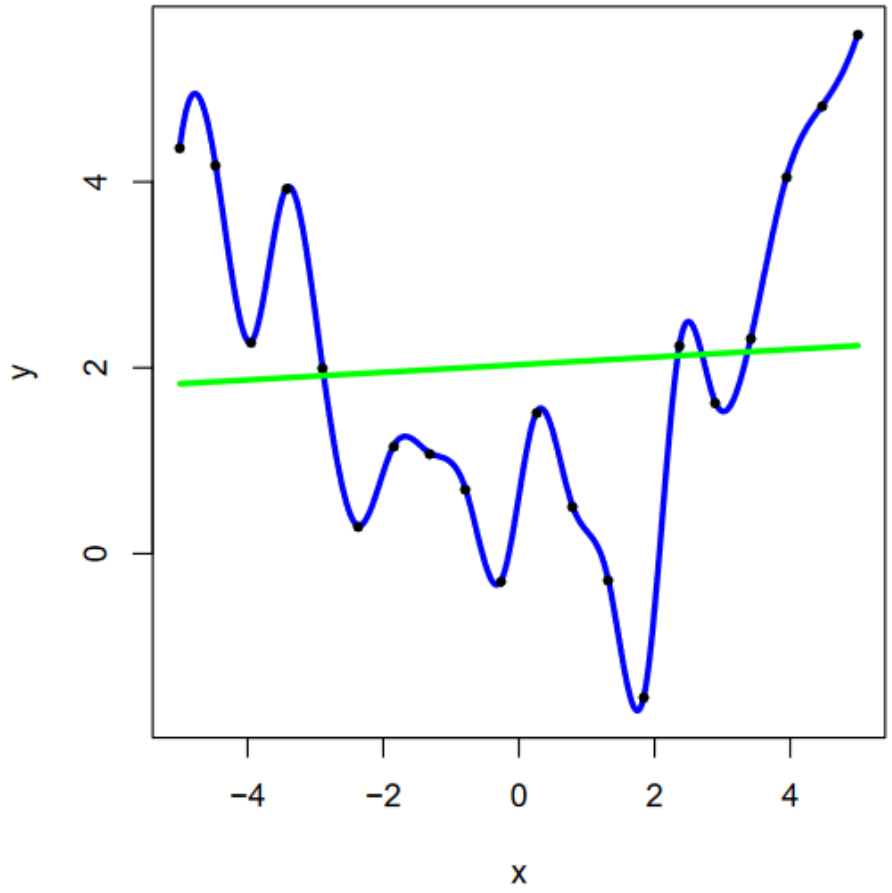


niebieski: $\delta \to +\infty$

Twierdzenie Vapnika – problem klasyfikacji

Dwukrotnie różniczkowalna funkcja f:

$\sum_{i=1}^{n}(f(x_i)-y_i)^2 \to \min$, p.w. $\int_D [f''(x)]^2 dx \leq \delta$

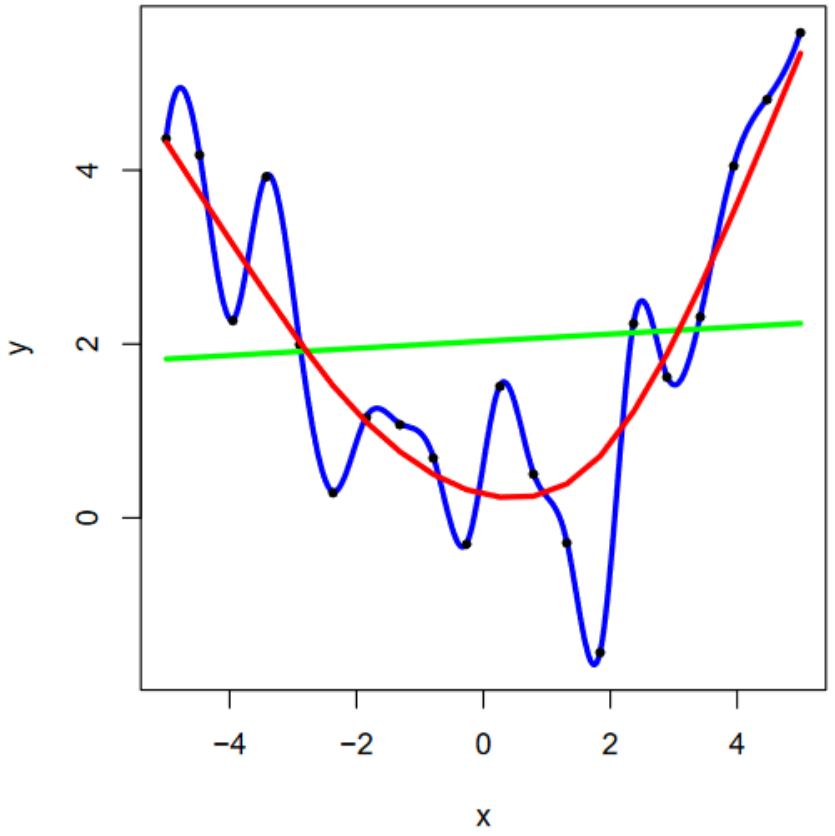


niebieski: $\delta \to +\infty$, zielony: $\delta = 0$

Twierdzenie Vapnika – problem klasyfikacji

Dwukrotnie różniczkowalna funkcja f:

$\sum_{i=1}^{n}(f(x_i)-y_i)^2 \to \min$, p.w. $\int_D [f''(x)]^2 dx \leq \delta$



niebieski: $\delta \to +\infty$, zielony: $\delta = 0$, czerwony: δ optymalne

Twierdzenie Vapnika – problem klasyfikacji

- Wybieramy rodzinę zagnieżdżonych klas funkcji:

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \dots$$

$$\Downarrow$$

$$h(\mathcal{F}_1) \leq h(\mathcal{F}_2) \leq h(\mathcal{F}_3) \leq \dots$$

- Wyznaczamy:

$$R_e(\mathcal{F}_1) \geq R_e(\mathcal{F}_2) \geq R_e(\mathcal{F}_3) \geq \dots$$

$$\varepsilon(\mathcal{F}_1) \leq \varepsilon(\mathcal{F}_2) \leq \varepsilon(\mathcal{F}_3) \leq \dots$$

- Wybieramy model oszacowany na podstawie klasy funkcji $\mathcal{F}_i$ minimalizującej oszacowanie R_p.

Twierdzenie Vapnika – problem klasyfikacji

- Ograniczenia twierdzenia Vapnika:
  - Trudność z wyznaczeniem wartości $h(\mathcal{F})$ dla złożonych klas funkcji
  - Nierówność z twierdzenia jest bardzo konserwatywna
- W praktyce stosujemy zwykle procedury alternatywne:
  - kryteria informacyjne (AIC, BIC,. . .)
  - Zbiór walidacyjny
  - Walidacja krzyżowa
  - bootstrapping