Regresja logistyczna, problemy bias/variance, diagnostyka

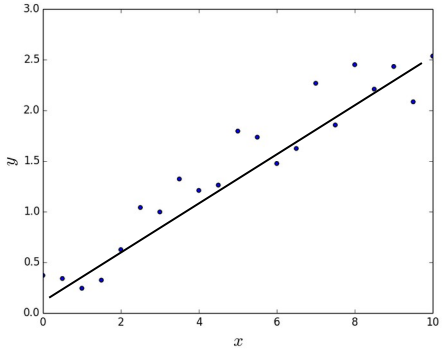
1. Regresja logistyczna

Zadania dla uczenia maszynowego

Punkt wyjścia:

- dane wejściowe (treningowe): $\mathbf{x} = (x_1, \ldots, x_n)$
- oczekiwane wyniki: $\mathbf{y}$
- liczba próbek danych treningowych: $\mathbf{m}$

REGRESJA



1. hipoteza:

$$h(x) = \theta_0 \cdot x_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n$$

2. funkcja kosztu:

$$J = \frac{1}{2m} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)} \right)^2$$

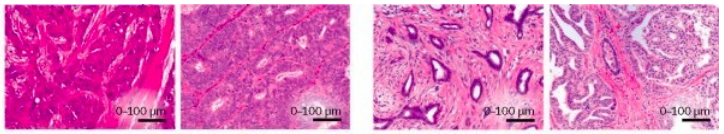
3. gradient descent:

$$\theta_j = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)} \right) \cdot x_j^{(i)}$$

KLASYFIKACJA

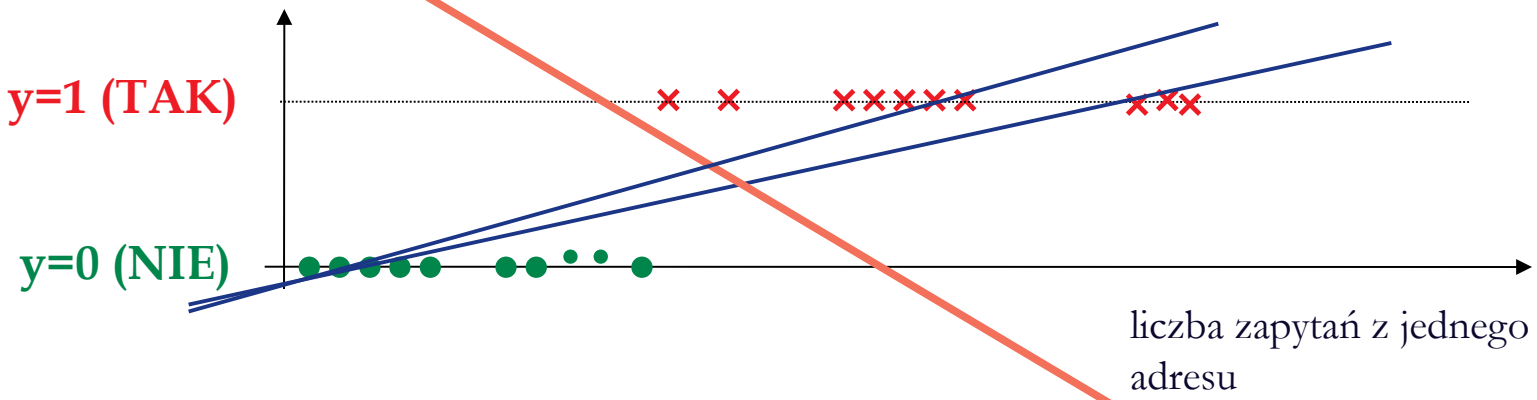
- rozpoznawanie

danych: obiektów graficznych,
dźwiękowych,
danych medycznych,
personalnych,
identyfikacja spamu,
zagrożeń sieciowych



Regresja liniowa?

Przykład – detekcja ataków DoS:



Hipoteza: $\mathbf{h(x) = \theta_0 \cdot x_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n ?}$

Regresja logistyczna

$$\mathbf{x} = (x_0, x_1, \ldots, x_n)$$

$$\theta = (\theta_0, \theta_1, \ldots, \theta_n)$$

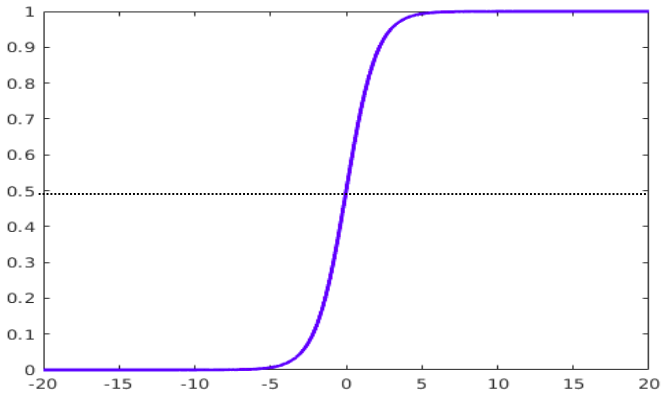
Hipoteza w regresji liniowej:

$$h(x) = \theta_0 \cdot x_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n = \theta x^T$$

Hipoteza w regresji logistycznej:

$$h(x) = g(\theta x^T) = \frac{1}{1 + \exp(-\theta x^T)}$$

funkcja sigmoidalna (logistyczna):



g(z) = 1 / (1 + exp(-z))

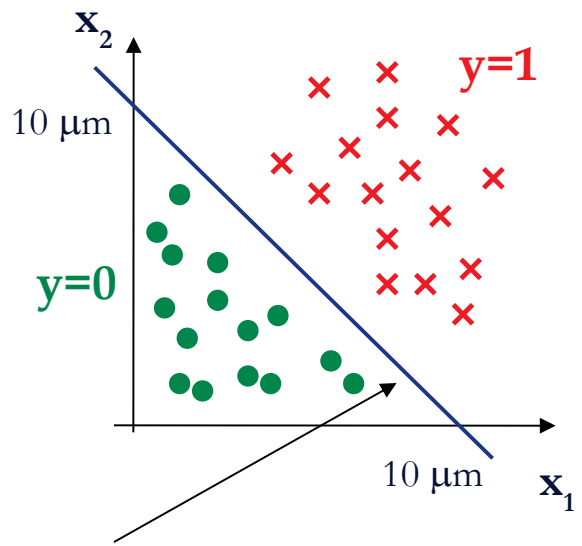
Interpretacja:

h(x) → prawdopodobieństwo, że y(x) = 1

$\theta x^T > 0$ → p>50%, że y(x) = 1

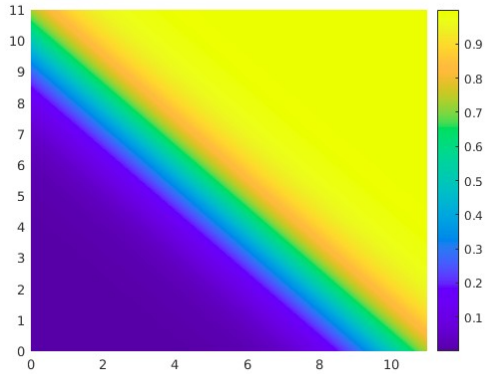
Przykłady regresji logistycznej

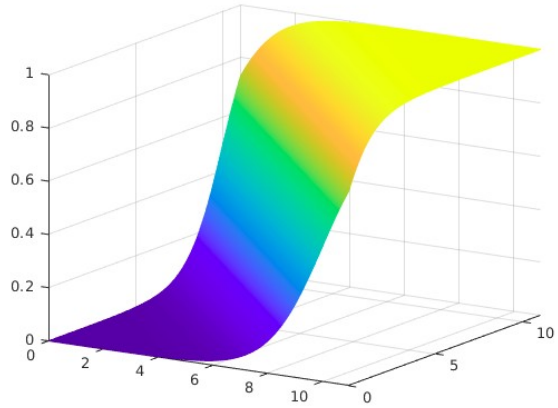
Przykład – detekcja bakterii:

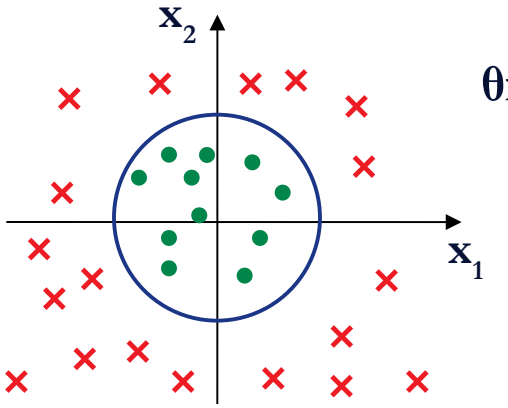


$$\theta x^T = x_1 + x_2 - 10$$

$$h(x) = \frac{1}{1 + \exp(-\theta x^T)}$$





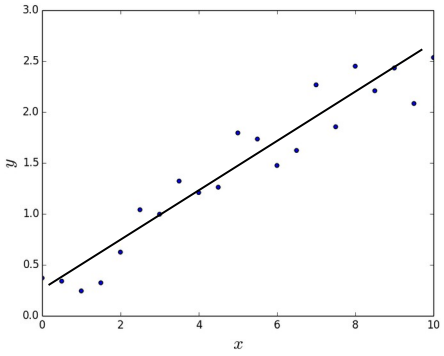


$$\theta x^T = x_1^2 + x_2^2 - r^2$$

$$h(x) = \frac{1}{1 + \exp(-x_1^2 - x_2^2 + r^2)}$$

Funkcja kosztu

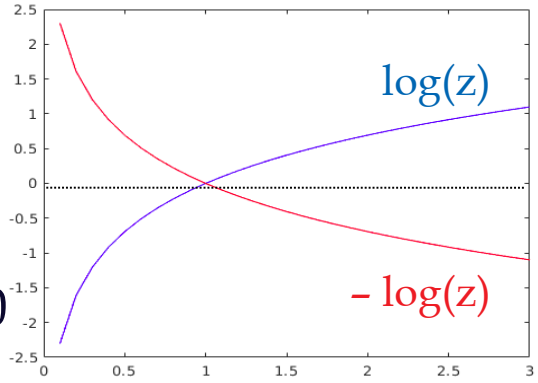
Regresja liniowa:



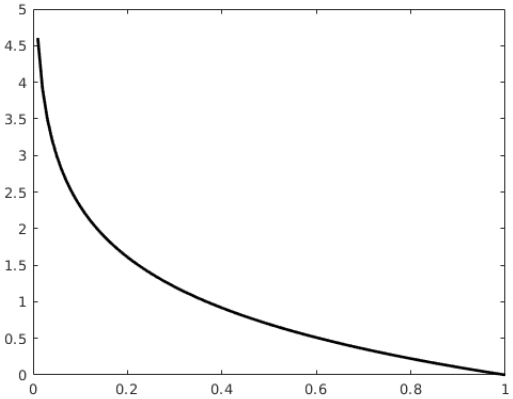
$$J = \frac{1}{2m} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)} \right)^2$$

Propozycja
dla regresji logistycznej:

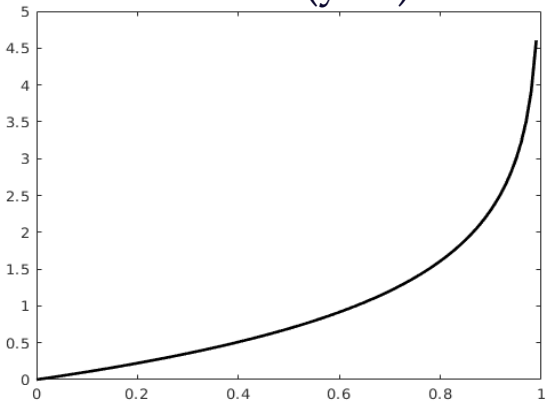
$$\textbf{Koszt} = \begin{cases} -\log(h(x)), \text{ dla } y=1 \\ -\log(1 - h(x)), \text{ dla } y=0 \end{cases}$$



Koszt (y=1)



Koszt (y=0)



$$J = -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \cdot \log(h(x^{(i)})) + (1-y^{(i)}) \cdot \log(1-h(x^{(i)}))$$

Gradient descent dla r. logistycznej

$$\textbf{Koszt} = \begin{cases} -\log(h(x)), \text{ dla } y=1 \\ -\log(1-h(x)), \text{ dla } y=0 \end{cases}$$

$$h(x) = \frac{1}{1+\exp(-\theta x^T)}$$

$$J = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \cdot \log(h(x^{(i)})) + (1-y^{(i)}) \cdot \log(1-h(x^{(i)}))$$

0 lub 1

Cel algorytmu gradient descent: policzyć $\min_{\theta_0,\theta_1,,\ldots,\theta_n} J(\theta)$

Dla każdej θ_j, j=0,1,…,n, powtarzamy:

$$\theta_j = \theta_j - \alpha \frac{d}{d\theta_j} J(\theta)$$

… (liczymy pochodną)

$$\theta_j = \theta_j - \alpha \frac{1}{m}\sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)} \right) \cdot x_j^{(i)}$$

Wygląda znajomo :-)

Jak liczyć błąd dla r. logistycznej?

$$h(x) = \frac{1}{1 + \exp(-\theta x^T)}$$

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \cdot \log(h(x^{(i)})) + (1-y^{(i)}) \cdot \log(1-h(x^{(i)}))$$

0 lub 1

Chcemy policzyć jak wiele klasyfikacji się nie udaje, a więc:

$$\textbf{err}(h(x),y) = \begin{cases} 0, \text{ gdy } h(x) \geq 0.5 \text{ i } y=1 \text{ oraz gdy } h(x)<0.5 \text{ i } y=0 \\ 1, \text{ w przeciwnym wypadku} \end{cases}$$

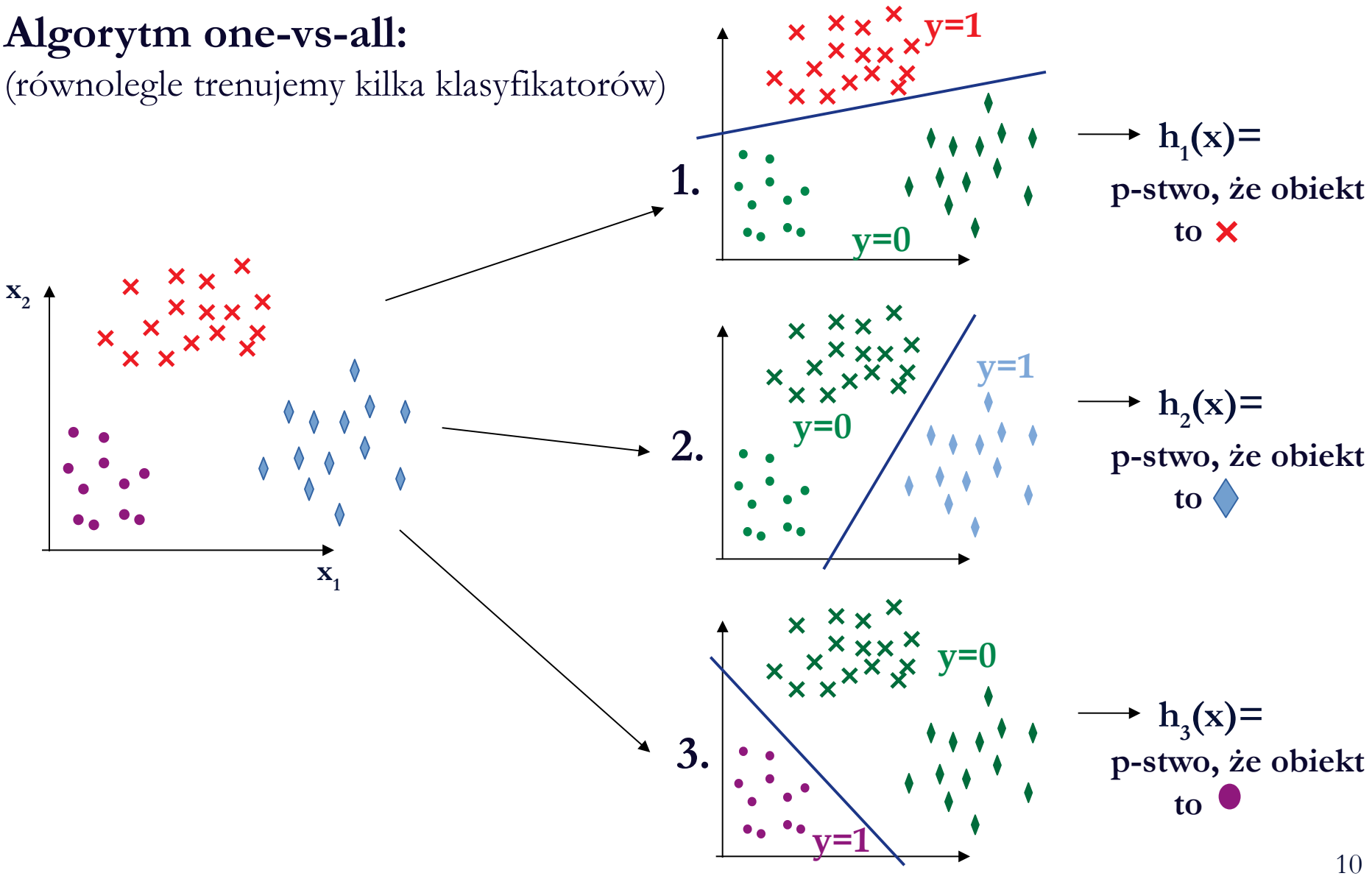
$$\textbf{Test error} = \frac{1}{m} \sum_{i=1}^{m} \text{err}\,(h(x^{(i)}), y^{(i)})$$

A jeżeli mamy więcej kategorii?

multi-class classification

Algorytm one-vs-all:

(równolegle trenujemy kilka klasyfikatorów)

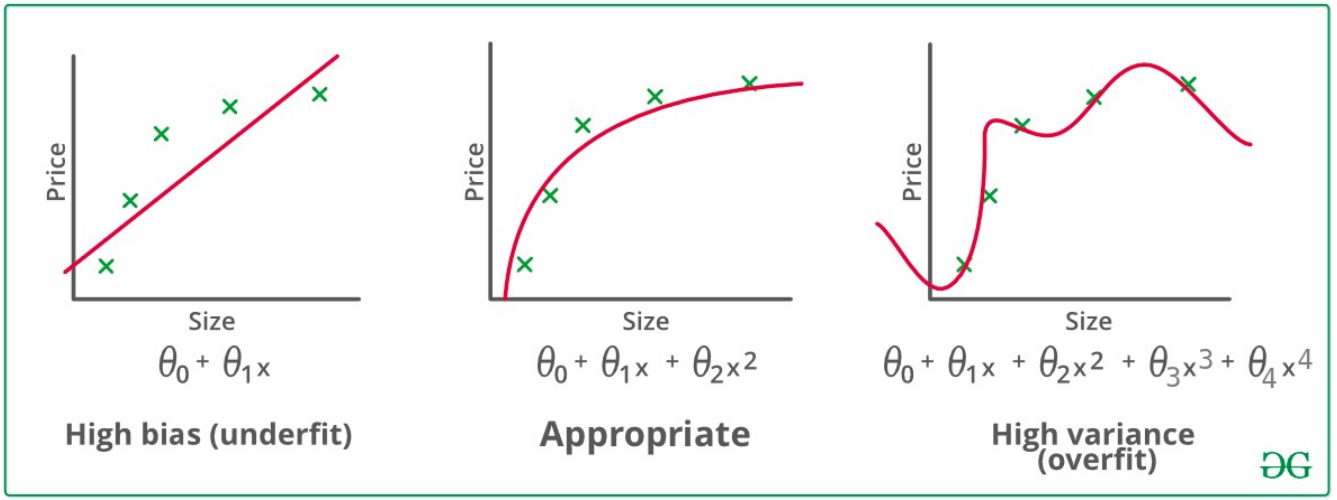


2. Problemy bias/variance

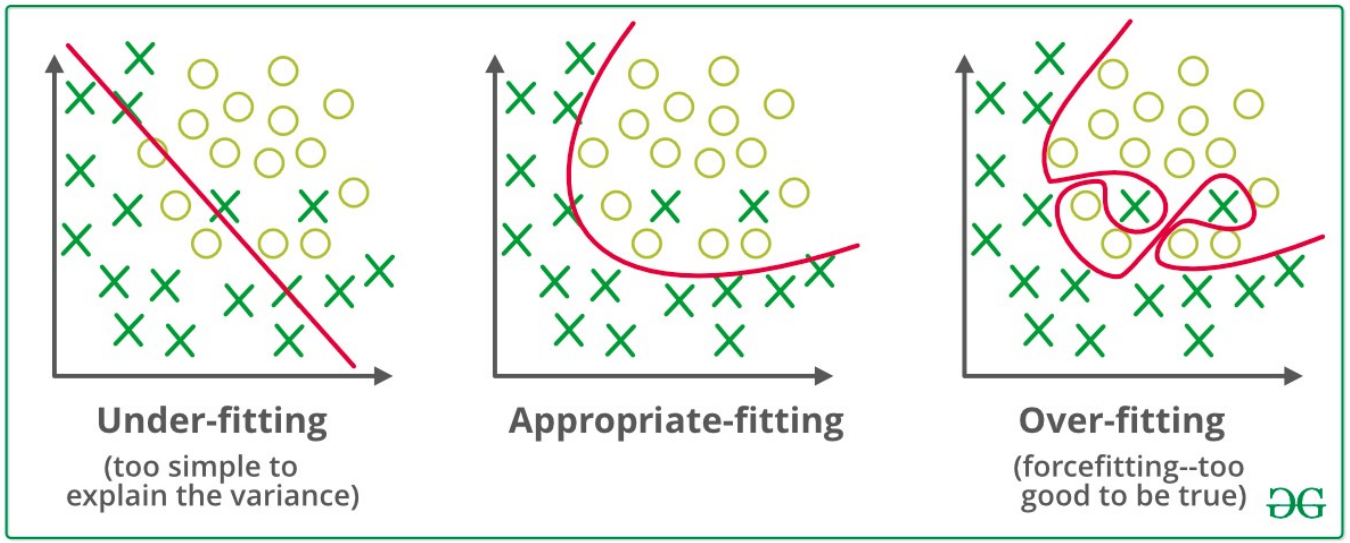
High bias/variance

underfitting / overfitting

Regresja liniowa:



Regresja logistyczna:



Underfitting – co robić?

1. **zwiększyć wymiar danych**
   (dodać nowe cechy – kolumny wśród danych wejściowych)

2. **zwiększyć wymiar danych**
   (dodać wielomiany lub inne funkcje już istniejących danych)

(model jest zbyt prosty)

Overfitting – co robić?

1. zmniejszyć wymiar danych
(usunąć wybrane cechy – kolumny wśród danych wejściowych)

2. zastosować regularyzację:
(zmniejszanie współczynników θ)

(mamy za mało próbek danych)

Regularyzacja dla r. liniowej

Hipoteza:

$$h(x) = \theta_0 \cdot x_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n$$

Funkcja kosztu:

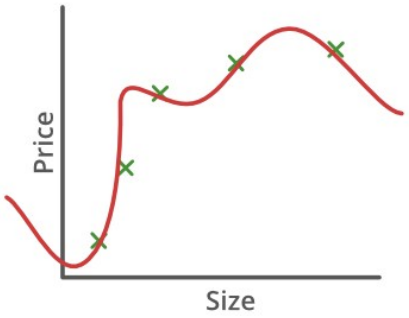
$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right]$$

współczynnik regularyzacji

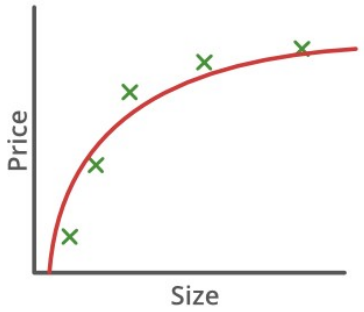
Gradient descent:

trochę mniejsze niż 1

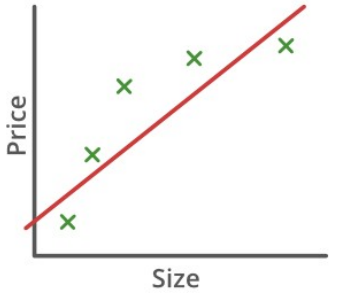
$$\theta_j = \theta_j \left(1 - \alpha \frac{\lambda}{m}\right) - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)} \right) \cdot x_j^{(i)}$$



λ zbyt mała lub 0



λ "w sam raz"



zbyt duża λ

Regularyzacja dla r. logistycznej

Hipoteza:

$$h(x) = \frac{1}{1 + \exp(-\theta x^T)}$$

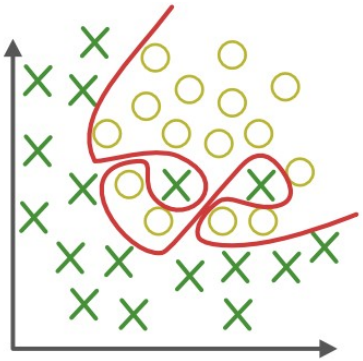
$$\theta x^T = \theta_0 \cdot x_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n$$

Funkcja kosztu:

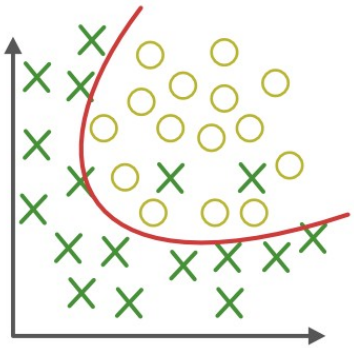
$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} \cdot \log(h(x^{(i)})) + (1-y^{(i)}) \cdot \log(1-h(x^{(i)}))] + \frac{1}{2m}\lambda\sum_{j=1}^{n}\theta_j^2$$

Gradient descent:

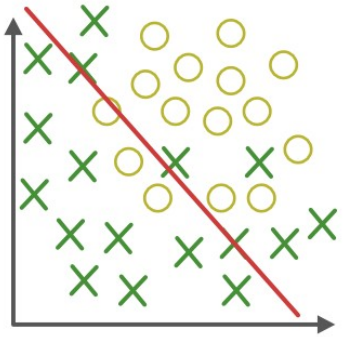
$$\theta_j = \theta_j \left(1 - \alpha\frac{\lambda}{m}\right) - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h(x^{(i)}) - y^{(i)}\right) \cdot x_j^{(i)}$$



λ zbyt mała lub 0



λ "w sam raz"



zbyt duża λ

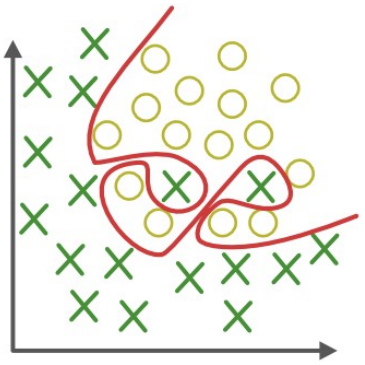
3. Diagnostyka

Mam algorytm, ale nie działa dobrze...

… błędy są zbyt duże...

Możliwe strategie:

- dodać więcej danych wejściowych (próbek)?
- zwiększyć wymiar danych (liczbę cech)?
- dodać nowe dane – wielomiany cech istniejących?
- zmniejszyć wymiar danych?
- regularyzacja – większa λ?
- regularyzacja – mniejsza λ?



Wielowymiarowe dane
nie dadzą się tak ładnie zwizualizować!

Diagnostyka – zbiory danych

Podział (losowy!) danych wejściowych:

x_1	x_2	x_3	y
6k	27	3.2	300
11.5k	39	4.3	500
8k	41	4.5	350
4k	84	7.8	650
15k	42	3.7	850
18.5k	36	4.1	800
22k	149	13.6	1800
21k	29	3.0	1000
23k	69	7.7	1400
24k	71	10.1	1450

m próbek danych

Po co nam 3 zbiory?

Uczenie modelu na zbiorze treningowym m_{train}:

- hipoteza h(x)
- funkcja kosztu: $$J_{train}(\theta) = \frac{1}{2m_{train}} \left[\sum_{i=1}^{m_{train}} \left(h(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right]$$
- gradient descent: iteracyjna optymalizacja θ

Walidacja parametrów modelu na zbiorze walidacyjnym m_{cv}:

- funkcja kosztu: $$J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} \left(h(x^{(i)}) - y^{(i)}\right)^2$$

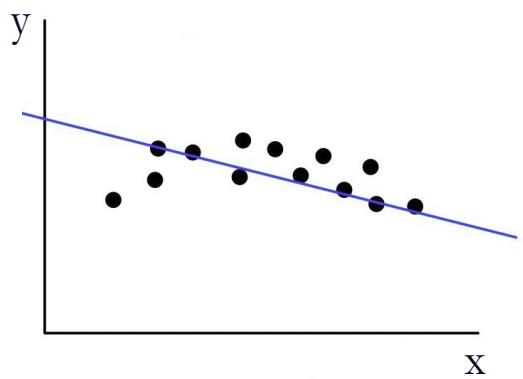
Ostateczny test modelu na niezależnym (*) zbiorze testowym m_{test}:

- funkcja kosztu: $$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} \left(h(x^{(i)}) - y^{(i)}\right)^2$$

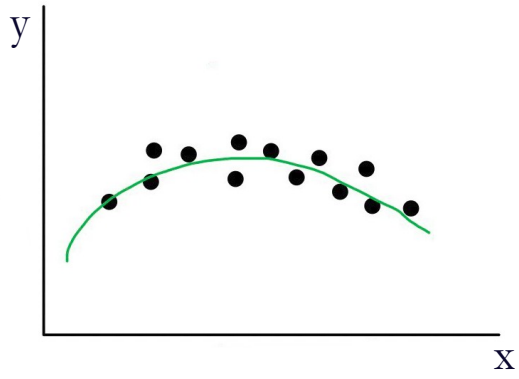
* "the equivalent of ensuring a pupil can't crib the answer sheet from last year's exam" – Blaise Agüera, Google

Wybór i walidacja parametrów modelu

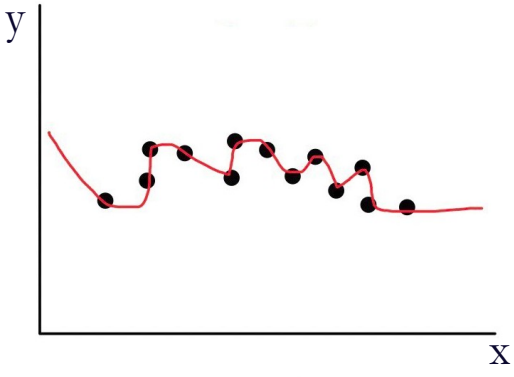
Przykład: regresja liniowa, dobór stopnia wielomianu:



$\theta_0 + \theta_1 x$

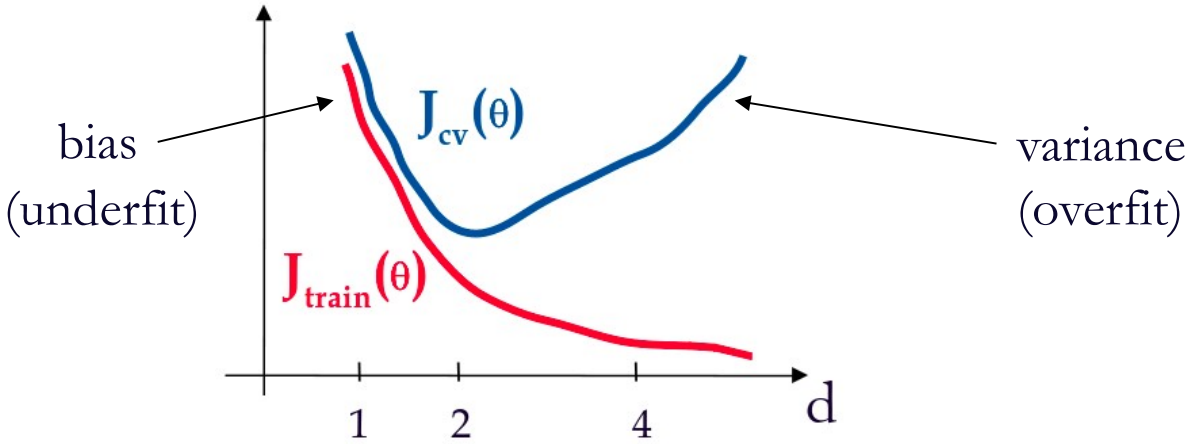


$\theta_0 + \theta_1 x + \theta_2 x^2$



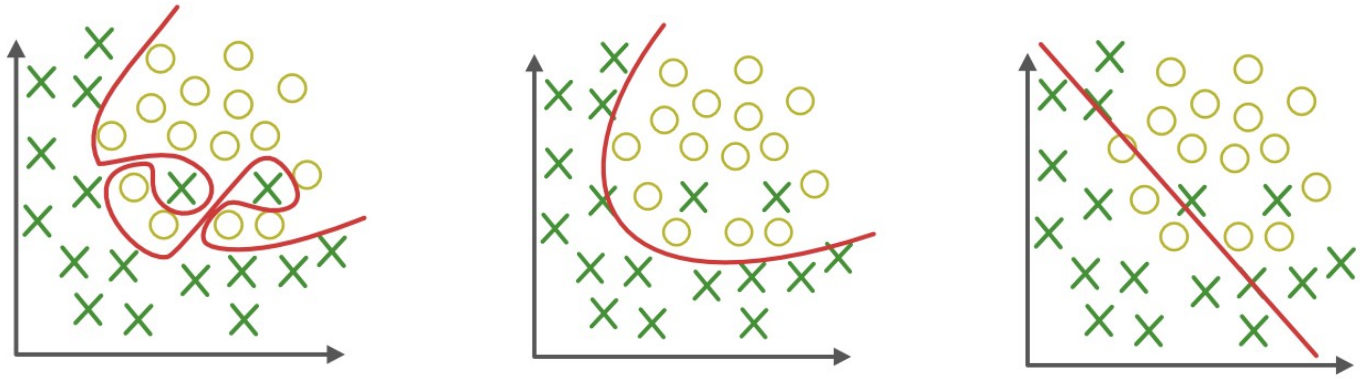
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Najwyższy stopień wielomianu: → d=1 d=2 d=4



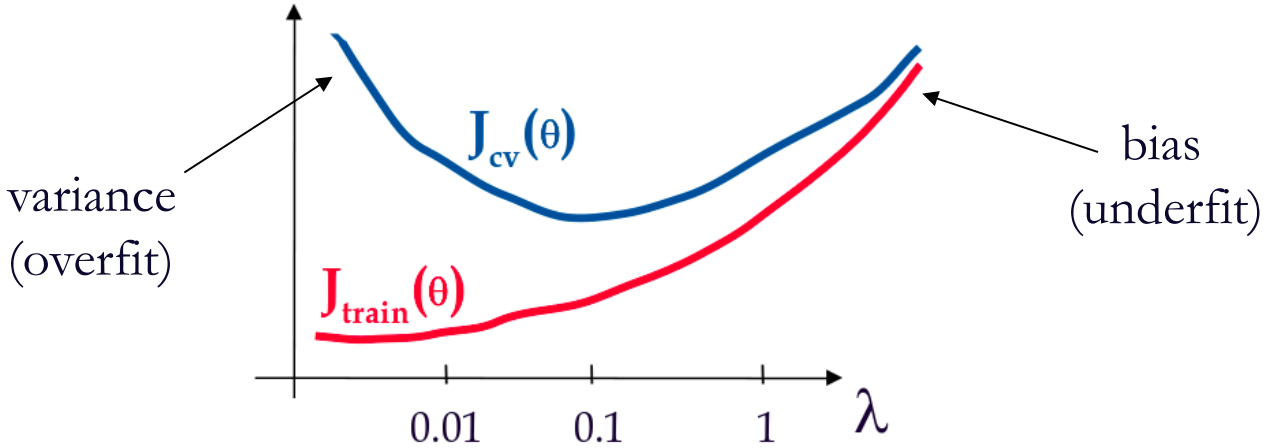
Wybór i walidacja parametrów modelu

Przykład: regresja logistyczna z regularyzacją, dobór λ :



Parametr regularyzacji:

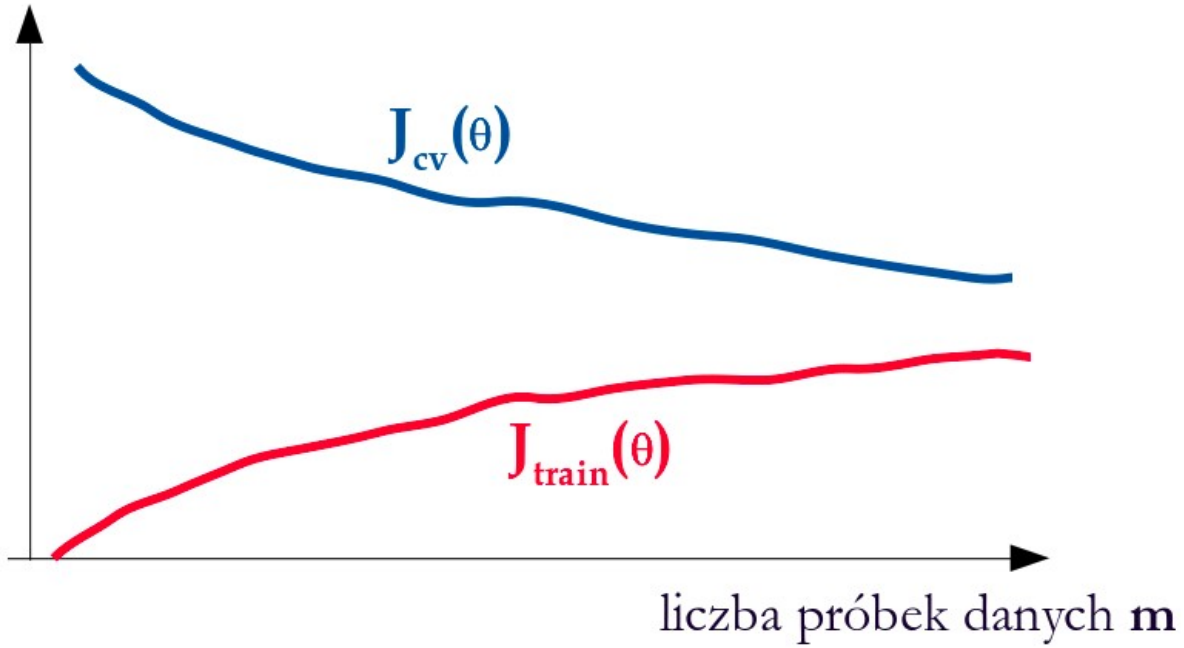
λ=0 λ=0.1 λ=1



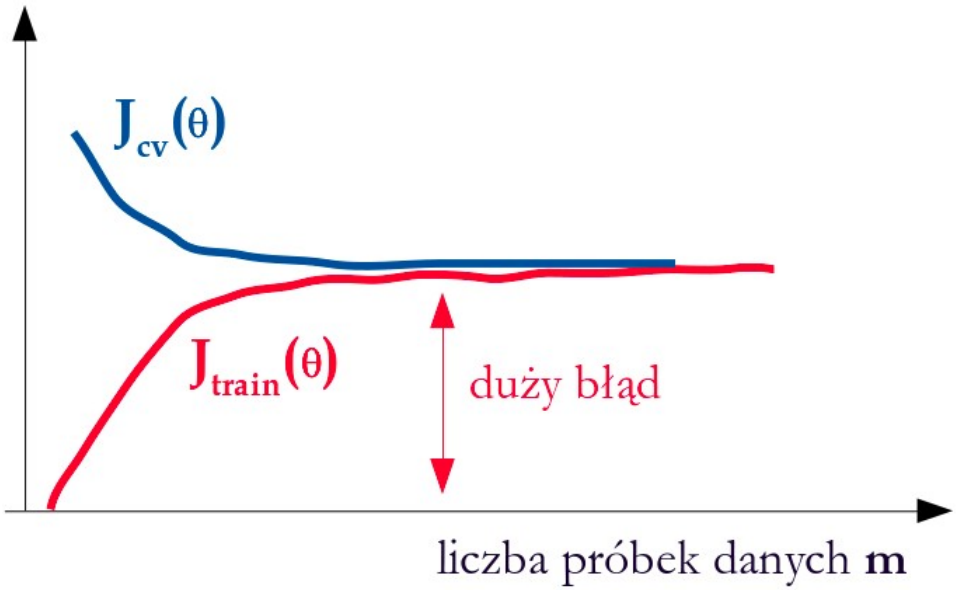
Krzywe uczenia

learning curves

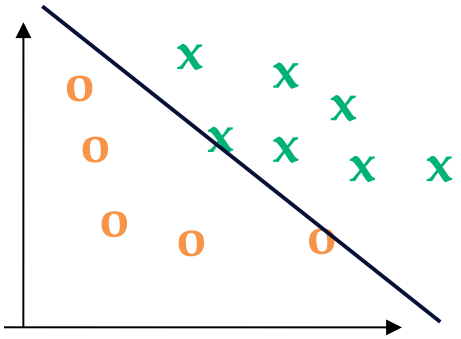
Wyznaczamy je dla konkretnego modelu (ustalony stopień wielomianu, ustalony parametr λ) sztucznie ograniczając próbkę danych:



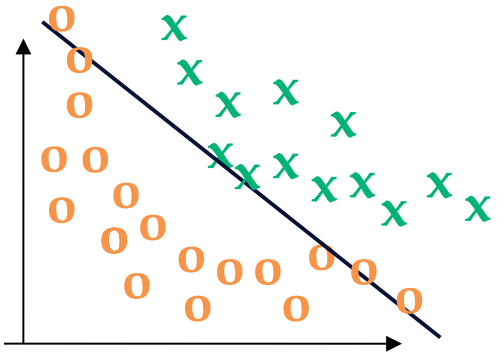
Krzywe uczenia – przykład high bias



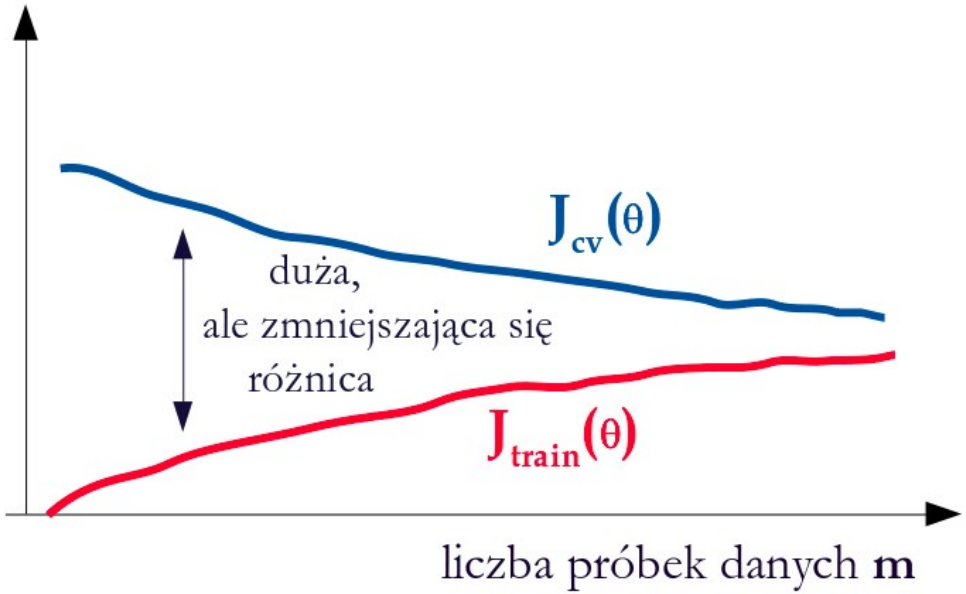
mało danych:



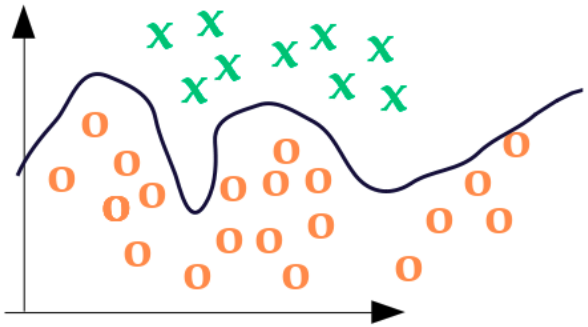
więcej danych:



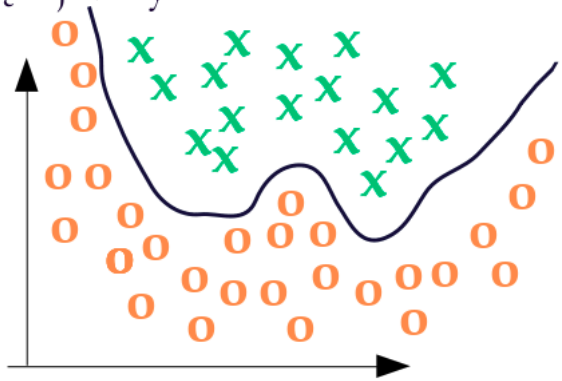
Krzywe uczenia – przykład high variance



mało danych:

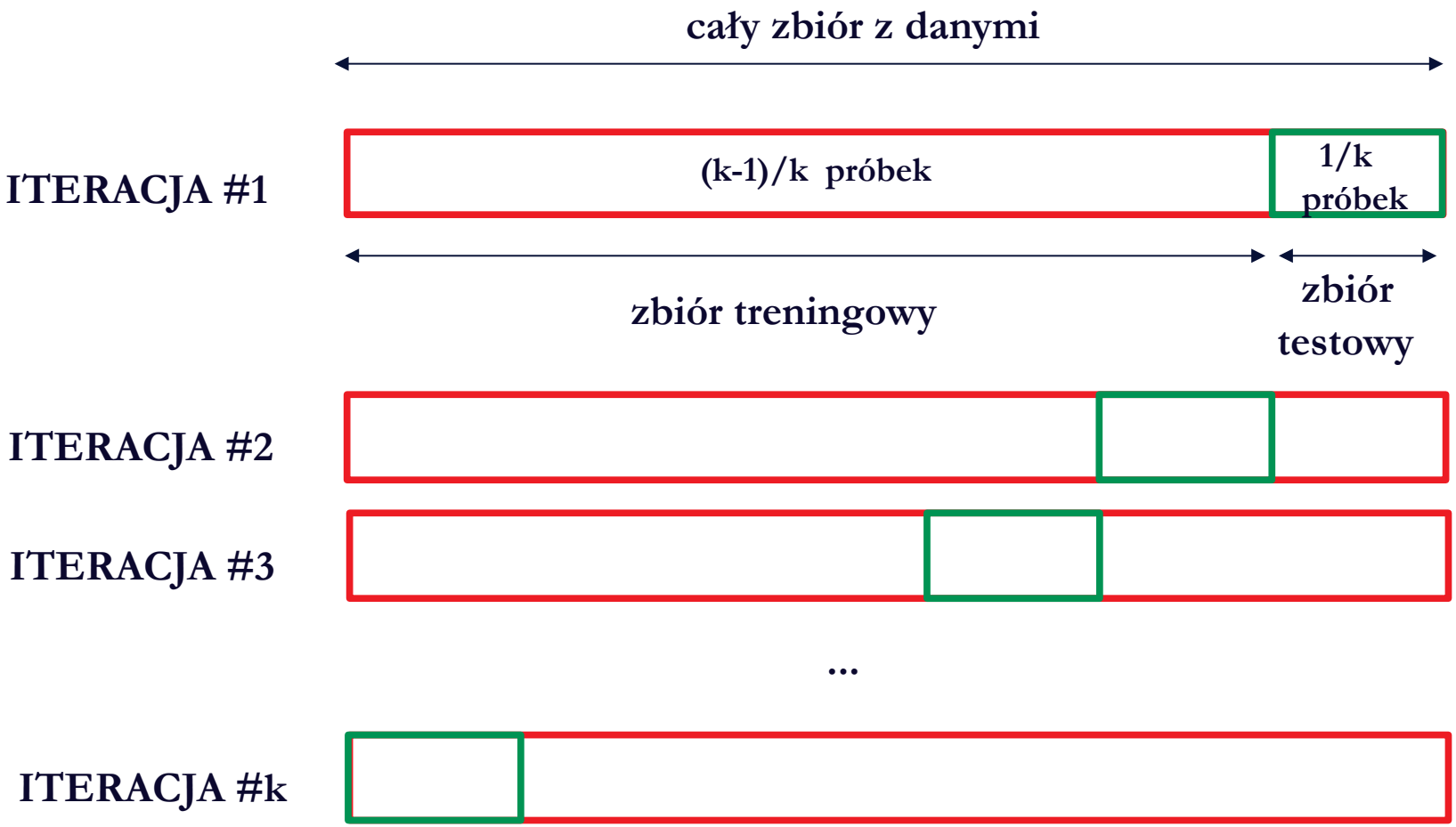


więcej danych:



Walidacja krosowa

k-fold cross validation



$\Rightarrow$ końcowe parametry (np. *accuracy*) uśredniamy po k iteracjach

Diagnostyka - podsumowanie

Możliwe strategie:

- dodać więcej danych wejściowych (próbek)?
  → pomoże w przypadku *high variance*
- zwiększyć wymiar danych (liczbę cech)?
  → pomoże w przypadku *high bias*
- dodać nowe dane – wielomiany cech istniejących?
  → pomoże w przypadku *high bias*
- zmniejszyć wymiar danych?
  → pomoże w przypadku *high variance*
- regularyzacja – większa λ?
  → pomoże w przypadku *high variance*
- regularyzacja – mniejsza λ?
  → pomoże w przypadku *high bias*

Prosty przykład kodowy *

*** załączony plik jupyter-notebook:**

http://tele.agh.edu.pl/~kulakowski/ml/logistic-regression.ipynb

Dziękuję za spotkanie!

Pytania?

Komentarze?