Springer Series in Statistics

Trevor Hastie
Robert Tibshirani
Jerome Friedman

The Elements of Statistical Learning

Data Mining, Inference, and Prediction

Second Edition

Springer

Wybór modelu i ocena jakości klasyfikatora

- Błąd uczenia i błąd testowania
- Obciążenie, wariancja i złożoność modelu (klasyfikatora)
- Dekompozycja błędu testowania
- Optymizm
- Estymacja błędu testowania
  - AIC, BIC, MDL, VC, SRM
  - Walidacja krzyżowa, Bootstrap

Błąd uczenia i błąd testowania

X – zmienna wejściowa (np. wektor cech)

Y – zmienna wyjściowa (np. numer klasy)

$Y^t = f(X)$ – prawdziwa zależność między Y a X

$Y = f(X) + \varepsilon$ – mierzona zależność między Y a X

ε – zakłócenia

$T = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ – zbiór obserwacji (np. zbiór uczący)

$L[Y, f(X)]$ – funkcja strat, np. $\left(Y - \hat{f}(X)\right)^2$ (błąd średniokwadratowy)

$I\left(Y \neq \hat{f}(X)\right)$ (zerojedynkowa funkcja strat)

Błąd uczenia (*training error*):

$$\overline{err} = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, \hat{f}\left(x_i\right)\right)$$

Błąd uogólniania (*generalization error, test error*):

$$Err_T = \mathrm{E}_{Y,X}\left[L\left(Y, \hat{f}\left(X\right)\right)\middle| Z\right]$$

jest to błąd wyznaczony na niezależnym zbiorze testującym T

Średni błąd testowania (uogólniania) (*expected test error*):

$$Err = \mathrm{E}_{T}\left[Err_T\right]$$

Obciążenie, wariancja i złożoność

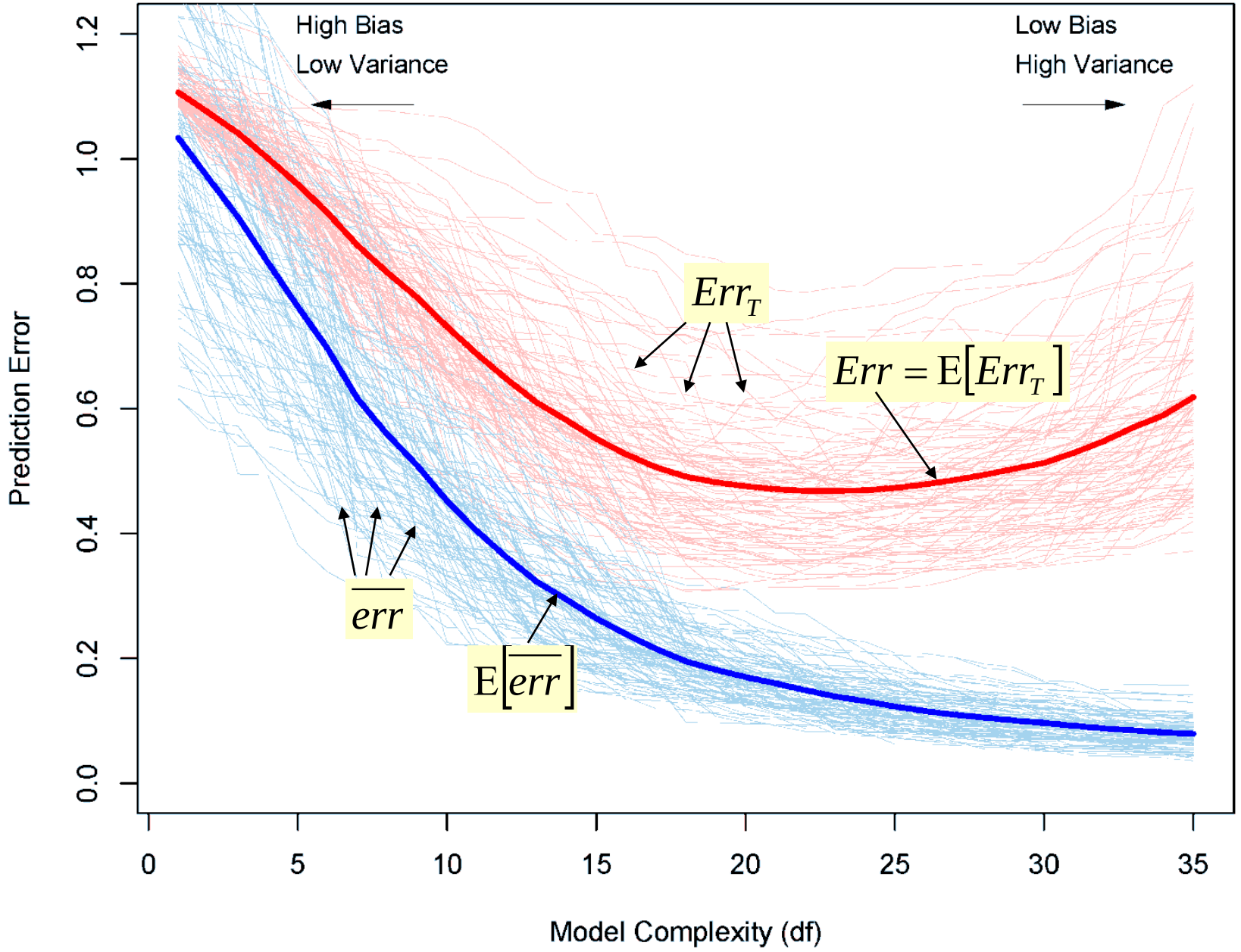
Złożoność modelu (klasyfikatora)

- liczba parametrów
- stopień wielomianu
- liczba funkcji bazowych
- liczba splajnów funkcji sklejanej
- liczba reguł
- liczba neuronów i warstw sieci neuronowej
- wartość parametru (np. *k* w algorytmie *k*-NN)
- liczba stopni swobody modelu

α – parametr określający złożoność modelu

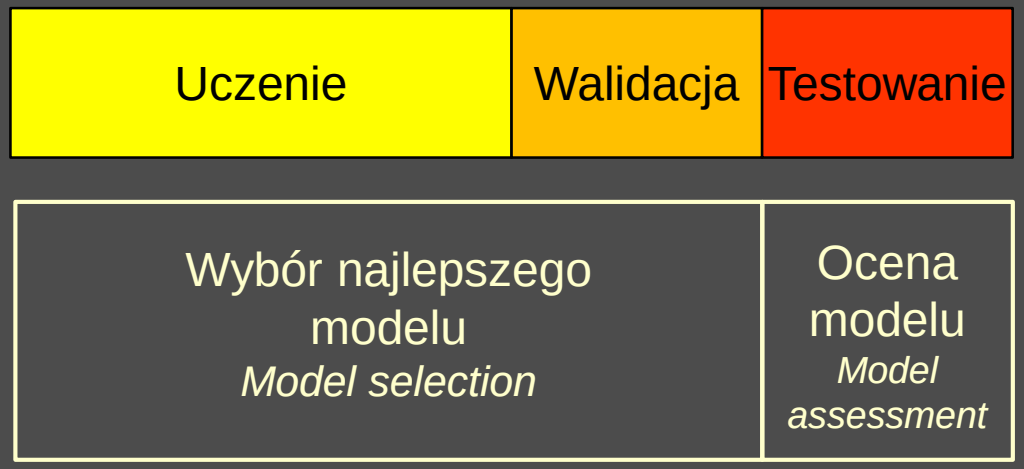
$\hat{f}_\alpha(X)$ – model (klasyfikator) o złożoności α

Jak znaleźć wartość α, dla której średni błąd testowania *Err* przyjmuje wartość najmniejszą?


High Bias
Low Variance
Low Bias
High Variance
Err_T
$Err = \mathrm{E}[Err_T]$
$\overline{err}$
$\mathrm{E}[\overline{err}]$
Prediction Error
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20
25
30
35
Model Complexity (df)

- Jak wyznaczać błędy testowania dla dużych prób?

Losowy podział zbioru uczącego:



- Metody przybliżonego wykonania walidacji
  - Kryteria analityczne
    - AIC – *Akaike Information Criterion*
    - BIC – *Bayesian Information Criterion*
    - MDL – *Minimum Description Length*
    - Wymiar VC (Vapnika-Chernovenkisa) i SRM – *Structural Risk Minimization*
  - Wielokrotny podział próby
    - Walidacja krzyżowa (CV – *Cross Validation*)
    - Bootstrap

Dekompozycja błędu testowania

$$Y = f(X) + \varepsilon$$

Załóżmy, że $E[\varepsilon] = 0$ oraz $Var[\varepsilon] = \sigma_\varepsilon^2$

Dla ustalonego wejścia $X = \mathbf{x}$ błąd testowania (uogólniania) ma postać:

$$Err(\mathbf{x}) = \mathrm{E}\left[L\left(Y, \hat{f}(\mathbf{x})\right) \middle| X = \mathbf{x} \right]$$

Przyjmując średniokwadratową funkcję strat L, możemy zdekomponować błąd testowania na obciążenie i wariancję.

$$\mathrm{E}\left[\left(Y-\hat{f}(\mathbf{x})\right)^2 \middle| X=\mathbf{x}\right] =$$

$$= \mathrm{E}\left[\left(Y-f(\mathbf{x})+f(\mathbf{x})-\hat{f}(\mathbf{x})\right)^2\right] =$$

$$= \mathrm{E}\left[\left(Y-f(\mathbf{x})\right)^2\right] + \mathrm{E}\left[\left(f(\mathbf{x})-\hat{f}(\mathbf{x})\right)^2\right] + 2\mathrm{E}\left[\left(Y-f(\mathbf{x})\right)\left(f(\mathbf{x})-\hat{f}(\mathbf{x})\right)\right] =$$

$$= \mathrm{E}\left[\varepsilon^2\right] + \mathrm{E}\left[\left(f(\mathbf{x})-\hat{f}(\mathbf{x})\right)^2\right] + 2\left(\mathrm{E}\left[Yf(\mathbf{x})\right] - \mathrm{E}\left[Y\hat{f}(\mathbf{x})\right] - \mathrm{E}\left[f^2(\mathbf{x})\right] + \mathrm{E}\left[f(\mathbf{x})\,\hat{f}(\mathbf{x})\right]\right) =$$

$$= \sigma_\varepsilon^2 + \mathrm{E}\left[\left(f(\mathbf{x})-\hat{f}(\mathbf{x})\right)^2\right] + 2\left(\mathrm{E}[Y]f(\mathbf{x}) - \mathrm{E}[Y]\mathrm{E}\left[\hat{f}(\mathbf{x})\right] - f^2(\mathbf{x}) + f(\mathbf{x})\mathrm{E}\left[\hat{f}(\mathbf{x})\right]\right) =$$

$$\mathrm{E}[Y] = f(\mathbf{x})$$

$$= \sigma_\varepsilon^2 + \mathrm{E}\left[\left(f(\mathbf{x})-\hat{f}(\mathbf{x})\right)^2\right] + 2\left(f^2(\mathbf{x}) - f(\mathbf{x})\mathrm{E}\left[\hat{f}(\mathbf{x})\right] - f^2(\mathbf{x}) + f(\mathbf{x})\mathrm{E}\left[\hat{f}(\mathbf{x})\right]\right) =$$

$$= \sigma_\varepsilon^2 + \mathrm{E}\left[\left(f(\mathbf{x})-\hat{f}(\mathbf{x})\right)^2\right]$$

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

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

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

$$+ 2\mathrm{E}\left[\left(f(\mathbf{x})-\mathrm{E}\left[\hat{f}(\mathbf{x})\right]\right)\left(\mathrm{E}\left[\hat{f}(\mathbf{x})\right]-\hat{f}(\mathbf{x})\right)\right] =$$

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

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

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

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

$$= Bias^2\left[\hat{f}(\mathbf{x})\right]+Var\left[\hat{f}(\mathbf{x})\right]$$

Błąd testowania został zdekomponowany na:

$$Err(\mathbf{x}) = \sigma_\varepsilon^2 + Bias^2\left[\hat{f}(\mathbf{x})\right] + Var\left[\hat{f}(\mathbf{x})\right]$$

wariancja zakłóceń – nie można jej zredukować, gdyż nie zależy od modelu

Błąd testowania *Err* minimalizujemy poprzez odpowiedni dobór $\hat{f}$

Problem: Zmiany $\hat{f}$ prowadzące do obniżenia obciążenia $Bias^2\left[\hat{f}(\mathbf{x})\right]$ prowadzą jednocześnie do podwyższenia wariancji $Var\left[\hat{f}(\mathbf{x})\right]$ i odwrotnie

Przykład 1: algorytm *k*-NN

$$
\begin{aligned}
Err(\mathbf{x}) &= \mathrm{E}\left[\left(Y - \hat{f}(\mathbf{x})\right)^2 \,\middle|\, X = \mathbf{x} \right] = \\
&= \sigma_\varepsilon^2 + Bias^2\left[\hat{f}(\mathbf{x})\right] + Var\left[\hat{f}(\mathbf{x})\right] = \\
&= \sigma_\varepsilon^2 + \left[f(\mathbf{x}) - \frac{1}{k}\sum_{\ell=1}^{k} f\left(x_{(\ell)}\right) \right]^2 + \frac{\sigma_\varepsilon^2}{k}
\end{aligned}
$$

gdzie ℓ indeksuje sekwencję *k* najbliższych sąsiadów wektora **x**.

Mała wartość k → małe obciążenie, duża wariancja

Duża wartość k → duże obciążenie, mała wariancja

Złożoność modelu ≡ 1/k

Przykład 2: model (klasyfikator) liniowy $\hat{f}_p(\mathbf{x}) = \mathbf{a}^T \varphi(\mathbf{x})$

p – liczba parametrów modelu

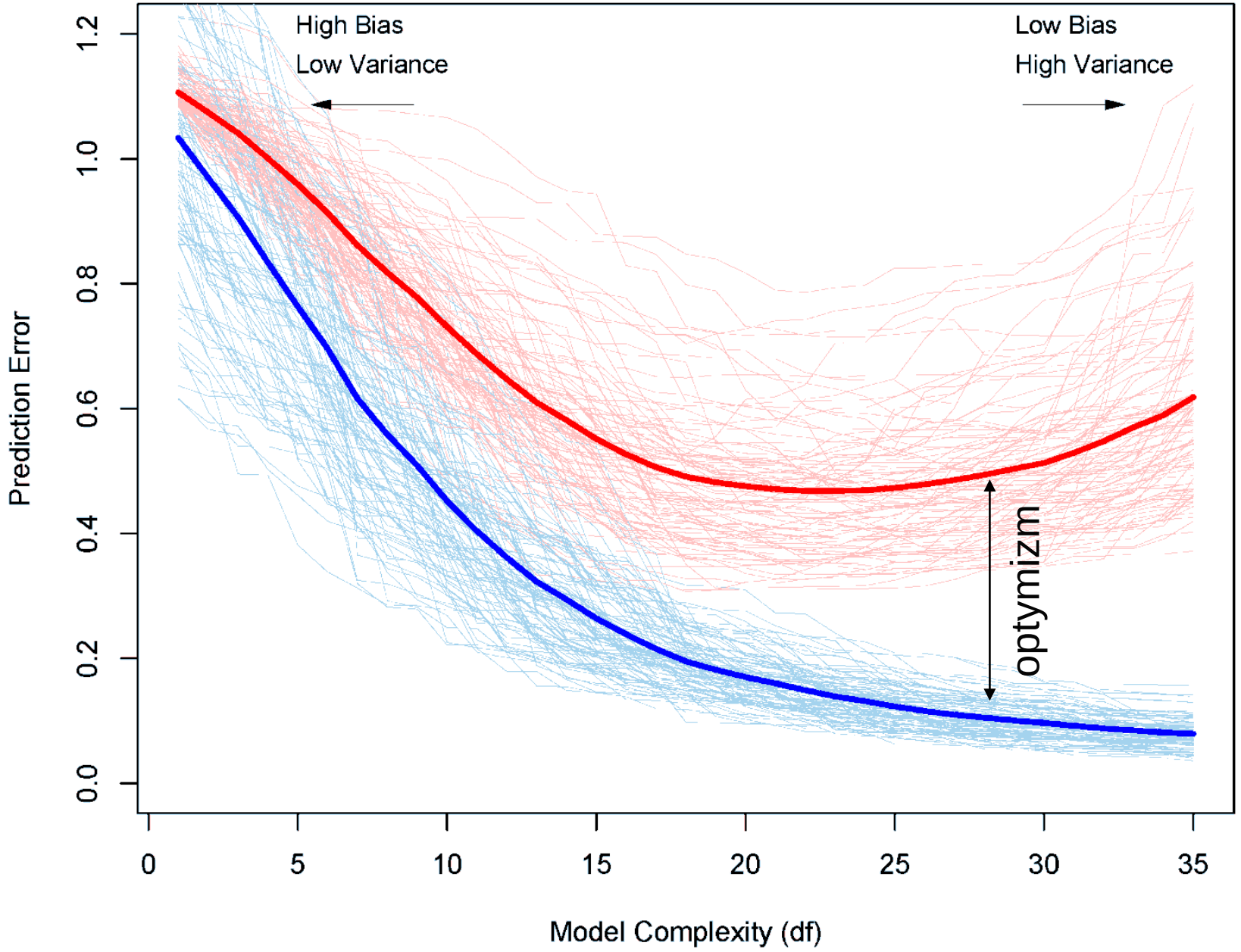
$$\frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} Err(x_i) = \sigma_\varepsilon^2 + Bias^2\left[\hat{f}_p(\mathbf{x})\right] + Var\left[\hat{f}_p(\mathbf{x})\right] =$$

$$= \sigma_\varepsilon^2 + \frac{1}{\mathrm{N}} \sum_{i=1}^{N} \left(f(x_i) - \mathrm{E}\left[\hat{f}_p(x_i)\right]\right)^2 + \frac{p}{N}\sigma_\varepsilon^2$$

Mała wartość p → duże obciążenie, mała wariancja

Duża wartość p → małe obciążenie, duża wariancja

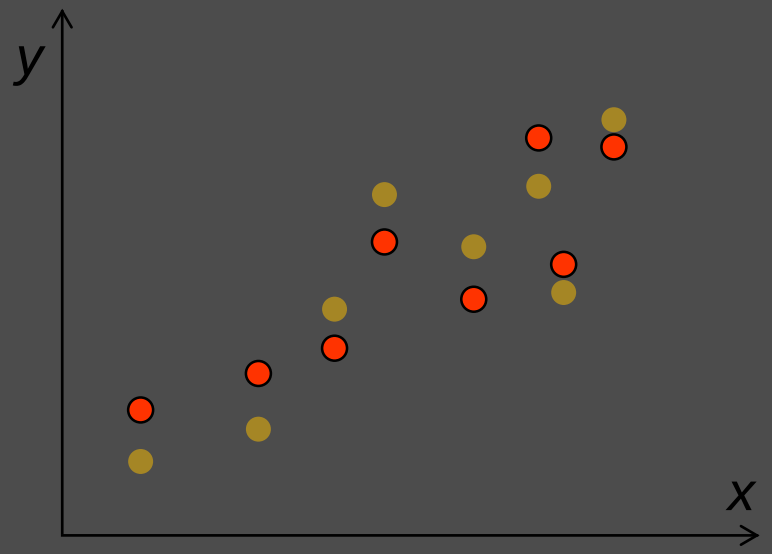
Złożoność modelu ≡ p


High Bias
Low Variance
Low Bias
High Variance
optymizm
Prediction Error
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20
25
30
35
Model Complexity (df)

Optymizm

Błąd $\overline{err}$ na zbiorze uczącym nie doszacowuje błędu Err_T na zbiorze testującym (jest zbyt „optymistyczny”)

Jak można „zmierzyć” ten optymizm?



Dla wszystkich N wejść $\mathbf{x}_i$ ze zbioru T zaobserwujmy ponownie wyjścia y_i, otrzymując w rezultacie nowy zbiór. Wielokrotne wykonanie tej procedury daje podstawę do zdefiniowania błędu Err_{in}:

$$Err_{\text{in}} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{E}_{Y}\left[L\left(Y_i, \hat{f}(x_i)\right) \middle| T \right]$$

Optymizm zdefiniowany jest jako: $op = Err_{\text{in}} - \overline{err}$ (na ogół $op > 0$)

Średni optymizm: $\omega = \mathrm{E}_{Y}(op)$

Można pokazać, że dla błędu średniokwadratowego i dowolnej funkcji strat zachodzi:

$$\omega = \frac{2}{N}\sum_{i=1}^{N} Cov\left(\hat{y}_i, y_i\right)$$

$$\mathop{E}_{Y}\left[Err_{\text{in}}\right] = \mathop{E}_{Y}\left[\overline{err}\right] + \frac{2}{N}\sum_{i=1}^{N} Cov\left(\hat{y}_i, y_i\right)$$

Im bardziej model jest dopasowany do danych w zbiorze T, tym większa kowariancja $Cov\left(\hat{y}_i, y_i\right)$, a co za tym idzie – tym większy optymizm.

Przykład: dla modelu liniowego o *d* wejściach lub funkcjach bazowych zachodzi:

$$\sum_{i=1}^{N} Cov\left(\hat{y}_i, y_i\right) = d\sigma_\varepsilon^2$$

więc

$$\omega = 2\frac{d}{N}\sigma_\varepsilon^2$$

Optymizm rośnie ze wzrostem *d*, ale maleje ze wzrostem rozmiaru *N* próby.

Zamiast estymować błąd *Err* przy użyciu złożonej procedury, będziemy w jego miejsce stosować prostszy estymator błędu Err_{in} :

$$\hat{Err}_{\text{in}} = \overline{err} + \hat{\omega}$$

Estymacji tego błędu dokonujemy dla zbioru modeli (klasyfikatorów) $\hat{f}_\alpha$, gdzie α jest parametrem opisującym złożoność modelu.

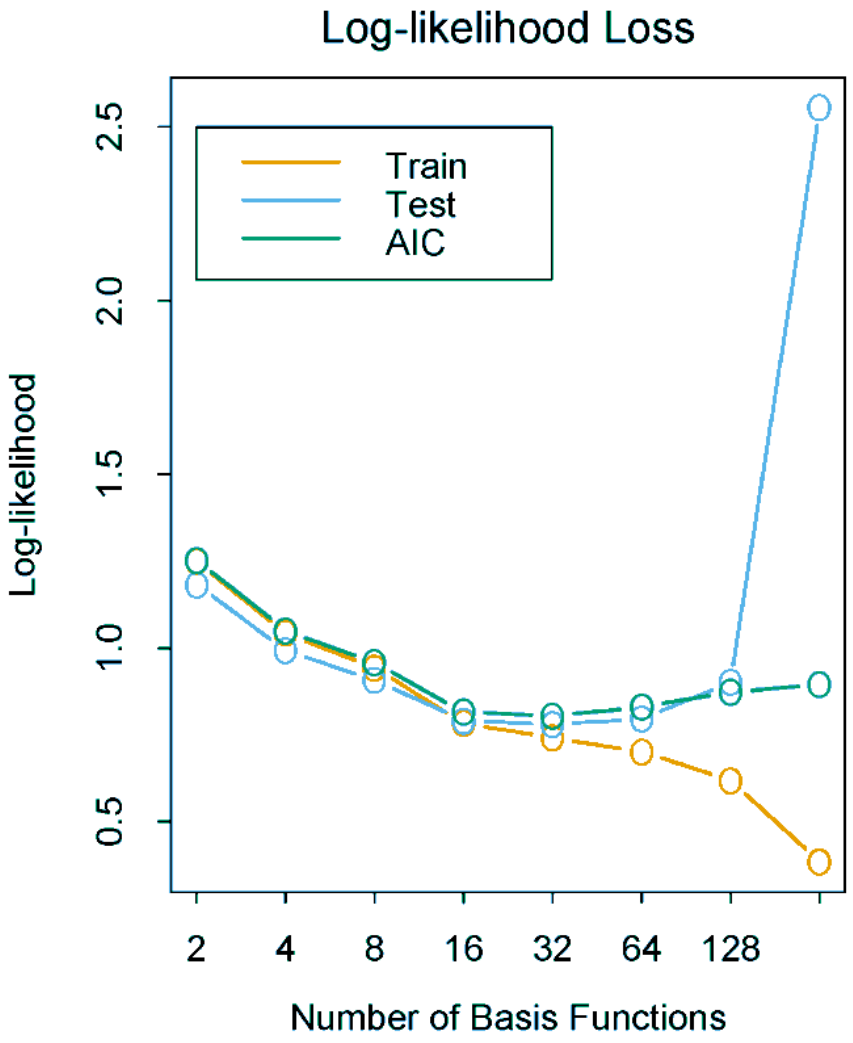
- AIC – *Akaike Information Criterion*

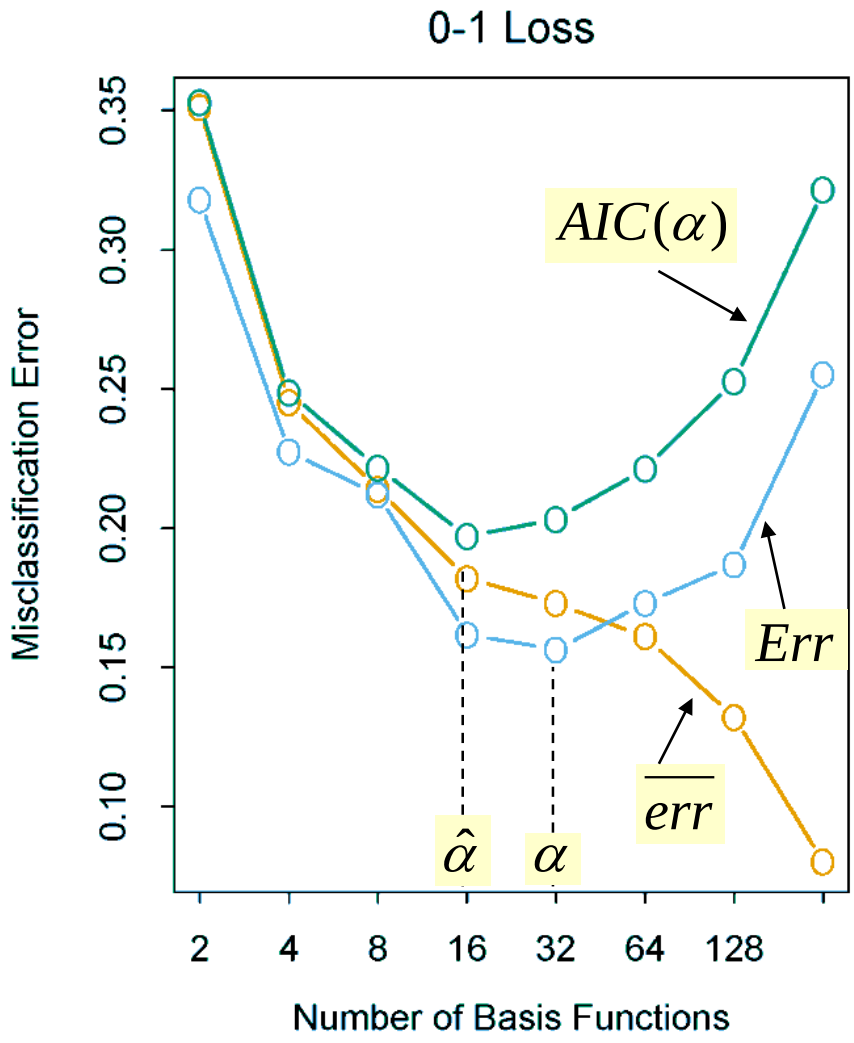
$$AIC(\alpha) = \overline{err}(\alpha) + 2\frac{d(\alpha)}{N}\hat{\sigma}_\varepsilon^2$$

$\hat{\sigma}_\varepsilon^2$ – estymator wariancji zakłóceń wyznaczony z wykorzystaniem błędu średniokwadratowego modelu o niskim obciążeniu

Szukamy modelu o takiej złożoności $\hat{\alpha}$, dla którego AIC(α) przyjmuje wartość najmniejszą. Uzyskany model to $\hat{f}_{\hat{\alpha}}$.

Przykład: rozpoznawanie fonemów





- BIC – *Bayesian Information Criterion*

$$BIC(\alpha) = \frac{N}{\hat{\sigma}_\varepsilon^2}\left[\overline{err}(\alpha) + \log N \frac{d(\alpha)}{N}\ \hat{\sigma}_\varepsilon^2\right]$$

  - Prowadzi do modelu o największym prawdopodobieństwie a'posteriori
  - Jeżeli klasa modeli zawiera „prawdziwy" model, to prawdopodobieństwo, że BIC do niego doprowadzi zmierza do 1 dla $N \to \infty$.
  - Dla małych prób BIC prowadzi do zbyt prostych modeli
  - Dla $N \to \infty$ AIC prowadzi do zbyt złożonych modeli
  - Małe N – lepsze AIC, duże N – lepsze BIC

- MDL – *Minimum Description Length*
  - Jest równoważne BIC, lecz wyprowadzone z teorii kodowania

wiadomość	z_1	z_2	z_3	z_4
kod	0	10	110	1110

  - Jak przyporządkować kody do wiadomości, aby minimalizować średnią długość wiadomości?
  - Częste wiadomości → krótsze kody, a dokładnie:

$$\text{długość kodu } z_i = \log P(z_i) \quad \text{(Shannon)}$$

$$\text{średnia długość wiadomości} = -\sum_i P(z_i) \log P(z_i)$$

- Analogia do wyboru modelu:
  - Odbiorca zna
    - wejścia X,
    - prawdopodobieństwa warunkowe wyjść y
  - Chcemy przesłać wiadomość y
  - Wówczas minimalizacja długości wiadomości prowadzi do modelu maksymalizującego prawdopodobieństwo a’posteriori, a tym samym minimalizujące BIC

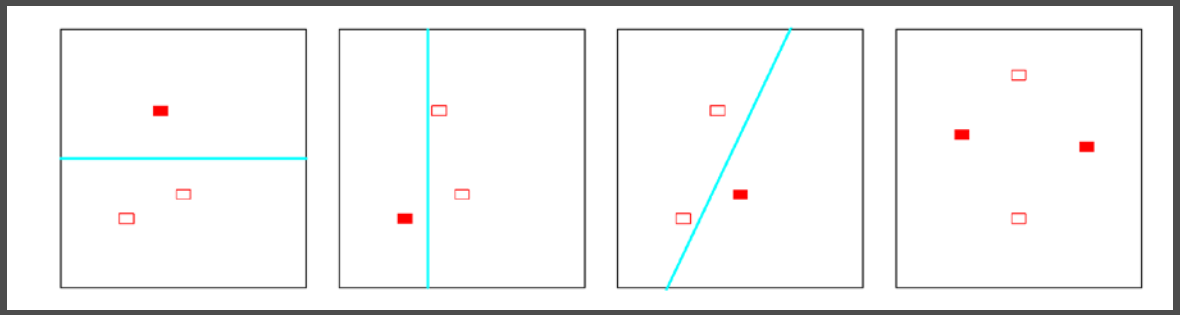
- VC – *wymiar Vapnika-Chernovenkisa*
  - Ogólna miara złożoności modelu (klasyfikatora)

Klasa modeli $\{ f(\mathbf{x}, \alpha) \}, \ \mathbf{x} \in \mathscr{R}^p$

Przykład 1 – klasyfikator binarny z dwoma parametrami:

$$\alpha = (\alpha_0, \alpha_1) \qquad f(\mathbf{x}, \alpha) = I(\alpha_0 + \alpha_1^T \mathbf{x} > 0)$$

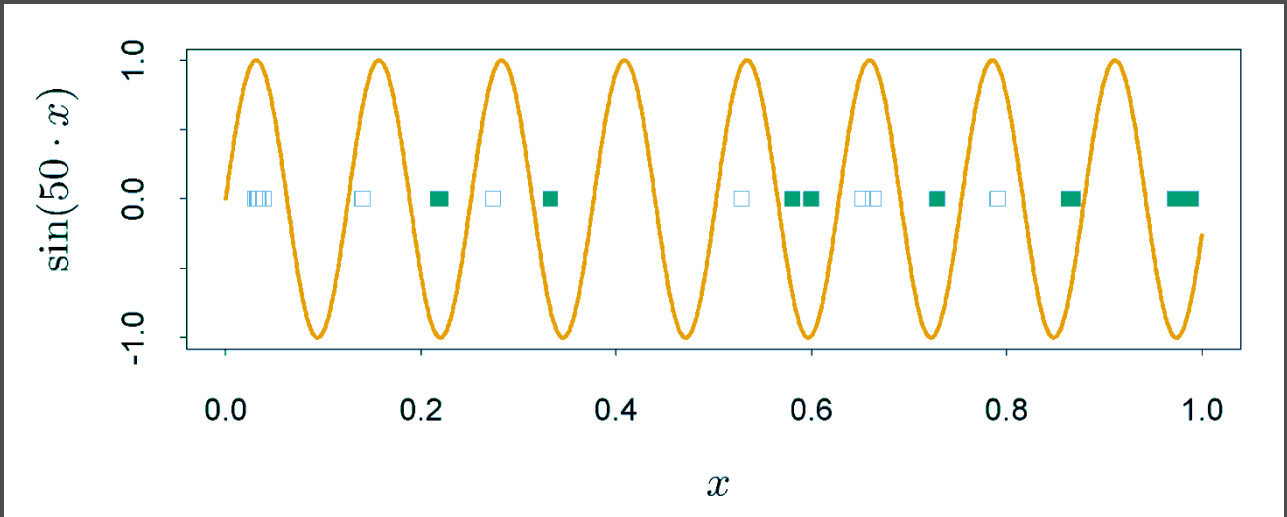
może odseparować na płaszczyźnie do trzech punktów



Przykład 2 – klasyfikator sinusoidalny z jednym parametrem:

$$f(\mathbf{x},\alpha) = I(\sin \alpha\, \mathbf{x} > 0), \qquad \alpha \in \mathscr{R}, \mathbf{x} \in \mathscr{R}^2$$

może odseparować na płaszczyźnie dowolny zbiór punktów, przy odpowiednio dobranej wartości parametru α



Pytanie: modele której z tych dwóch klas są bardziej złożone?

- **Def**. Wymiar Vapnika-Chernovenkisa (VC) klasy $f(\mathbf{x},\alpha)$ jest równy największej liczbie punktów, które mogą być odseparowane przez klasyfikatory z tej klasy.
- Wymiar VC p-wymiarowego klasyfikatora liniowego jest równy $p + 1$
- Wymiar VC modeli z klasy $\sin \alpha\mathbf{x}$ jest nieskończony
- Dla funkcji $g(x,\alpha)$ przyjmującej wartości rzeczywiste, wymiar VC jest zdefiniowany jako wymiar VC klasy $I\left(g\left(\mathbf{x},\alpha\right)-\beta>0\right)$, gdzie β przyjmuje wartości z tej samej dziedziny, co funkcja g

- Związek VC i optymizmu
  - Jeżeli dopasowujemy *N* punktów za pomocą modeli z klasy f(x,α) o wymiarze VC równym *h*, wówczas z prawdopodobieństwem 1 – η zachodzi:

$$Err_T \le \overline{err} + \frac{\xi}{2}\left(1 + \sqrt{1 + \frac{4\overline{err}}{\xi}}\right)$$

(klasyfikator binarny)

$$Err_T \le \frac{\overline{err}}{\left(1 - c\sqrt{\xi}\right)_+}$$

(regresja)

gdzie: $\xi = a_1 \frac{h\left[\log\left(a_2 N / h\right) + 1\right] - \log\left(\eta / 4\right)}{N}$, $a_1 \in (0,4],\ a_2 \in (0,2]$

Zaleca się $c = 1$, a dla regresji $a_1 = a_2 = 1$.

- Alternatywne, praktyczne ograniczenie dla regresji:

$$Err_T \le \overline{err}\left(1 - \sqrt{h/N\left(1 - \log\left(h/N\right)\right) + \frac{\log N}{2N}}\right)_+^{-1}$$

- Widać, że optymizm rośnie ze wzrostem *h* i maleje ze wzrostem *N*

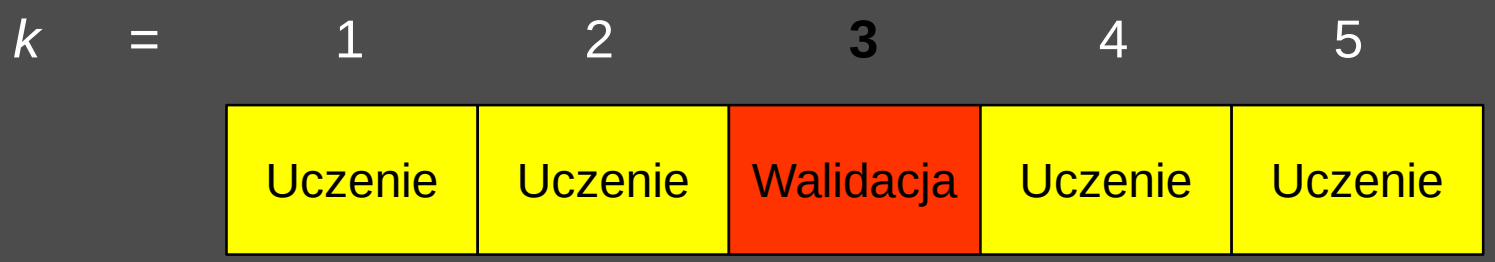
- SRM – *Structural Risk Minimization*
  - Polega na sekwencji uczenia kolejnych zagnieżdżonych modeli o rosnących wymiarach VC: $h_1 < h_2 < \ldots$
  - Wybierany jest model z najmniejszą wartością górnego ograniczenia na Err_T
  - Obliczanie wymiaru VC oraz SRM to trudne zadania obliczeniowe
  - SRM można łatwo przeprowadzić dla klasyfikatorów SVM

Walidacja krzyżowa
Cross Validation (*CV*)

- Dysponując niewielką próbą, staramy się estymować błąd testowania $Err = \mathrm{E}\left[L\left(Y, \hat{f}(X)\right)\right]$

K-krotna walidacja krzyżowa (*K-fold* CV)

- Dzielimy zbiór uczący na *K* mniej więcej równych partycji
- *k*-ty krok (*k* = 1,…,*K*): *k*-tą partycję przeznaczamy na obliczenie błędu testowania, dla pozostałych *K*-1 partycji uczymy model (klasyfikator)



- Oznaczmy przez $\kappa : \{1,\ldots,N\} \to \{1,\ldots,K\}$ funkcję indeksującą, która *i*-tej obserwacji przyporządkowuje partycję

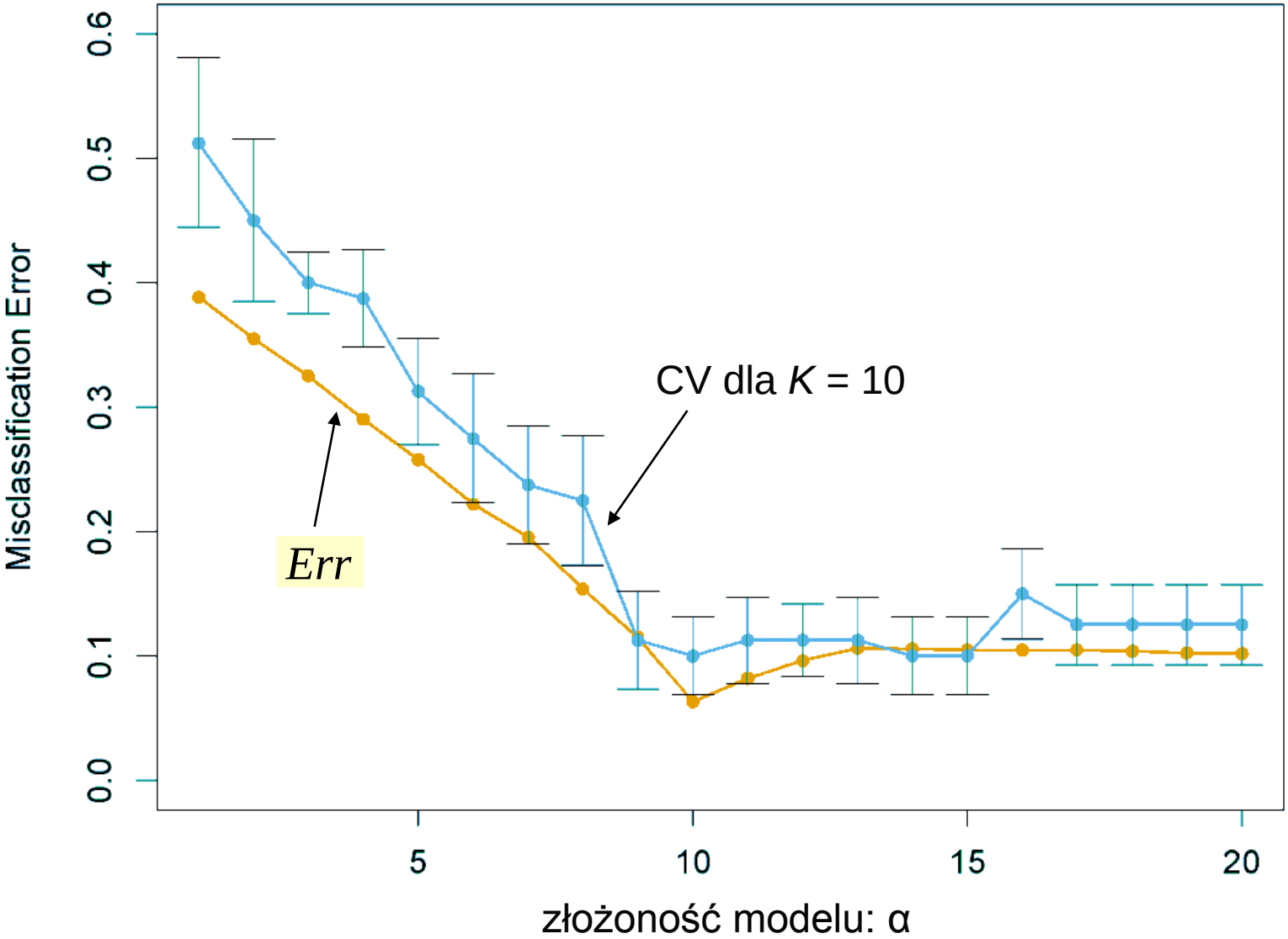
  $\hat{f}^{-k}(\mathbf{x},\alpha)$ – model uczony z pominięciem *k*-tej partycji

- Estymator CV błędu testującego jest dany wzorem:

$$CV(\hat{f},\alpha) = \frac{1}{N}\sum_{i=1}^{N} L\left(y_{i,}\, \hat{f}^{-\kappa(i)}(x_i,\alpha)\right)$$

- Typowe wartości *K* to 5 i 10. Przypadek *K* = *N* nosi nazwę *leave-one-out* CV (wówczas $\kappa(i) = i$).

- W procedurach rozpoznawania obrazów (z etapami grupowania i selekcji cech) należy pamiętać, że partycje metody CV muszą zostać uwzględnione na wszystkich poprzednich etapach


Misclassification Error
0.0
0.1
0.2
0.3
0.4
0.5
0.6
5
10
15
20
złożoność modelu: α
CV dla K = 10
Err

Bootstrap

- Podobnie jak poprzednio, staramy się z niewielkiej próby

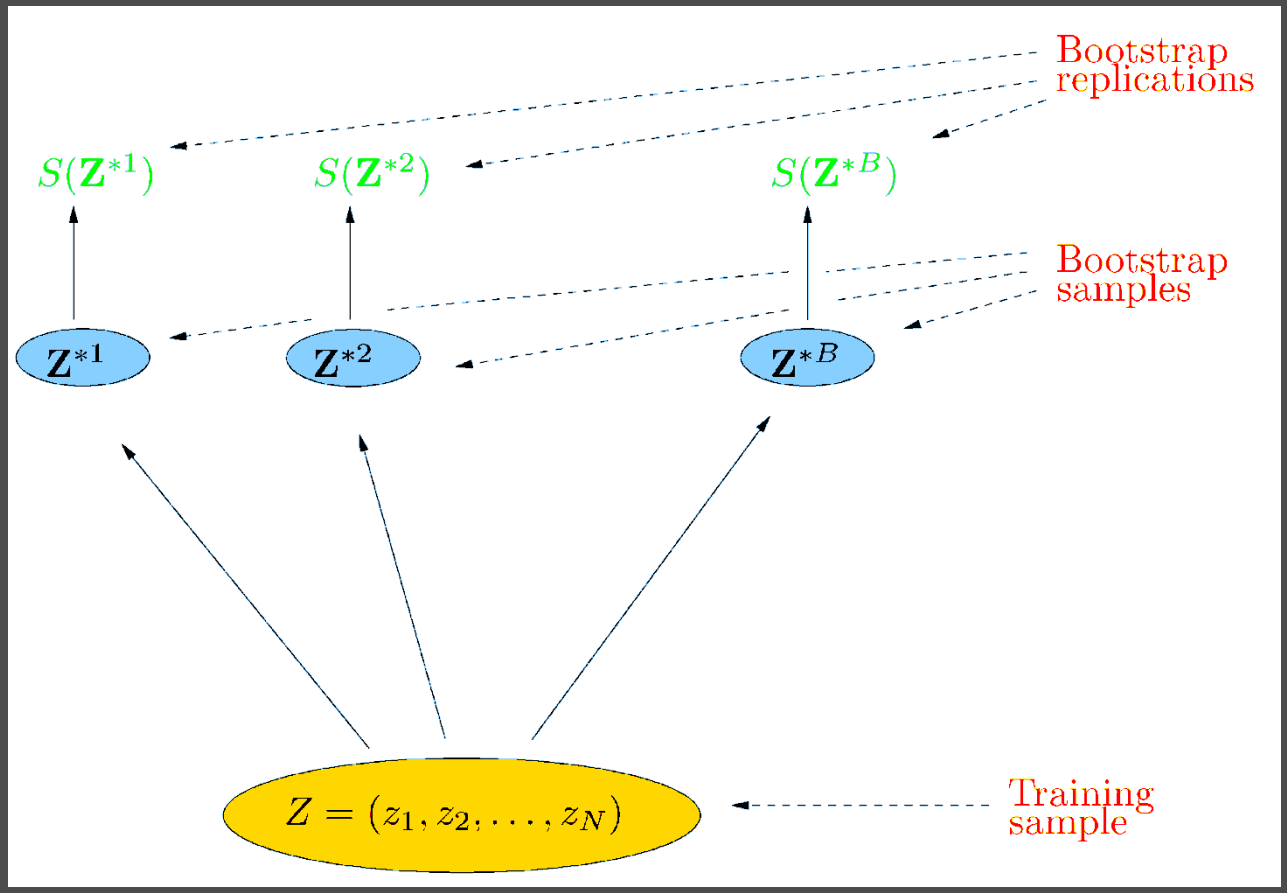
$$Z = \{z_1, z_2, \ldots, z_N\}, \quad z_i = (x_i, y_i)$$

estymować błąd testowania $Err = \mathrm{E}\left[L\left(Y, \hat{f}(X)\right)\right]$

- Próba bootstrapowa – ze zbioru Z zawierającego *N* obserwacji losujemy ze zwracaniem *N* elementów, otrzymując zbiór Z^*
- Bootstrapowa replikacja $S(Z^*)$ – dowolna statystyka wyznaczona z próby bootstrapowej
- Metoda bootstrap pozwala estymować różne parametry rozkładu statystyki $S(Z)$

Metoda bootstrap

- Generujemy *B* bootstrapowych prób Z^{*1},…, Z^{*B}
- Uzyskujemy bootstrapowe replikacje $S(Z^{*1}),\ldots,S(Z^{*B})$



- Traktując bootstrapowe replikacje jak próbę, estymuję wybrane parametry rozkładu $S(Z)$, np. średnią:

$$\bar{S}^* = \frac{1}{B}\sum_{b=1}^{B} S\left(Z^{*b}\right)$$

i wariancję

$$\hat{Var}[S(Z)] = \frac{1}{B-1}\sum_{b=1}^{B}\left(S\left(Z^{*b}\right) - \bar{S}^*\right)^2$$

- Estymacja błędu testowania metodą bootstrap
  - Ze zbioru uczącego Z generujemy *B* bootstrapowych prób $Z^{*1},\ldots, Z^{*B}$
  - Uczymy model na próbach $Z^{*1},\ldots, Z^{*B}$:

    $\hat{f}^{*b}$ – model uczony na *b*-tej próbie bootstrapowej
  - Replikacjami bootstrapowymi $S(Z^{*b};Z)$, $b = 1,2,\ldots,B$ są błędy testowania na oryginalnym zbiorze Z:

$$\hat{Err}_{\text{boot}} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L\left(y_i, \hat{f}^{*b}(x_i)\right)$$

- Jakość estymatora jest niska, gdyż zbiory Z i Z^{*b} mają wiele obserwacji wspólnych (w metodzie CV zbiory uczący i testujący są rozłączne)
- Pomysł: Dla każdej obserwacji $\mathbf{x}_i$ należy uwzględniać tylko błędy wyznaczone z tych prób bootstrapowych, które nie zawierają $\mathbf{x}_i$ (*leave-one-out bootstrap estimate*):

$$\hat{Err}^{(1)} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C^{-i}|}\sum_{b\in C^{-i}} L\left(y_i, \hat{f}^{*b}(x_i)\right)$$

gdzie C^{-i} to zbiór indeksów prób bootstrapowych b nie zawierających $\mathbf{x}_i$, natomiast $|C^{-i}|$ jest liczbą tych prób

- B musi być wystarczająco duże, aby wszystkie $|C^{-i}| > 0$