Strojno učenje

B&V; Ansambli

Tomislav Šmuc

Literatura

- The Elements of Statistical Learning
  Hastie, Tibshirani, Friedman (ch. 15)

- AI – Modern approach
  Russel & Norvig (ch 18.4)

- T. Dieterich*:* Ensemble Methods in Machine Learning
  Lecture Notes in Computer Science, Vol. 1857 (2000), pp. 1-15

- Bagging (L. Breiman)
  Random forests: http://stat-www.berkeley.edu/users/breiman/rf.html
  Bolje: R -package (randomForest); PARF => IRB

- Boosting (www.boosting.org)??
  Y. Freund, Robert E. Schapire: Experiments with a new boosting algorithm. In: Thirteenth International Conference on Machine Learning, San Francisco, 148-156, 1996

Literatura

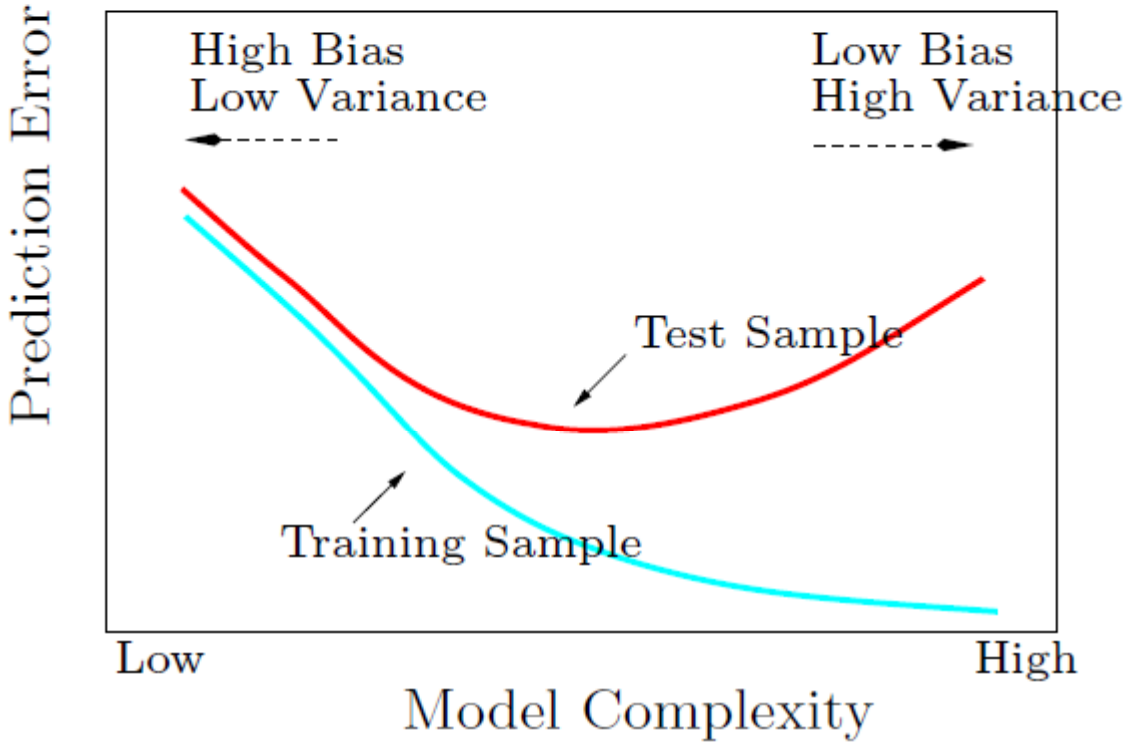


FIGURE 2.11. *Test and training error as a function of model complexity.*

Dekompozicija prediktivne pogreške: Pristranost i varijanca modela

Kod određivanja aproksimacije funkcije
– ciljna varijabla *y* se obično može izraziti kao:

$$y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon$$

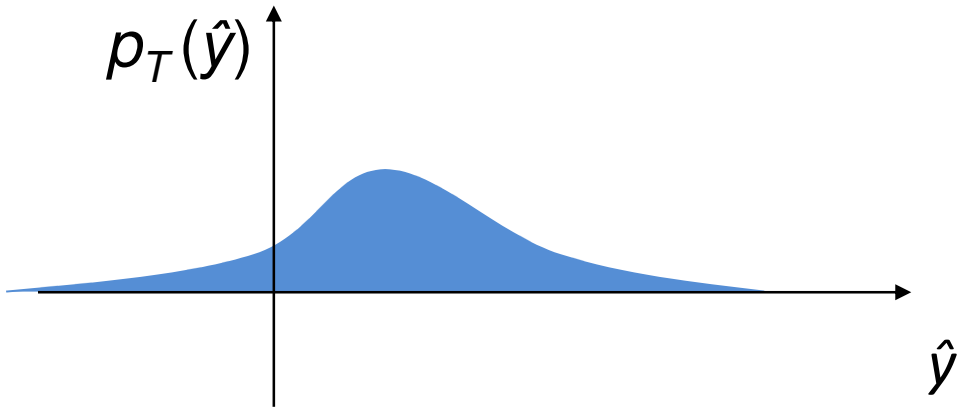
- $f(x)$ - "ciljna funkcija"
- ε - šum; obično $E(\varepsilon \mid \mathbf{x}) = 0$

Uz određene modifikacije slijedeće razmatranje se može promijeniti i na klasifikacijske probleme

Dekompozicija prediktivne pogreške: Pristranost i varijanca modela

Skup za učenje je T slučajno uzorkovan

=> predikcija $\hat{y}$ slučajna varijabla



Dekompozicija prediktivne pogreške: Pristranost i varijanca modela

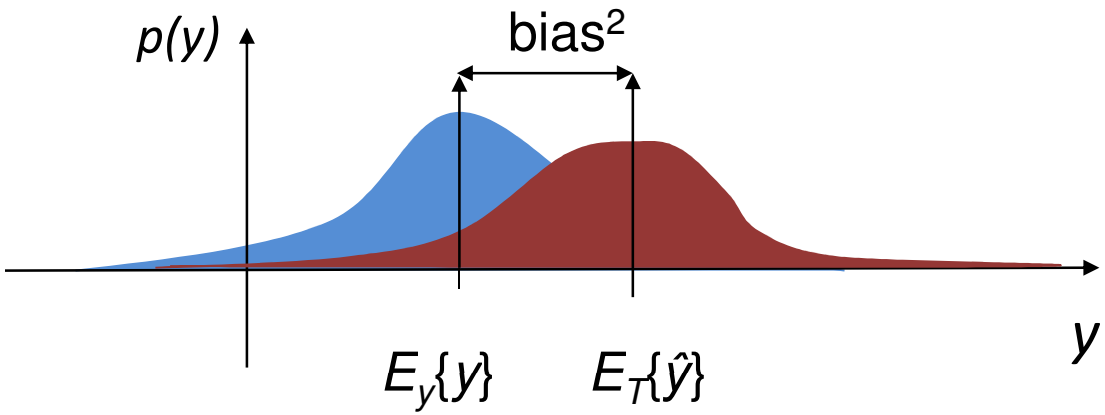
Očekivane vrijednosti

$$E(e) = \text{bias}^2 + \text{varijanca} + \text{šum}$$

$$E_T[\,(y - \hat{y})^2\,] = \underbrace{(E_T[\hat{y}] - y)^2}_{a} + \underbrace{E_T[(\hat{y} - E_T[\hat{y}])^2]}_{b} + \underbrace{E[\varepsilon \mid x]}_{c}$$

- Pristranost/Bias: sistematska greška na točki *x* - prosjek preko "svih" skupova za učenje T veličine N
- Varijanca: Varijacija greške oko prosječne vrijednosti
- Šum: Greška u određivanju stvarnih vrijednosti y(x)

Dekompozicija prediktivne pogreške: Pristranost i varijanca modela

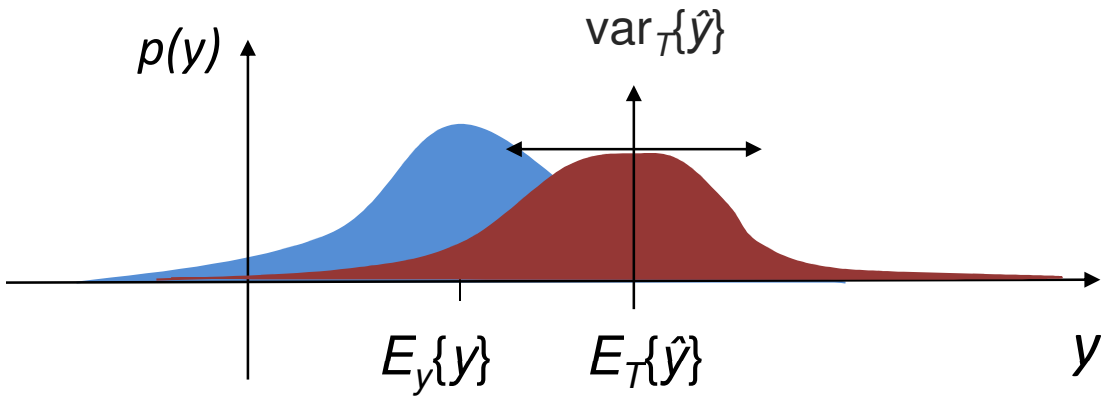


$$(E_y\{y\}-E_T\{\hat{y}\})^2$$

$E_T\{\hat{y}\}$ = prosječni rezultat modela (preko svih T)

bias² = greška između stvarne vrijednosti i prosječnog estimacijskog modela

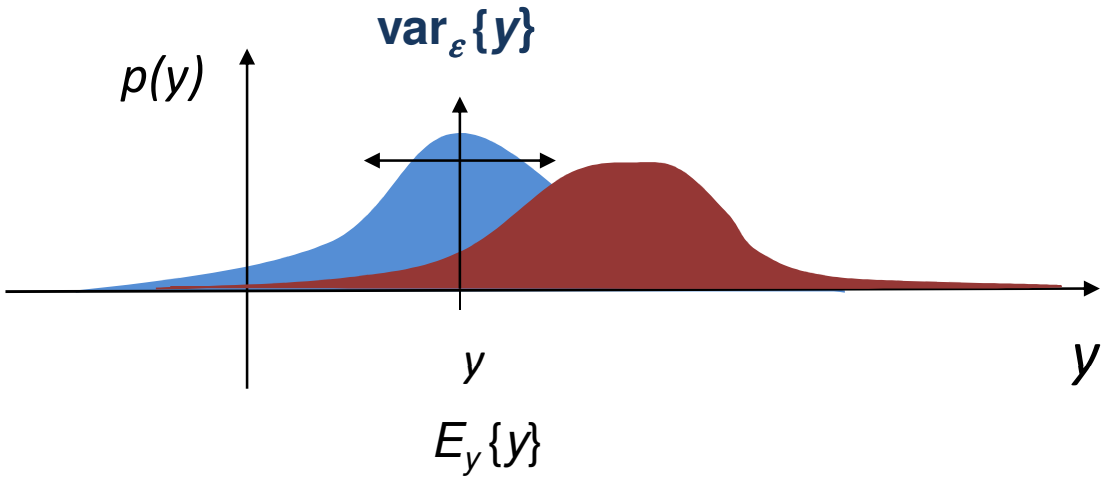
Dekompozicija prediktivne pogreške: Pristranost i varijanca modela



$$\text{var}_T\{y\} = E_T\{(\hat{y} - E_T\{\hat{y}\})^2\}$$

$\text{var}_T\{\hat{y}\}$ = estimacijska varijanca = zbog over-fitinga

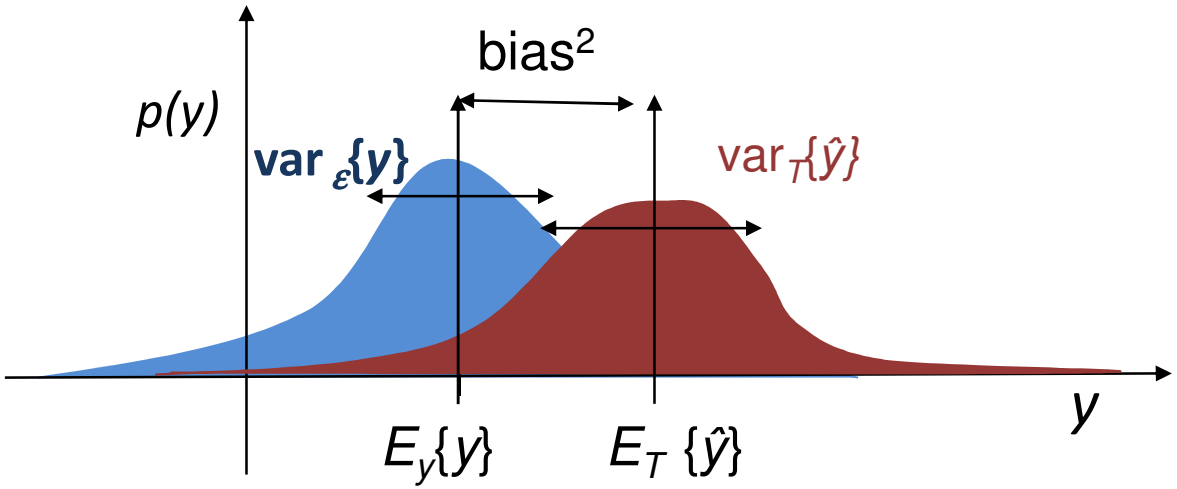
Dekompozicija prediktivne pogreške: Pristranost i varijanca modela



$$\text{var}_\varepsilon\{y\} = E_y\{(y - E_y\{y\})^2\}$$

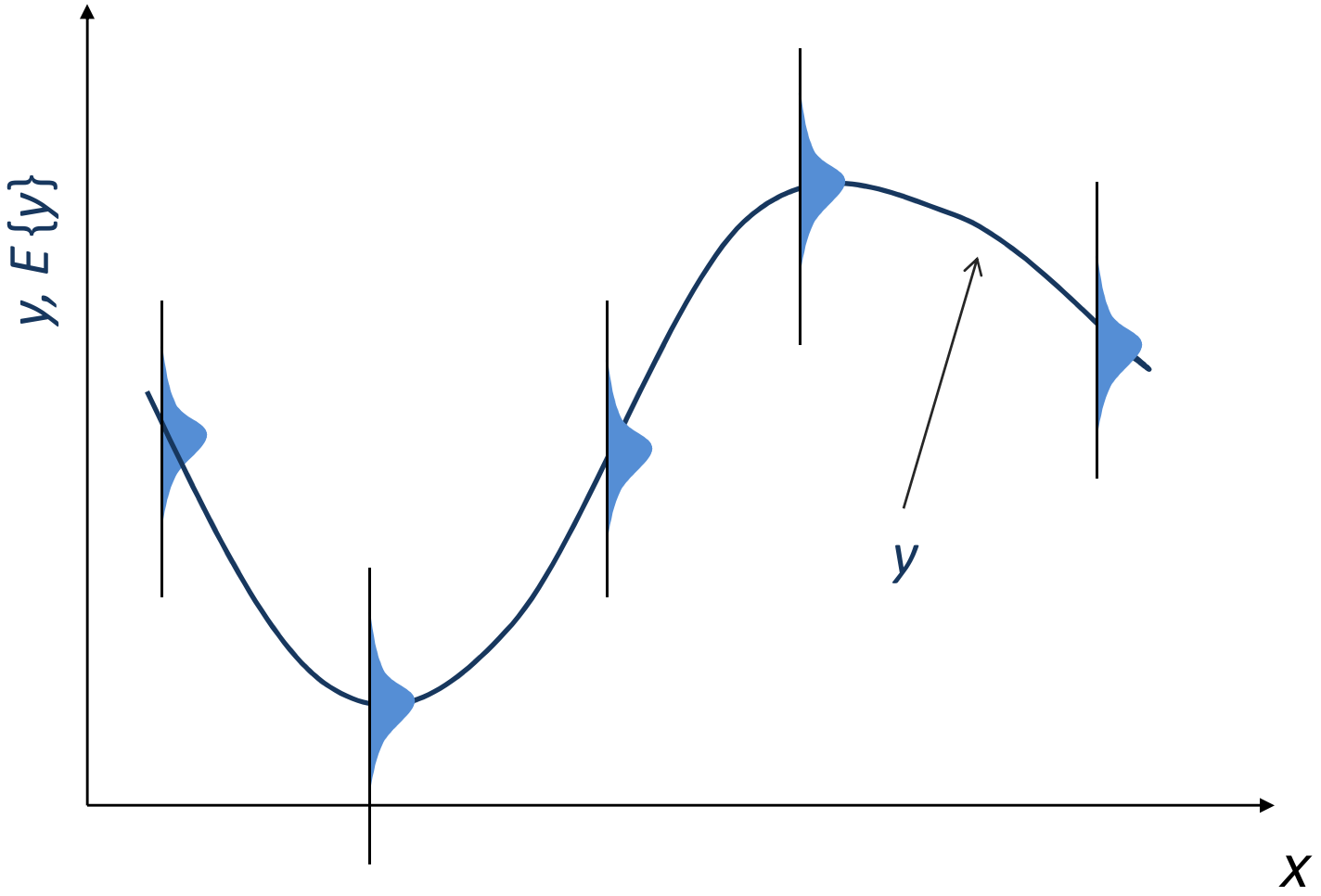
rezidualna greška = minimalna greška koju možemo dostići

Dekompozicija prediktivne pogreške: Pristranost i varijanca modela

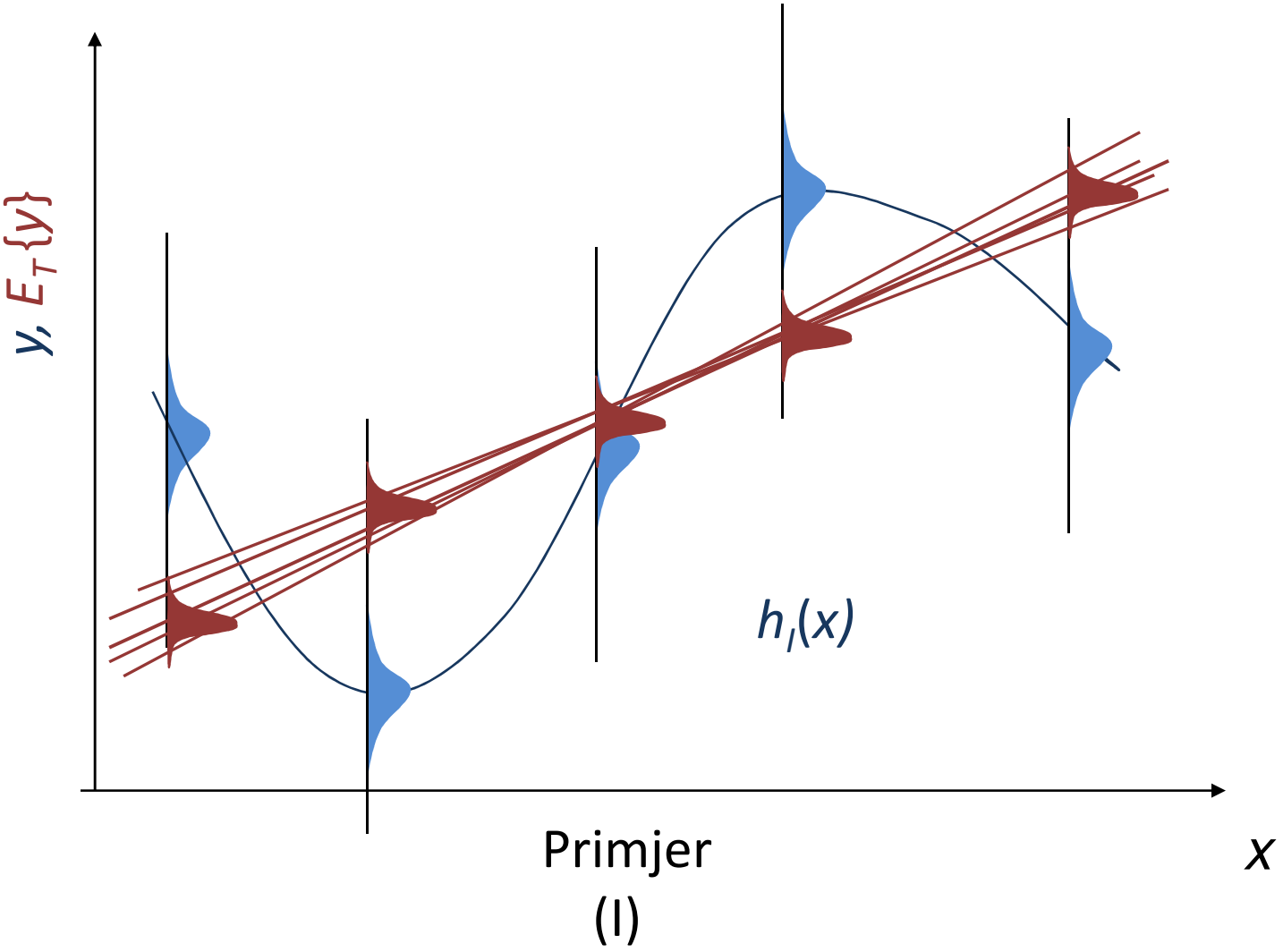


$$E = \text{var}_{\varepsilon}\{y\} + \text{bias}^2 + \text{var}_{T}\{y\}$$

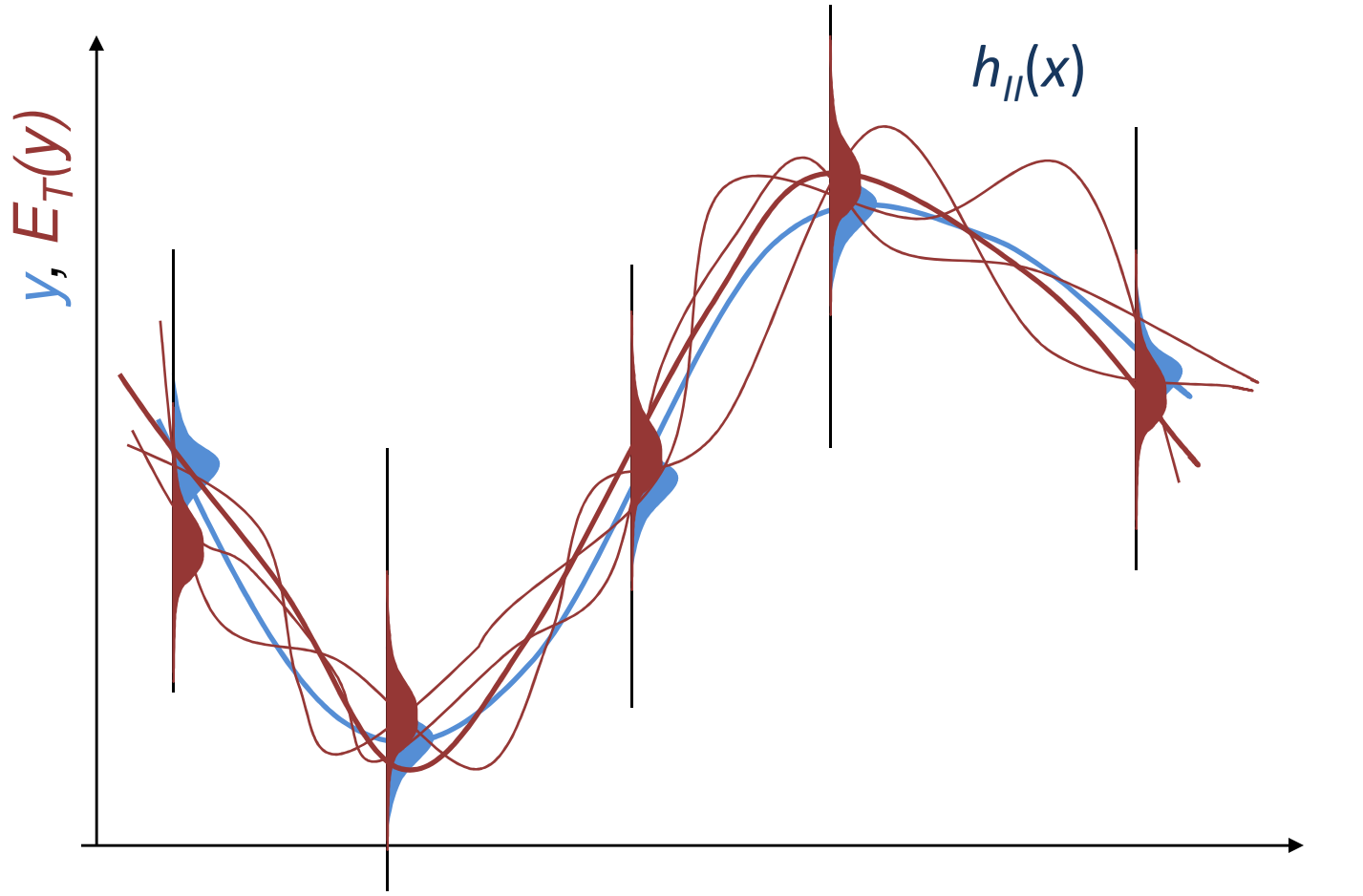
Dekompozicija prediktivne pogreške: Pristranost i varijanca modela



Dekompozicija prediktivne pogreške: Pristranost i varijanca modela

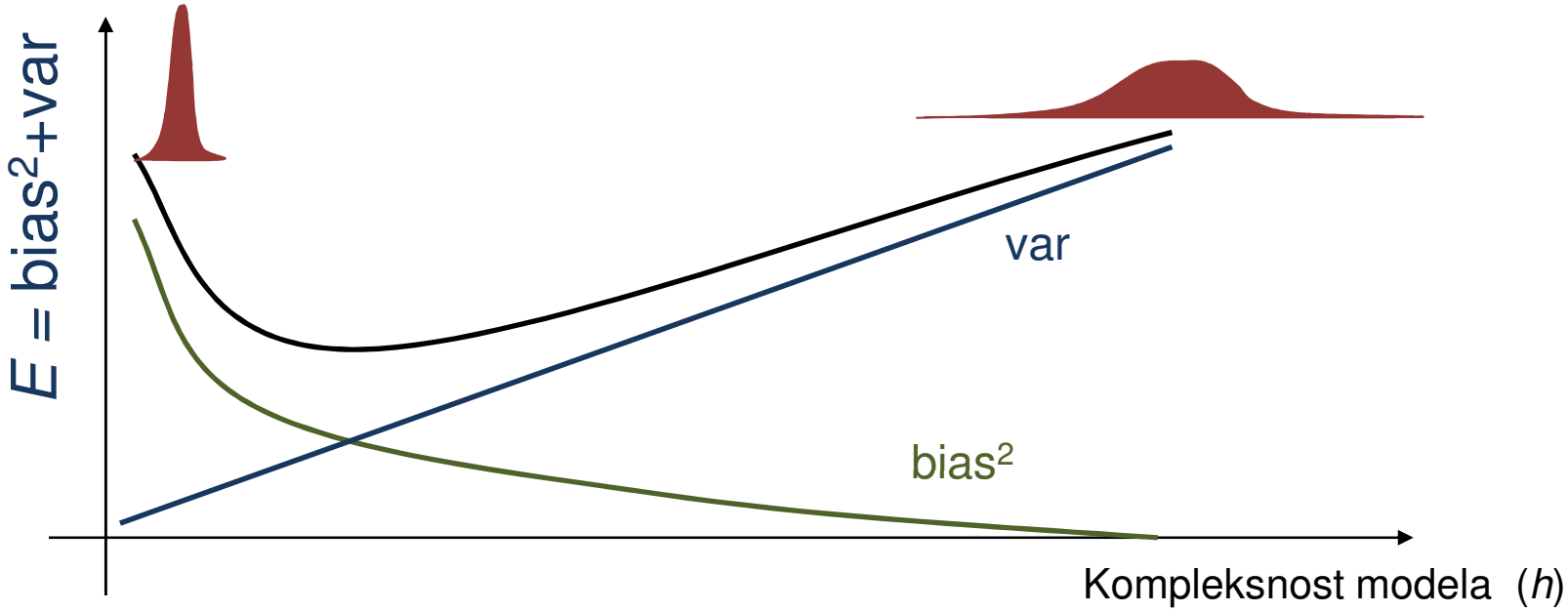


Dekompozicija prediktivne pogreške: Pristranost i varijanca modela



Primjer
II

Dekompozicija prediktivne pogreške: Pristranost i varijanca modela



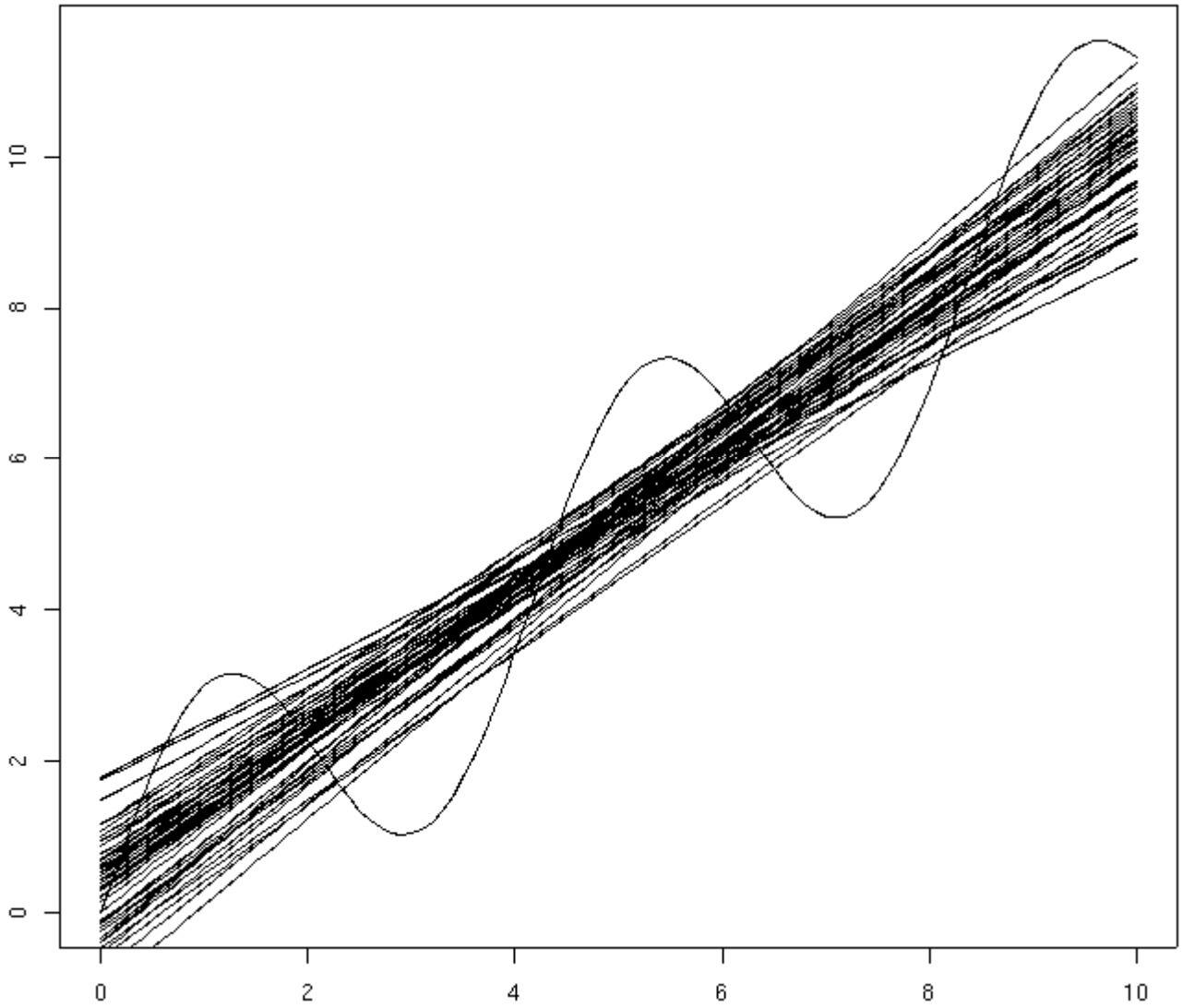
Pristranost (bias) obično pada s povećanjem kompleksnosti modela, dok se varijanca povećava s kompleksnosti modela

Dekompozicija prediktivne pogreške: Pristranost i varijanca modela

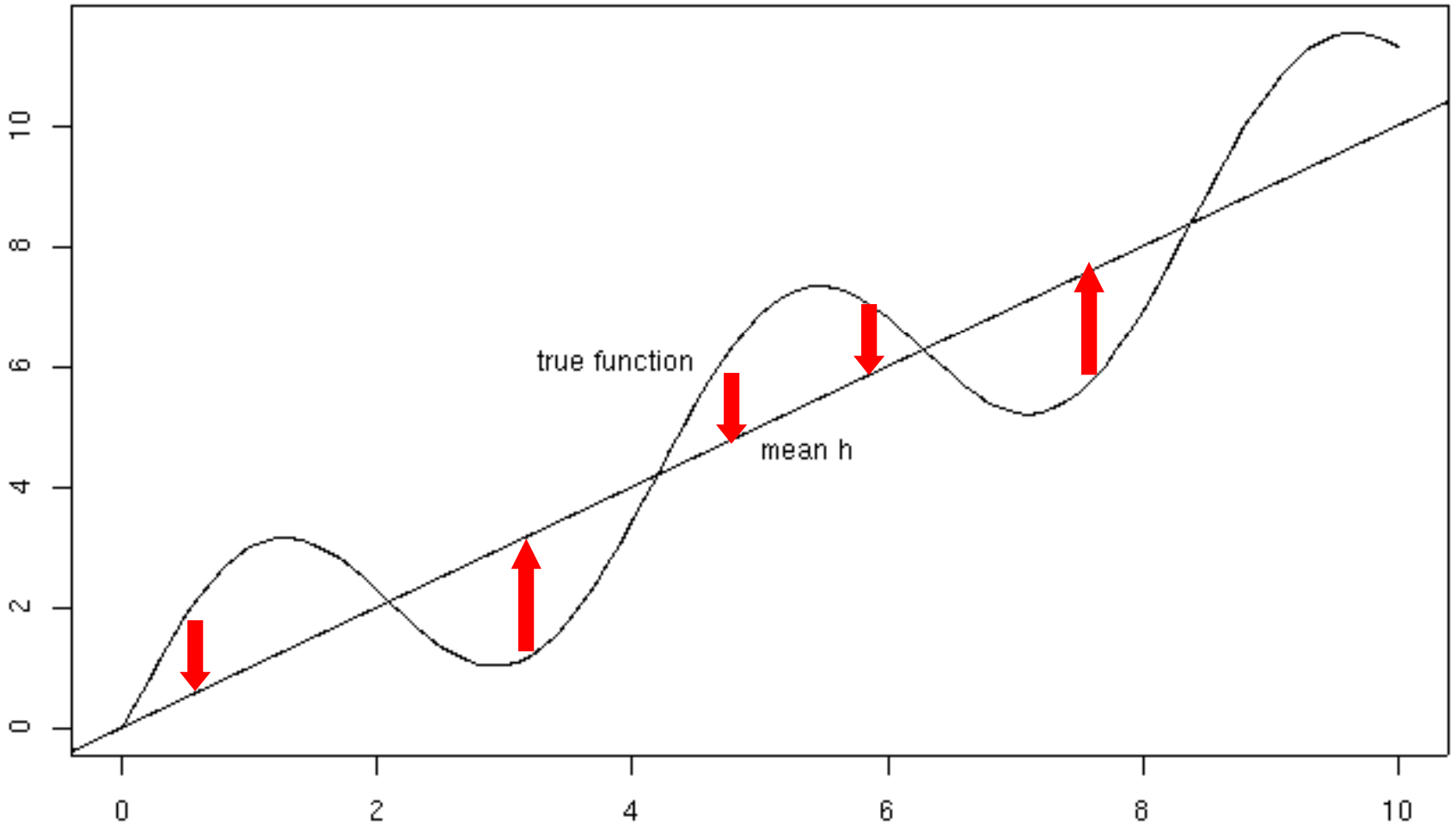
Mjerenje pristranosti (bias) i varijance

- Pristranosti i varijanca – definirani su kao očekivanja !
- Da bi se odredili moramo simulirati velik broj skupova *T*
- Na taj način možemo odrediti i velik broj modela – te ih iskoristiti za određivanje prosječnih vrijednosti

Dekompozicija prediktivne pogreške: Pristranost i varijanca modela

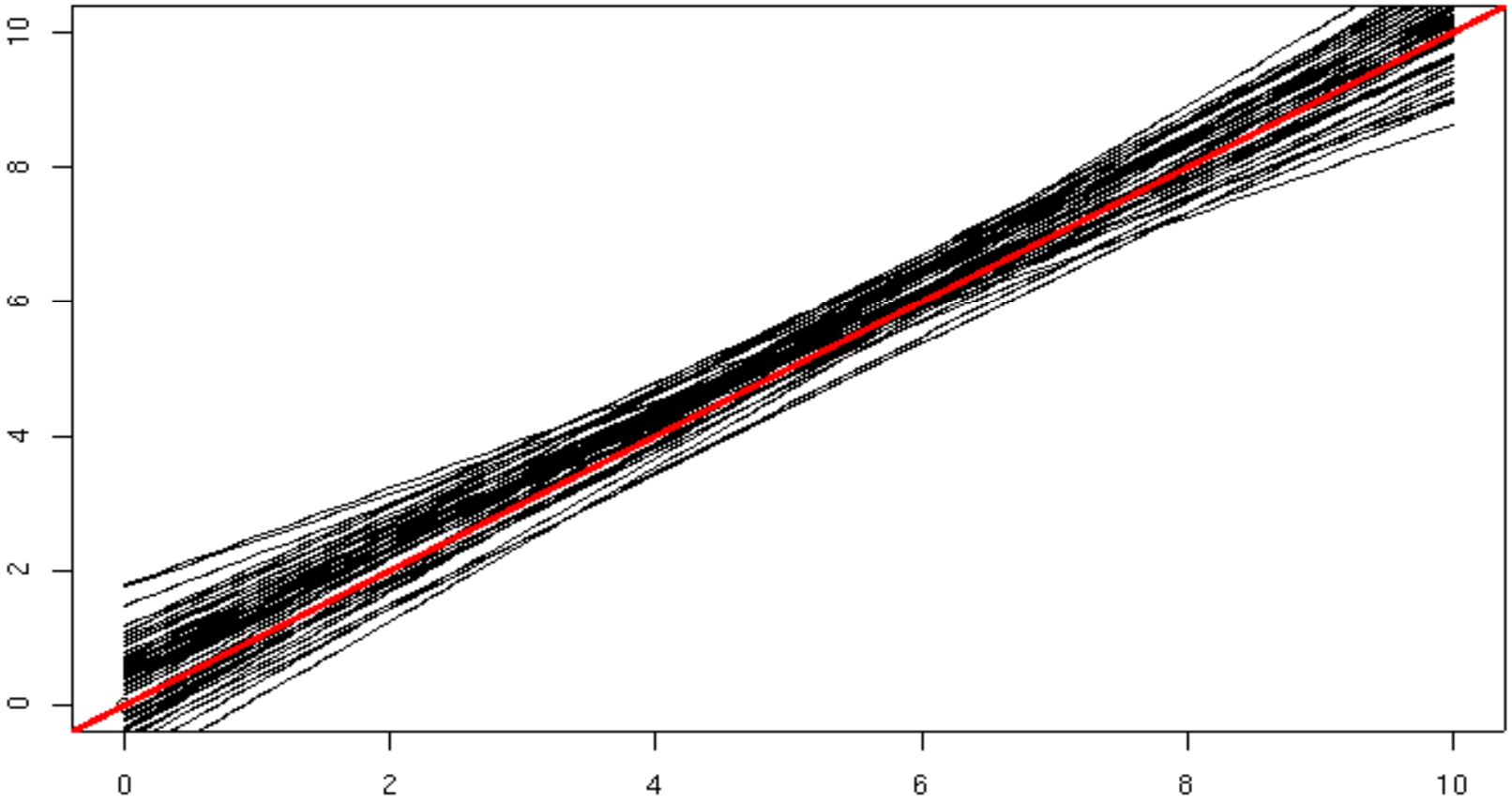


Pristranost modela (Bias)



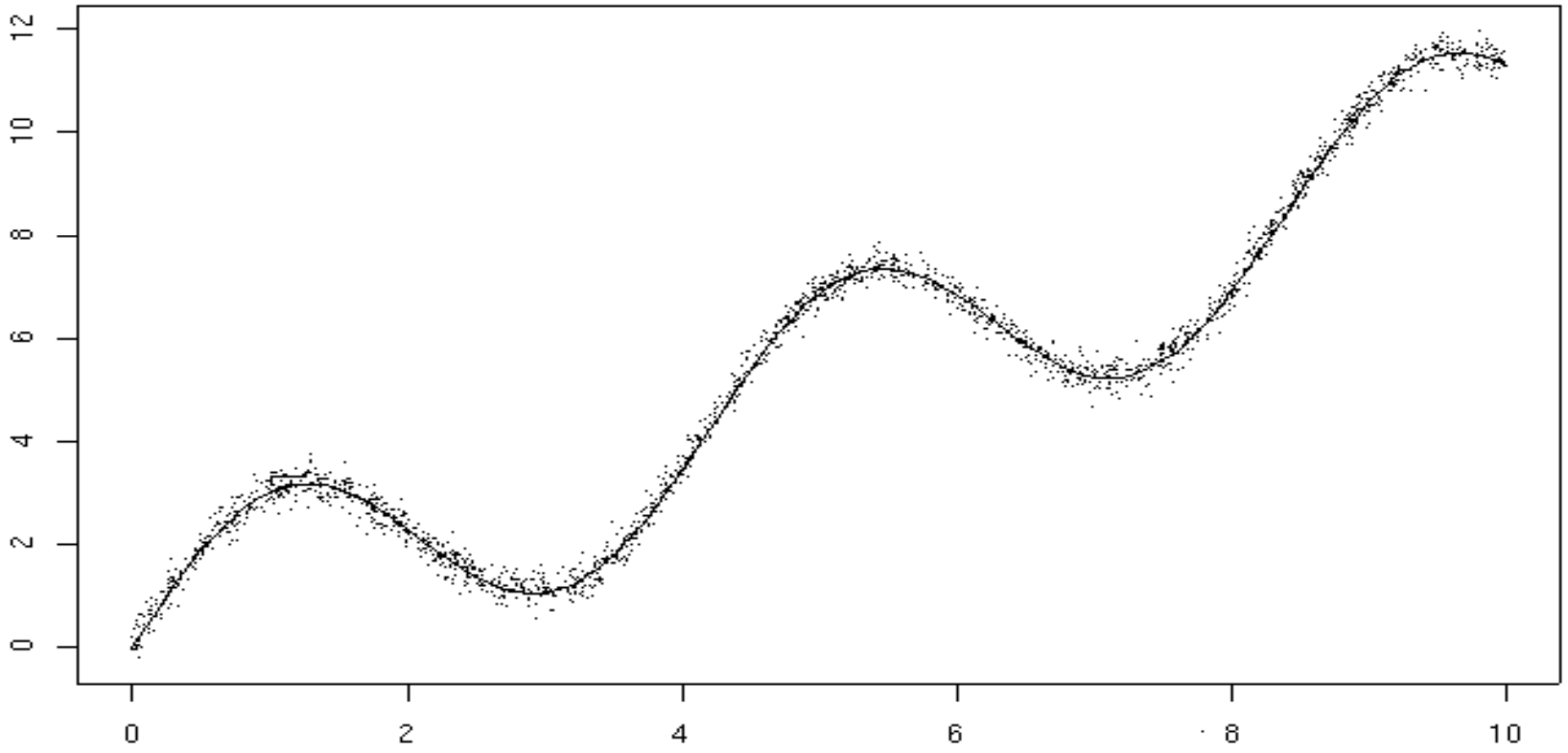
Dekompozicija prediktivne pogreške: Pristranost i varijanca modela

Varijanca



Dekompozicija prediktivne pogreške: Pristranost i varijanca modela

Šum



Bootstrap pristup

1 Uz dani skup podataka *D,* odvojimo (1/3) podataka u skup za testiranje (D_h – hold-out set), a preostale ostavimo u D_t ;

2 Iz skupa D_t *(*veličine *N),* konstruiramo tzv. "bootstrap" repliku skupa - D_b , tako da uzorkujemo *N* primjera iz D_t (uz dozvoljeno višestruko izvlačenje < istog primjera!) ;

3 Algoritmom strojnog učenja konstruiramo model h_b , treniranjem na D_b

4 Korištenjem h_b odredimo predikcije na primjerima iz D_h , te odredimo grešku

5 Ovaj proces se tipično ponovi velik broj puta (K>30)

Određivanje pristranosti(bias) i varijance korištenjem bootstrap uzoraka

1 Za svaki **x** – skup predikcija $h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_K(\mathbf{x})$

Prosječna predikcija:

$$\overline{h}(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K} h_k(\mathbf{x})$$

2 Bias :

$$Bias\ (\mathbf{x}) = y - \overline{h}(\mathbf{x})$$

3 Varijanca:

$$Varijanca\quad (\mathbf{x}) = \frac{1}{K-1}\sum_{k=1}^{K} (h(\mathbf{x}) - \overline{h}(\mathbf{x}))^2$$

Ansambli

- Kombiniranje predikcija više modela koji su napravljeni s istim/različitim algoritmom s ciljem poboljšavanja predikcije u odnosu na jedan model
- Taksonomija – važnije grupe:
  - Tehnike usrednjavanja:
    - "Paralelno/nezavisno" generirani modeli - usrednjena predikcija
    - Bagging, random forests
    - Ovim pristupima smanjuje se primarno varijanca greške
  - Tehnike "boosting" tipa (en. boost - pojačati)
    - "Sekvencijalno" generirani modeli
    - Primjeri: Adaboost, MART
    - Ovim pristupom smanjuje se primarno pristranost (bias)

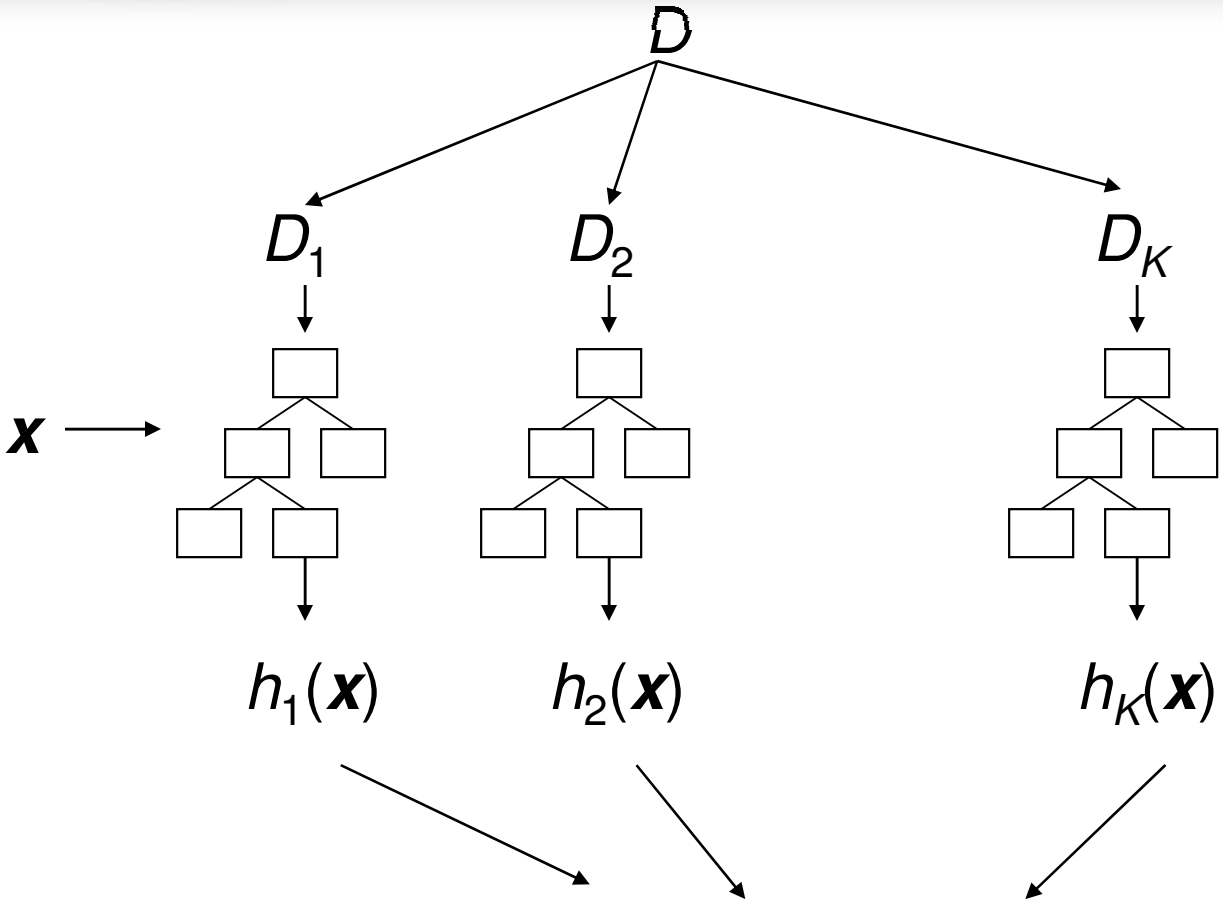
Bagging (Bootsrap aggregating)

- Nastavlja se na ideju bootstraping-a
- Ako imamo velik broj bootstrap modela - zašto ih ne bi sve koristili za jedan zajednički (bagged) model – odnosno predikciju
- Svi modeli “glasaju”:
  - U slučaju klasifikacije – pravilo većine
  - U slučaju regresije – prosjek svih predikcija

Bagging

- Koji modeli (tipovi algoritama) bi najviše dobili baggingom ?
- Baggingom eliminiramo varijancu => dakle kompleksni modeli
- To uključuje stabla odlučivanja – tipično “nestabilni” model, ali i Neuralne mreže ...

Ansambli: bagging

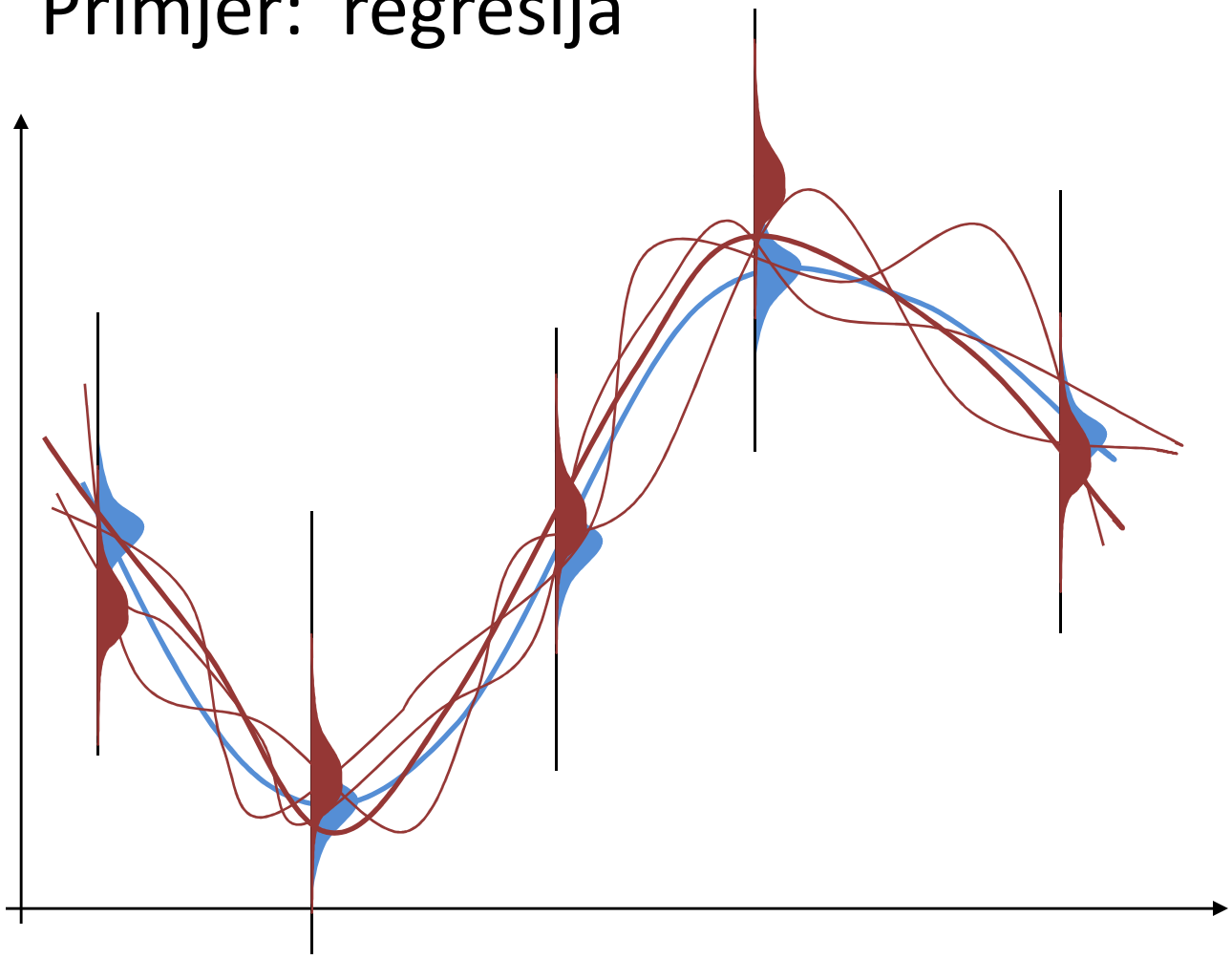


Klasifikacija: $h(\boldsymbol{x})$ = većina u $\{h_1(\boldsymbol{x}),\ldots,h_K(\boldsymbol{x})\}$

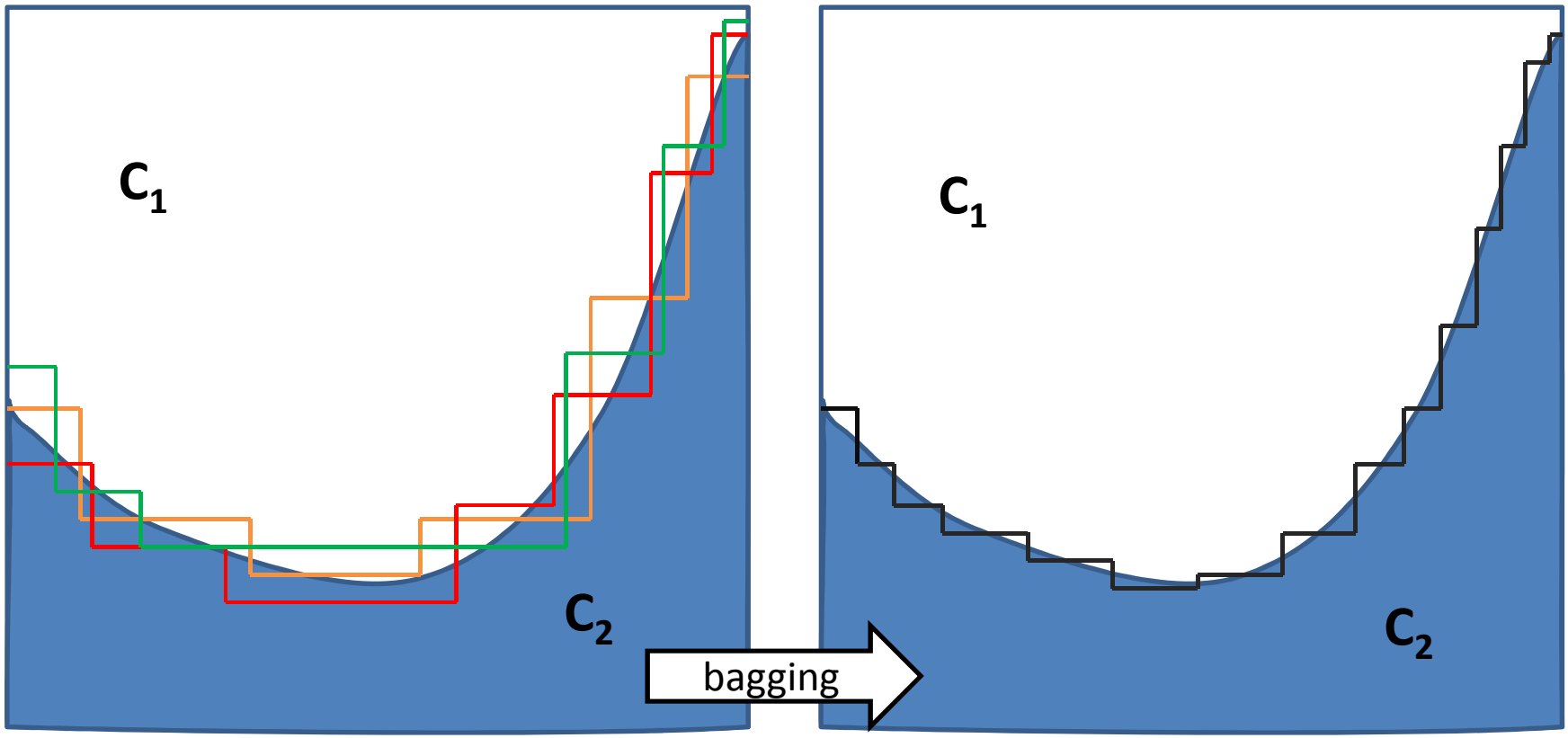
Regresija: $h(\boldsymbol{x}) = 1/K * (h_1(\boldsymbol{x})+h_2(\boldsymbol{x})+\ldots+h_K(\boldsymbol{x}))$

Ansambli: bagging

Primjer: regresija



klasifikacijski primjer: stabla odlučivanja



Random forests

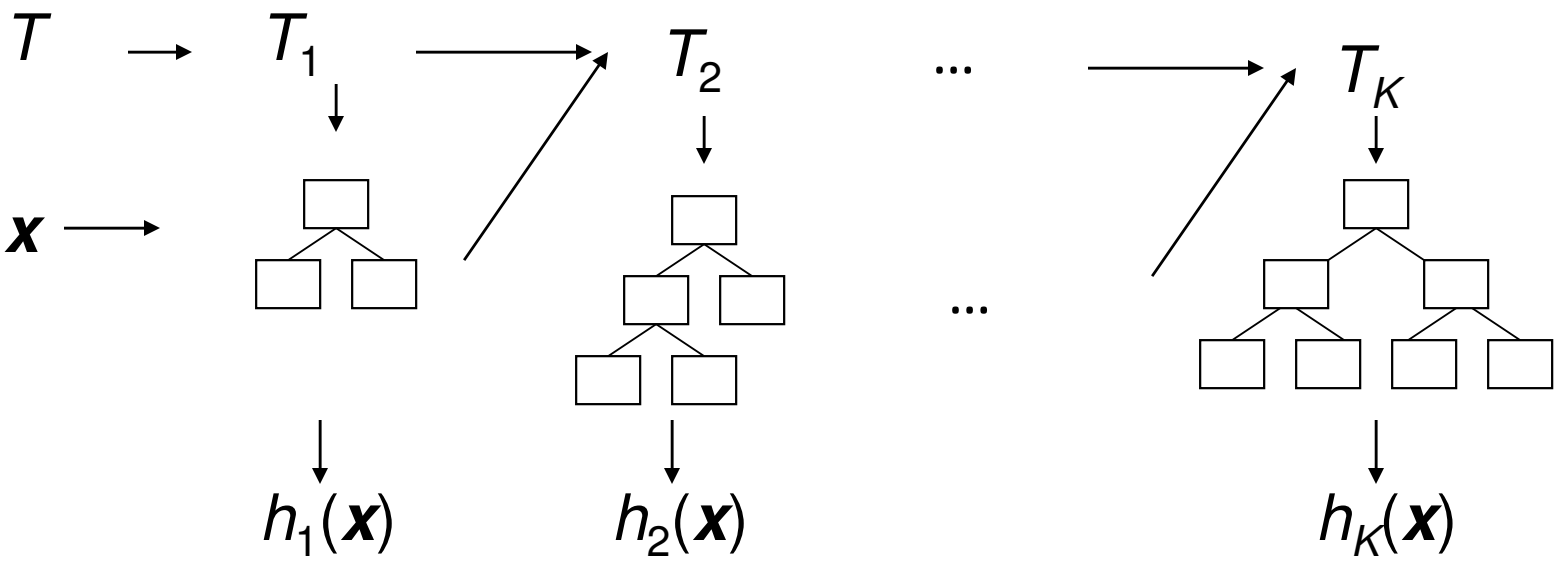
- Kombinira bagging sa slučajnim odabirom podskupa varijabli/atributa (perturbacija modela)
  - Gradi stabla odlučivanja iz bootstrap uzorka skupa za učenje
  - Umjesto izabiranja najboljeg atributa za split – između svih atributa – izabire između *k* slučajno odabranih atributa

  (= bagging , kad je *k* i jednak broju atributa – tipično k=$\sqrt{n}$)
- Balans bias/varijanca korištenjem k:
  - Što je manji k – veća je redukcija varijance, ali je i veći bias

Boosting metode – "jačanje" slabih modela

- Motivacija:
  - kombiniranje outputa "slabih" modela da bi se napravio jači ansambl modela.
- "Slabi" modeli:
  - modeli s visokim bias-om (klasifikacija - malo bolji od slučajne predikcije)
- U odnosu na bagging:
  - Modeli se rade "sekvencijalno" **na modificiranim verzijama podataka**
  - Krajnja predikcija je kombinacija predikcija pojedinačnih modela uz korištenje težinksih faktora

Ansambli: boosting



Klasifikacija: $h(\boldsymbol{x})$ = večina od $\{h_1(\boldsymbol{x}),\ldots,h_K(\boldsymbol{x})\}$
uz težine $\{\beta_1,\beta_2,\ldots,\beta_K\}$

Regresija: $h(\boldsymbol{x}) = \beta_1 h_1(\boldsymbol{x}) + \beta_2 h_2(\boldsymbol{x}) + \ldots + \beta_K h_K(\boldsymbol{x}))$

AdaBoost (Adaptive Boosting) algoritam

- Generira modele tako da sukcesivno mijenja težine primjera u skupu za učenje

- Adaboost povećava težine primjera za koje su prethodni modeli imali loše predikcije – dakle fokusira učenje na “teške” slučajeve

- Na kraju: glasanje s težinskom-većinom; točniji modeli imaju veći utjecaj u glasanju

AdaBoost algoritam

Algorithm 1: Adaboost Algorithm (Freund and Schapire)

Input : A weak learning algorithm $WeakLearn$, an integer T specifying number of iterations, and N labelled training data $\{(x_1, y_1), \ldots, (x_N, y_N)\}$.

Output : A strong classifier F.

Initialize the weight vector $w_i^1 = \frac{1}{N}$, *for* $i = 1, \ldots, N$.

for $t \leftarrow 1, 2, \ldots, T$ **do**

1. $\mathbf{p}^t \leftarrow \mathbf{w}^t / \sum_{i=1}^{N} w_i^t$.
2. Call $WeakLearn$, providing it with the distribution on $\mathbf{p}^t$; get back a weak learner $h_t : X \rightarrow \pm 1$.
3. Calculate the weight error of h_t: $\epsilon_t = \sum_{i=1}^{N} p_i^t \frac{1}{2} |h_t(x_i) - y_i|$.
4. $\alpha_t \leftarrow \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
5. $w_i^{t+1} \leftarrow w_i^t \exp\left(\alpha_t \frac{1}{2} |h_t(x_i) - y_i|\right)$, for $i = 1, 2, \ldots, T$.

Output the final strong classifier:

$$F(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^{T} \alpha_t h_t(x) \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

AdaBoost algoritam

Ulaz: **"**Slabi" algoritam za učenje ***L*** *(*algoritam s visokim high biasom*)*
T – broj iteracija, *N* – broj primjera za učenje

Izlaz: "Ojačani " (boosted) klasifikator ***F*** (algoritam s visokim niskim biasom)

Inicijaliziraj težine primjera: $w_i^1 = 1/N, \text{ za } i = 1,...,N$

Za *t <- 1,2,....,T* **radi**

1. $\mathbf{p}^t = \mathbf{w}^t / \sum_{i=1}^{N} w_i^t$
2. Pozovi ***L*** **(p^t,X)** => rezultat je "slabi" model (***h_t : X -> Y***)
3. Odredi grešku h_t: $\varepsilon_t = \sum_{i=1}^{t} p_i^t \left| h_t(x_i) - y_i \right|$
4. Odredi $\alpha^t = \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
5. Odredi nove težine primjera $w_i^{t+1} = w_i^t \exp(\alpha_t \left| h_t(x_i) - y_i \right|), \text{ za } i = 1,2,...,N$

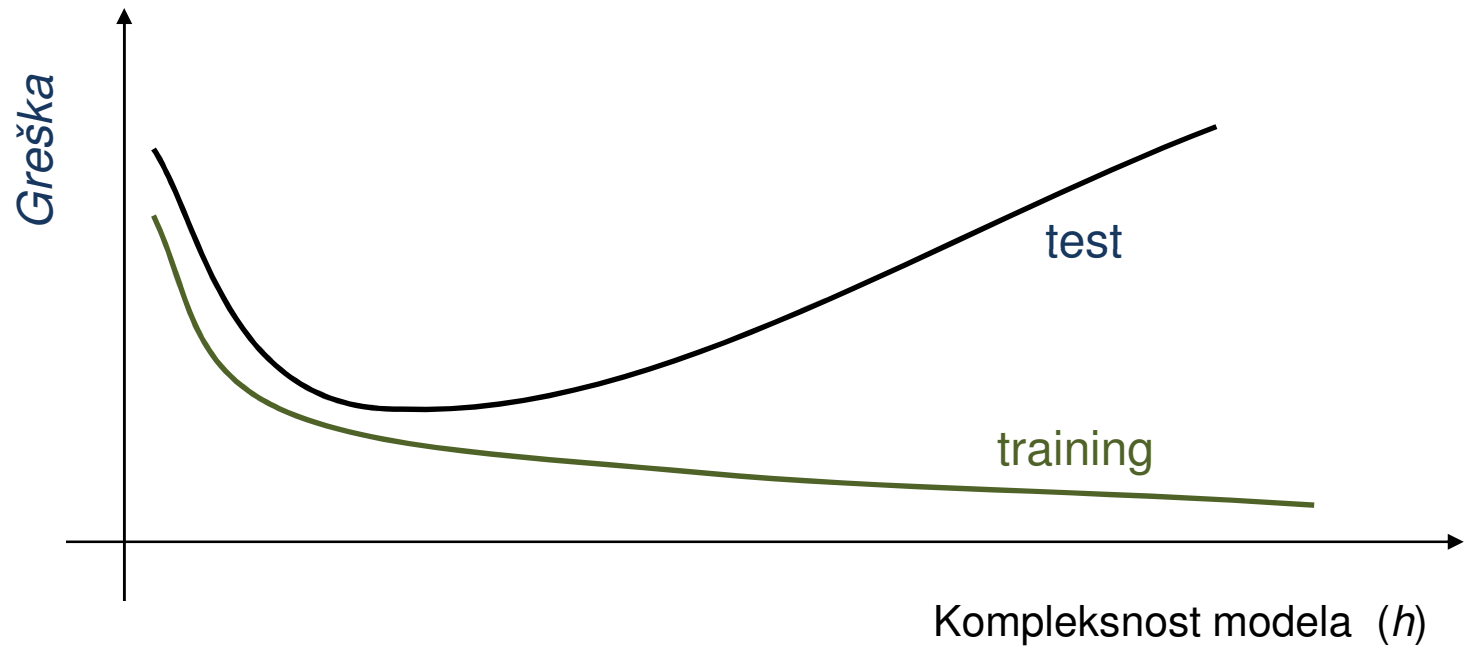
Vrati klasifikator ***F:***

$$F(x) = \begin{cases} 1, & \text{ako je } \sum_{i=1}^{T} \alpha_t h_t(x) \geq 0, \\ -1, & \text{inace} \end{cases}$$

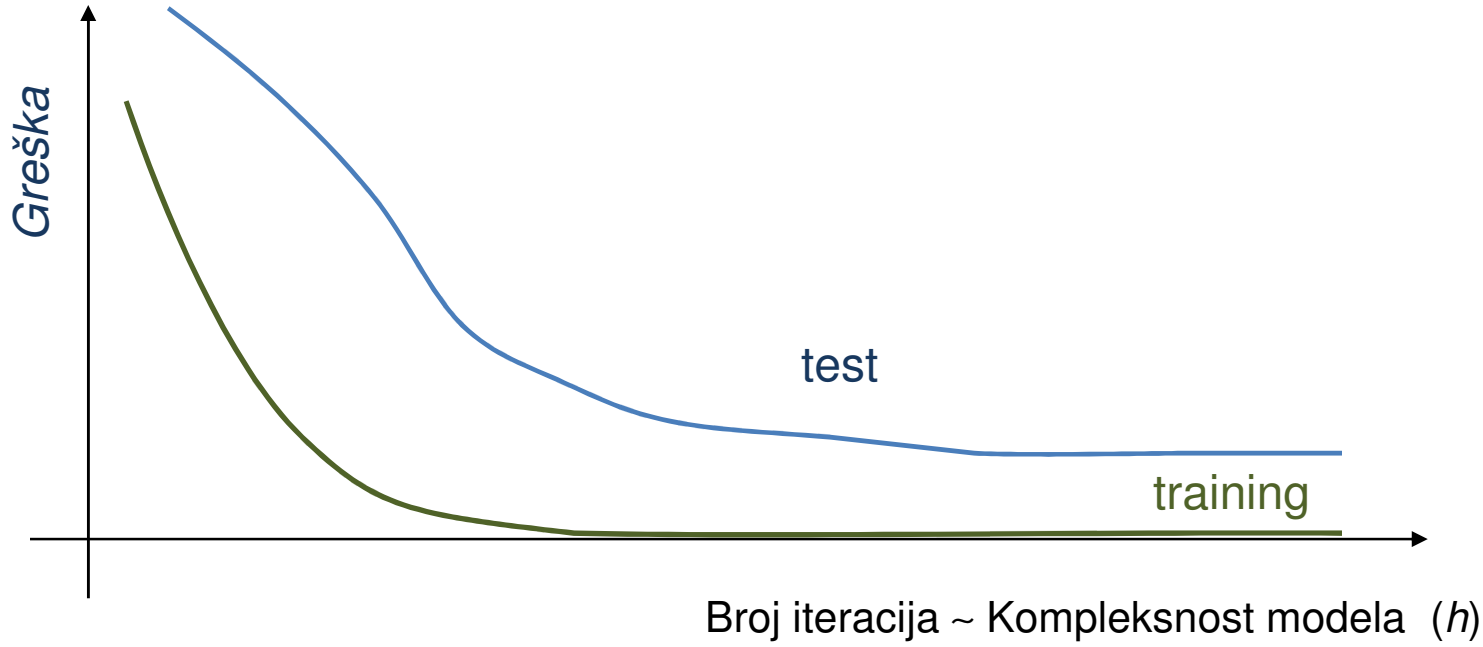
Zašto boosting radi dobro ?

- Kombinira modele koji imaju visoki bias, tako da se dobiju kompleksniji/ekspresivniji klasifikatori
- Boosting => redukcija pristranosti (bias-a)
- Što se dešava ako imamo velik broj iteracija (K) ? Dobit ćemo vrlo složeni model...a greška na novim primjerima ?

Algoritmi strojnog učenja - tipični slučaj



Boosting – tipični slučaj !?



Objašnjenje

- Modeli koji su generirani boosting-om:
  - h(**x**) = sign(f(**x**))
  - Klasifikacija primjera je korektna ako je sign(f(**x**))=y
  - Margina primjera:

    m(f(**x**),y)=y*f(**x**)
  - Što je margina veća na skupu za učenje – za očekivati je i manju grešku i na testnom skupu (generalizacijska greška)
  - Boosting algoritmi rade na povećavanju ove margine
  - Intuitivno – povećanje margine je slično smanjenju varijance modela

ECOC: **E**rror **C**orrecting **O**utput **C**oding

- Reformuliranje originalnog problema
- Tipično: za probleme s više klasa (K>>2)

Učenje

Za i=1,M

a) Slučajno particioniranje K klasa u dva različita podskupa $\{A, B\}_i$

b) Re-labeliranje primjera u dvije nove klase $\{A, B\}_i$

c) Učenje h_i za klasifikaciju primjera $\{A, B\}_i$

d) Ponavljaj

Klasifikacija novog primera

a) Ako je $h_i(\mathbf{x})=A_i$, tada sve originalne klase u A_i dobijaju glas; odnosno ukoliko je $h_i(\mathbf{x})=B_i$ sve originalne klase u B_i dobijaju glas

b) Konačno, klasa s najviše dobivenih glasova je predikcija ECOC ansambla

Stacking; Stacked generalization

(? Stog modela; generalizacija preko stoga modela?)

Učenje meta-modela nad predikcijama baznih modela

Učenje se odvija u dva nivoa:

1 Razdvoji skup za učenje *T* na dva dijela T_I *i* T_{II} *(slučajno* stratificirano uzorkovanje)
Na prvom dijelu T_I se "uči" nekoliko baznih algoritama - što različitijih

2 Nakon što su naučeni modeli baznih algoritama na T_I , ti se modeli iskoriste za predikcije na T_{II}
Novi algoritam uči kombinirati predikcije modela (meta-model) na T_{II}

Klasifikacija novog primjera

1 Bazni modeli prvo daju svoje predikcije
2 Meta-model koristi ove predikcije da bi napravio konačnu predikciju ansambla

Ansambli: sažetak

- Metode bazirane na kombiniranju više modela u jednu predikciju
- Poboljšavaju točnost u odnosu na individualne modele, jer reduciraju ili varijancu ili bias (ili oboje !)
- Bagging – redukcija "varijance"; efikasna za nestabilne, kompleksnije modele/hipoteze
  - "paralelno" stvaranje modela
  - Osnova su: repetitivno (bootstrap) uzorkovanje i usrednjavanje predikcija (regresija), odnosno većinsko glasanje (klasifikacija)
- Boosting – redukcija pristranosti (bias-a), ali i povećanje margine
  - "sekvencijalno" stvaranje modela
  - fokus na teže dijelove/primjere; daje težinu pojedinim modelima prema njihovoj točnosti

Ansambli: sažetak

- S obzirom da zahtijevaju učenje većeg broja modela
  - vremenski su i memorijski zahtjevne metode

- Gotovo na svim realnim problemima, kod kojih je važna preditkivna točnost - najbolje rezultate postižu ansambli