Vytěžování Dat
Přednáška 12 – Kombinování modelů

Miroslav Čepek
Pavel Kordík a Jan Černý (FIT)

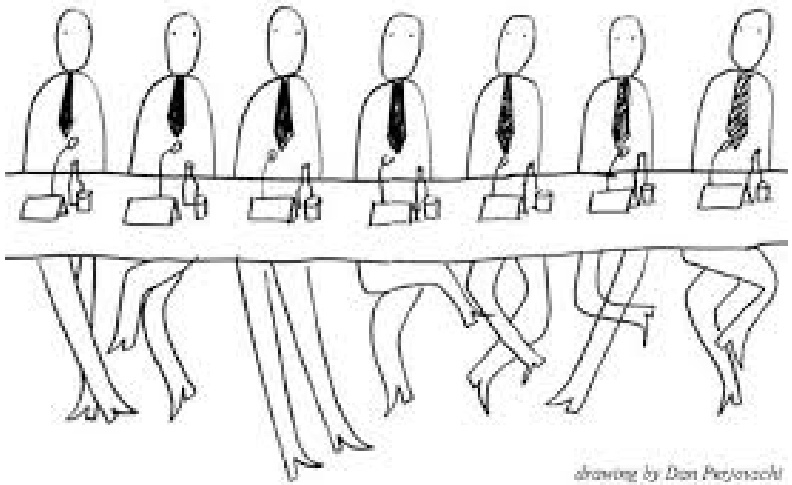
Fakulta Elektrotechnická, ČVUT

Evropský sociální fond Praha & EU:
Investujeme do vaší budoucnosti

16.12.2014

Ensembles (Kombinování modelů)

- Aneb víc hlav, víc ví!
- Co když jsme se dostali na hranice možností jednoho modelu?
- Co když už další učení nebo složitější model vede jen k přeučení?
- Co s tím?



Motivace z praxe

- Soutěž – Netflix prize (predikce, které filmy by se zákazníkovi mohly také líbit, když viděl a nějak ohodnotil jinou skupinu filmů?)
- Respektive – cílem je předpovědět hodnocení filmu konkrétním zákazníkem, když víme, jak hodnotil jiné filmy v minulosti.
- Cenu $1,000,000 získá nejlepší, kdo současně dokáže na testovací množině zlepšit stávající přesnost alespoň o 5%.

Motivace z praxe (2)

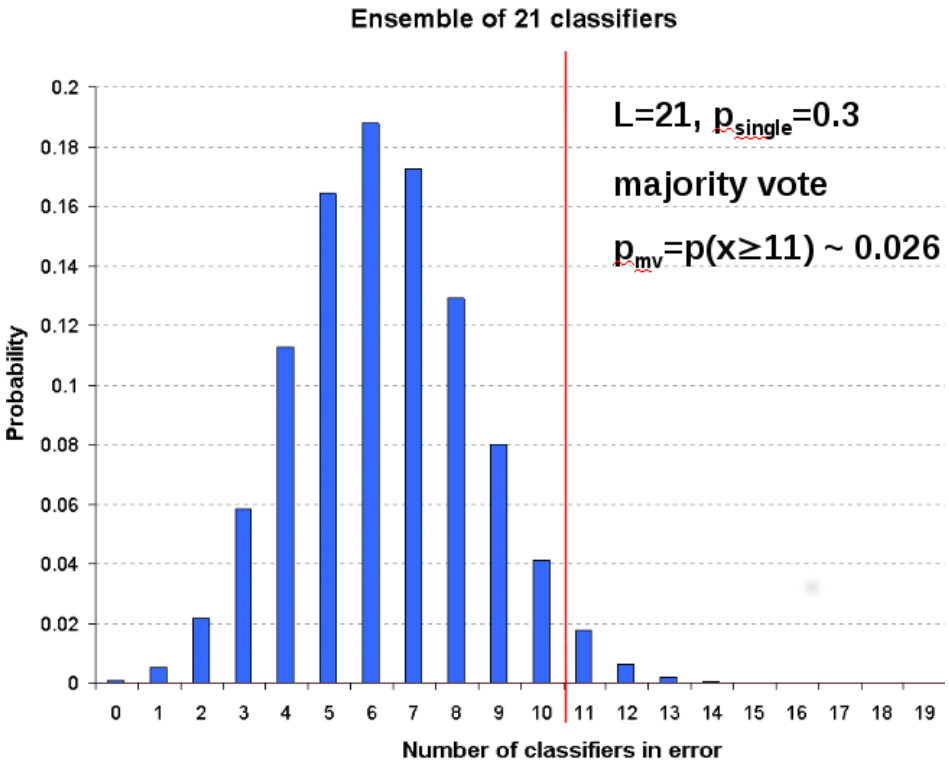
Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11

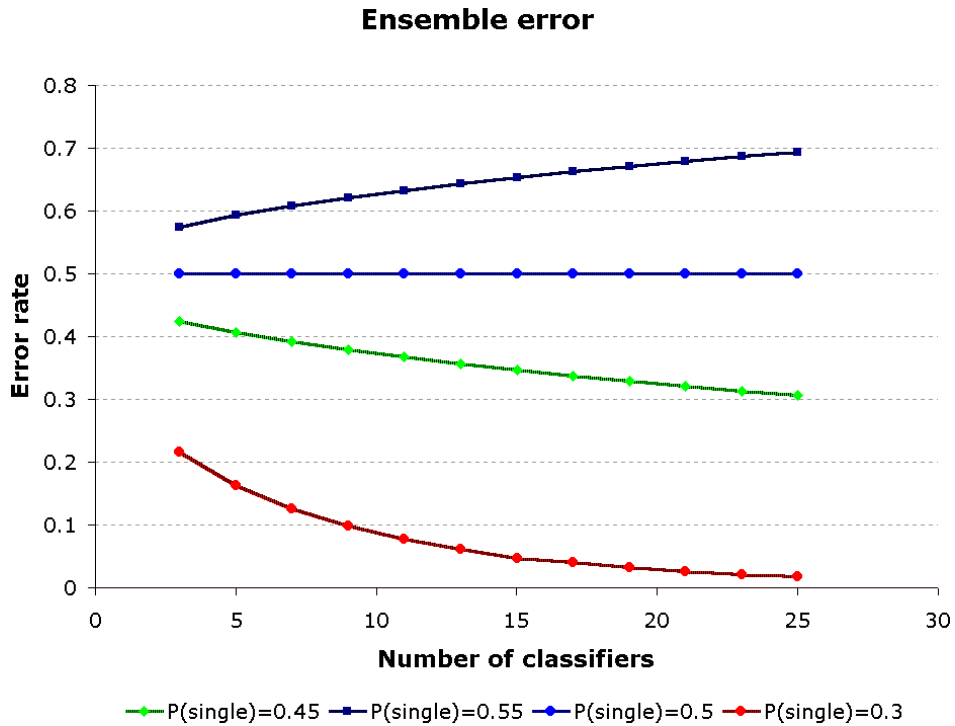
- A všechny TOP týmy používají kombinace modelů.

Motivace pro kombinaci modelů

- Cesta k vyšší přesnosti než u nejlepšího z dílčích modelů.
- Každý model dělá chyby pro trochu jiná data. A trochu jiné chyby.
  - Když se zkombinují dohromady, možná své chyby eliminují.
- Řízený experiment
  - dvě třídy s $p_{c1} = p_{c_2}$, modely mají pevný chybový poměr p_{single},
  - každý dílčí klasifikátor chybuje **nezávisle** na ostatních,
  - výsledná klasifikace dána majoritním neváženým hlasováním,
  - jak to dopadne (přesnost jako funkce počtu modelů)?

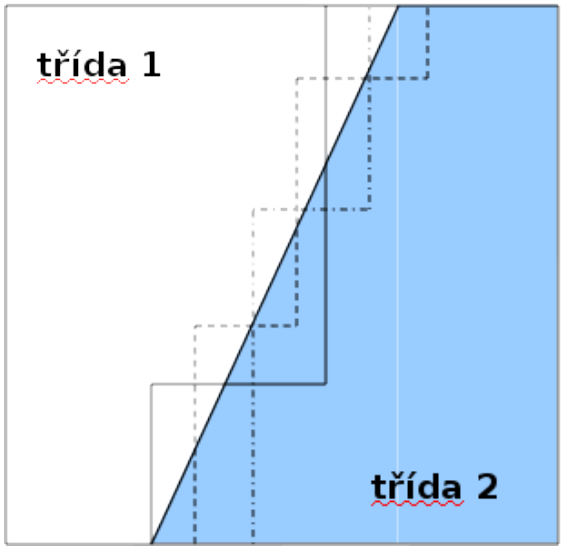
Motivace pro kombinaci modelů (2)

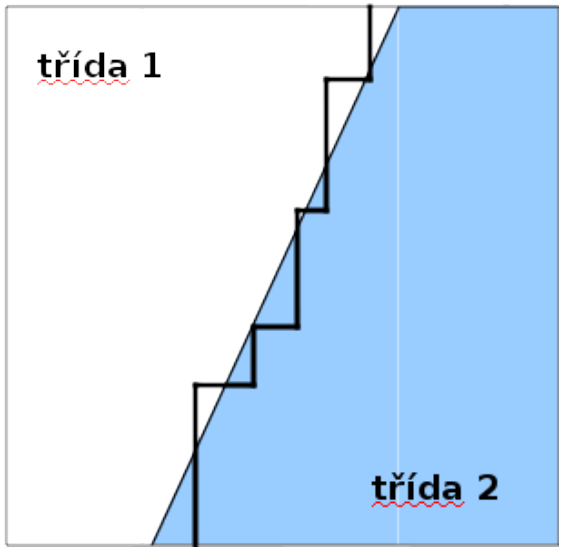


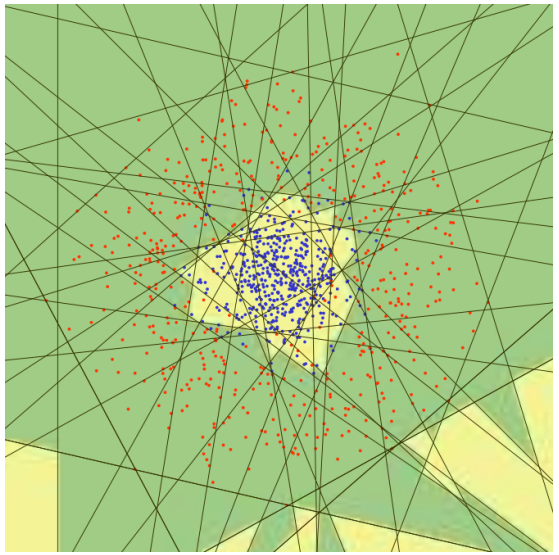


Motivace pro kombinaci modelů (3)

- Analogie z idealizované zkoušky
  - každý student se naučí látku (každý s jinými chybami),
  - společně rozhodují o odpovědích na otázky,
  - v ideálním případě by měli u každé otázky odpovědět správně :).
- Kombinací různých modelů dokážeme aproximovat rozhodovací hranici, kterou by samostatné modely proložit nedokázaly.





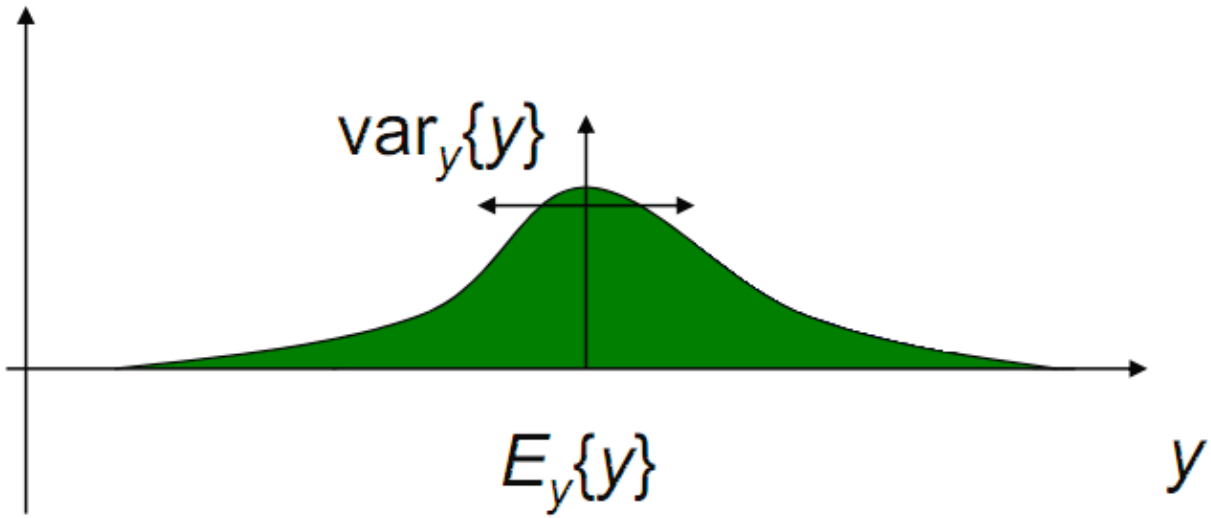


Chyby modelů

- Vraťme se ještě k chybám modelů z minulé přednášky.
- Když vytvořím různé modely, nebudou dělat identické chyby.
  - Například použiji jiné počáteční parametry,
  - nebo vezmu jinou podmnožinu trénovací množiny.
- U kombinací modelů
  - Nebudu usilovat o maximální přesnost dílčích = slabých modelů,
  - Budu spoléhat na filtrování chyb při společném rozhodování.

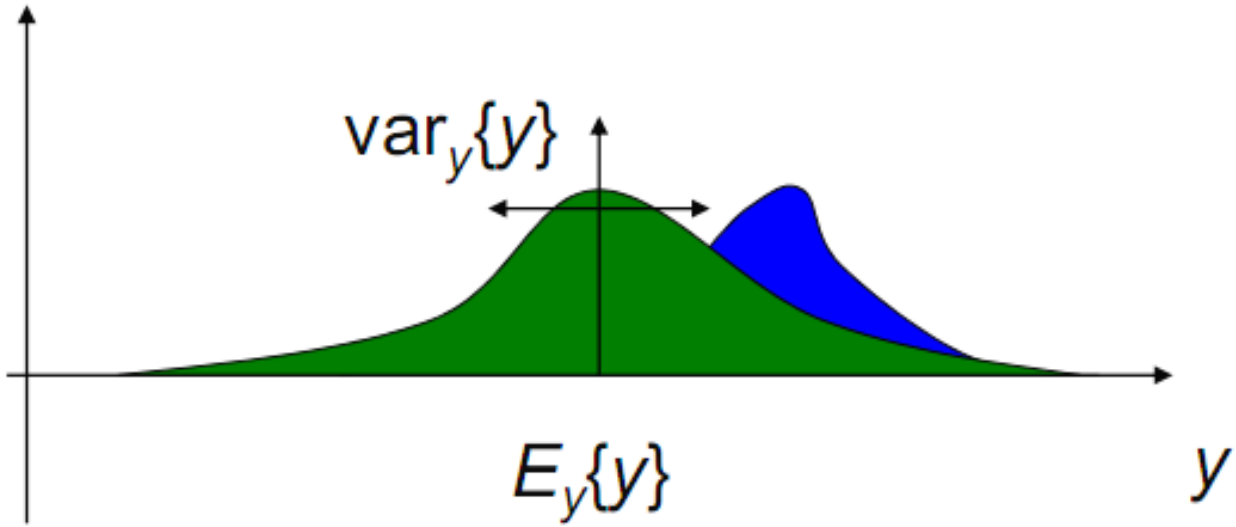
Variance modelů (rozptyl chyby modelů)

- Nicméně chyby všech takto vytvořených modelů by měly být z normálního rozdělení se stejnou střední hodnotou a rozptylem.
- Čili rozptyl modelů udává jak moc se liší chyba jednotlivých modelů od střední chyby.



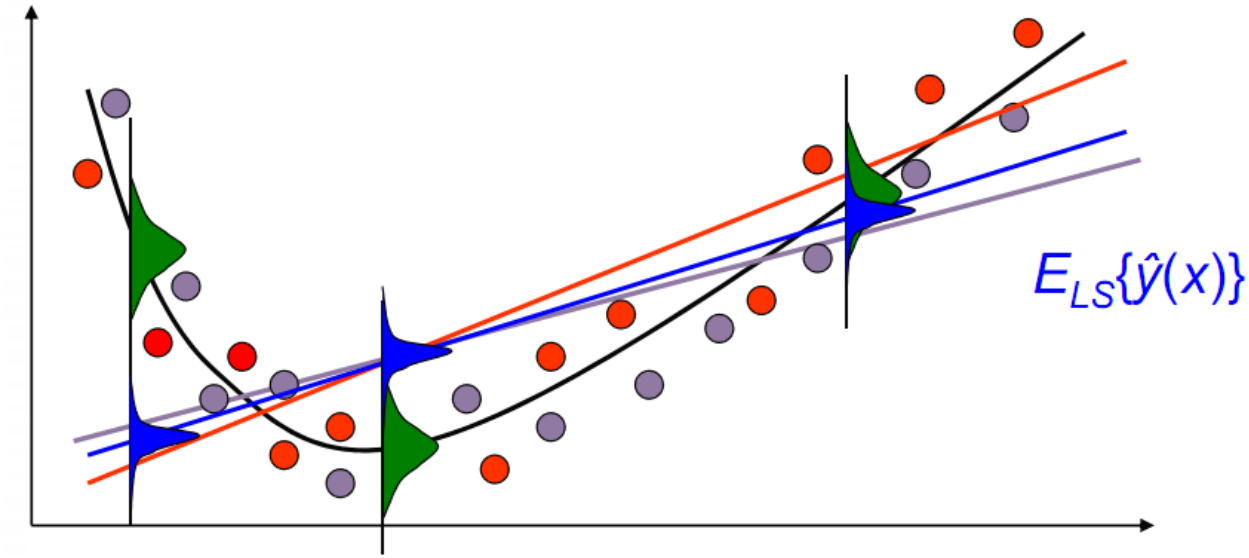
Bias modelů (Zaujetí modelů)

- Bias (zaujetí) vyjadřuje systematickou chybu způsobenou (například) špatně zvolenou trénovací množinou.



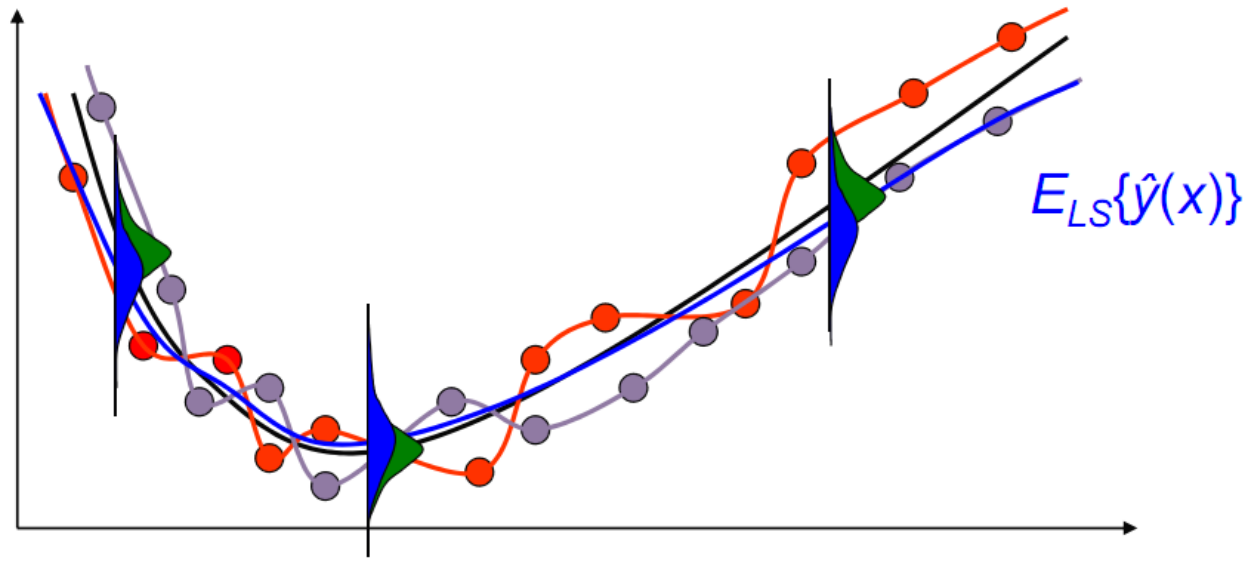
Nedoučení modelu (underfitting)

- Model (například lineární regrese) je příliš jednoduchý, aby dokázal popsat data.
- Modely budou mít nízkou varianci, ale vysoký bias.
- Co to znamená?
- Modely si budou podobné, ale mají (velkou) chybu už na trénovacích datech.



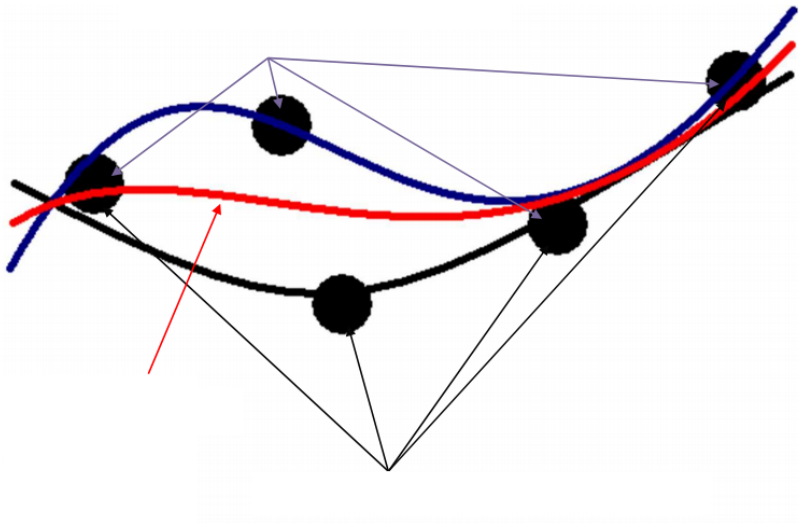
Přeučení modelu (overfitting, už zase)

- Model je příliš ohebný a naučil se i šum, tj. vztahy, které v datech ve skutečnosti nejsou.
- Modely budou mít nízký bias, ale vysokou varianci.
- Co to znamená?
- Jednotlivé modely budou mít nízkou chybu na trénovacích datech, ale budou hodně rozdílné a budou výrazně více chybovat na testovacích datech.



Kombinování modelů – prevence přeučení

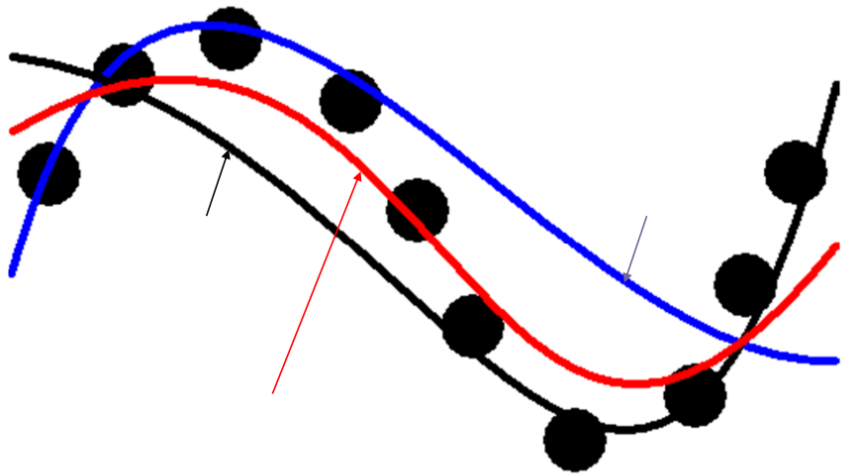
- Důsledek přeučení
  - model na testovacích datech vykazuje velké chyby.
- Mám skupinu modelů naučených na různých podmnožinách trénovacích dat.
- Každý model jsem možná přeučil.
- Můžu něčeho dosáhnout kombinací těchto modelů?



- Snížili jsme rozptyl modelů.

Kombinování modelů – snižování zaujetí

- Jednoduché modelovací metody s malou ohebností (opět naučené na různých podmnožinách dat) nedokáží dobře aproximovat rozhodovací hranici.



- Kombinací opět dosáhnu ”ohebnější” hranice a tedy i menší chyby.

Skupiny modelů

- Síla seskupování modelů tkví v diverzitě (různorodosti) modelů.
- Diverzity můžu dosáhnout dvěma cestami:
  - Použít různé modelovací techniky.
  - Vytvořit různé podmnožiny trénovací množiny (instance, příznaky).
- Základní schéma použití algoritmů pro kombinaci modelů:
  1. Vyber stavební jednotky ensemblu (vhodné modely).
  2. Vytvoř pro každý trénovací množinu.
  3. Natrénuj všechny modely v ensemblu (učení jednotlivých modelů může být závislé na učení ostatních modelů v ensemblu).
  4. Spočítej výstup všech modelů v ensemblu.
  5. Jejich výstup zkombinuj do výsledného výstupu.

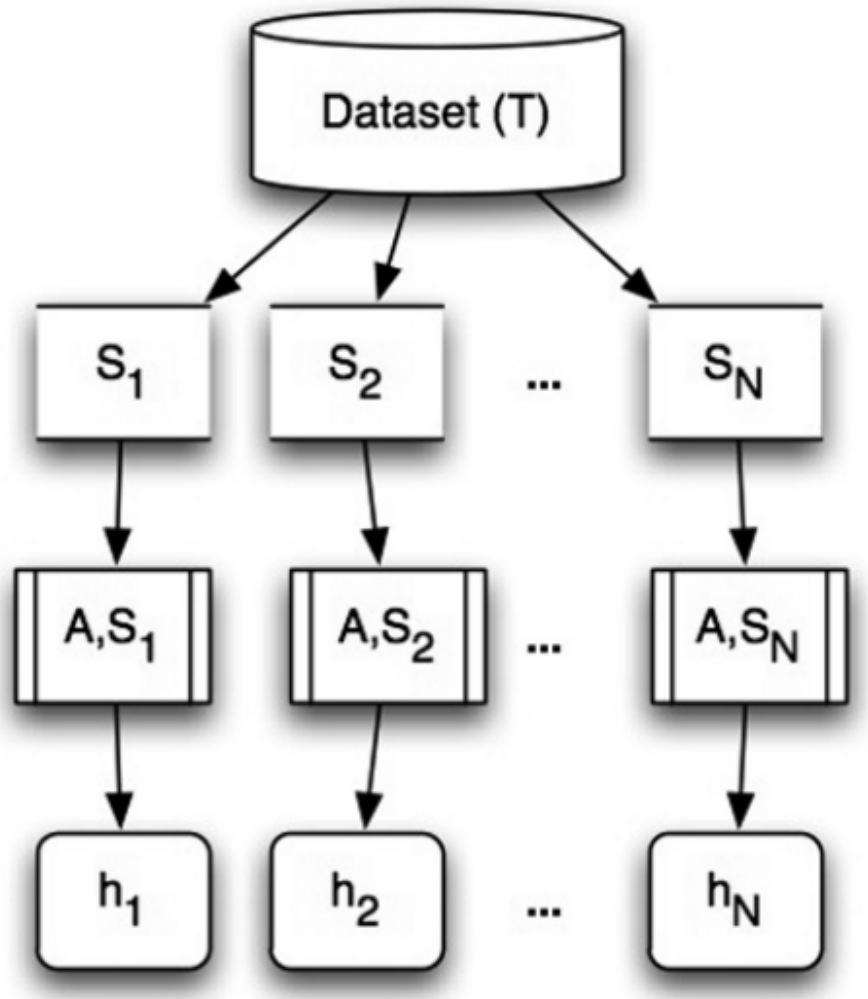
Základní ensemblovací algoritmy

- Pro klasifikační a regresní úlohy se dají použít:
  - bagging,
  - boosting,
  - stacking,
  - cascade generalization.
- Pouze pro klasifikační úlohy se také dají použít:
  - cascading,
  - delegating,
  - arbitrating.

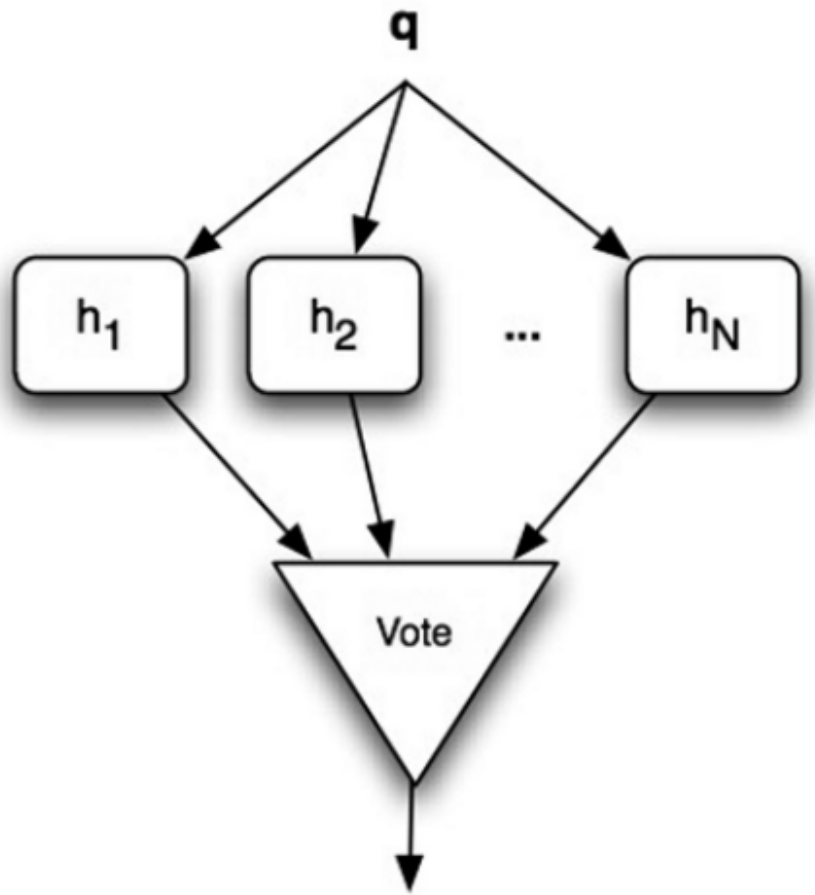
Bagging

- Jednodušší metoda kombinace.
  - Homogenní z hlediska algoritmu učení, tj. použit pouze jeden.
  - Různorodosti dosáhne odlišností trénovacích množin, vzorkování s opakováním do původní velikosti trénovacích dat.
  - Slabé (= dílčí) modely učí nezávisle.
- Výstup ensemblu se určí:
  - pro regresi – spočítám průměrnou hodnotu ze všech výstupů modelů v ensemblu.
  - pro klasifikaci – spočítám majoritu z výstupů modelů v ensemblu.

Bagging (2)



(Meta)Learning

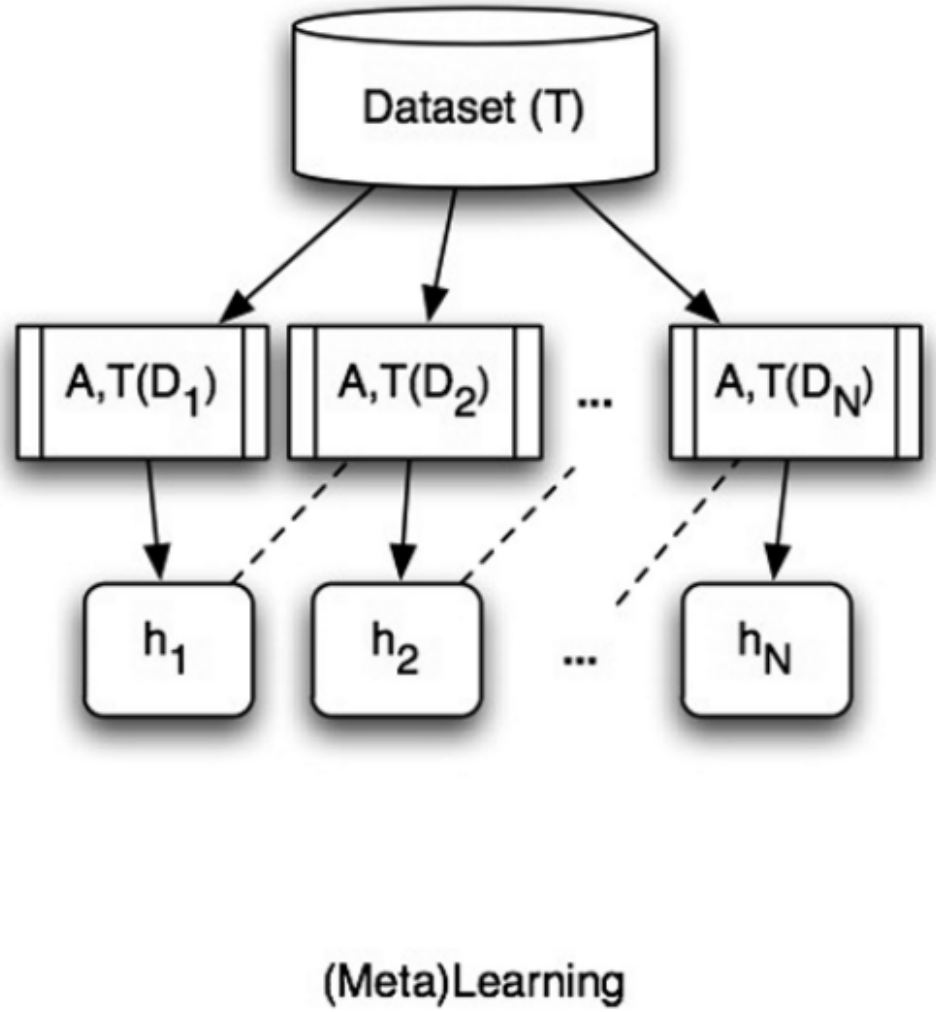


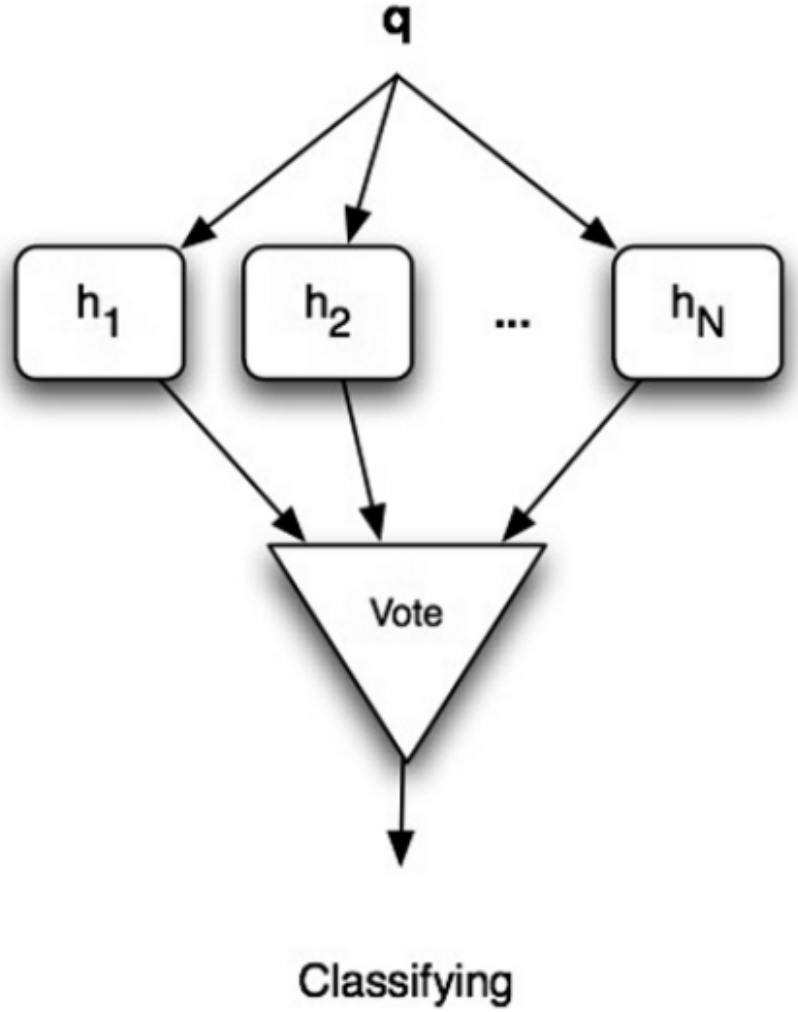
Classifying

Boosting

- Naučím posloupnost modelů, každý další model si bude všímat té části vstupních dat, ve které předchozí modely chybovaly.
- To, jak moc si bude model všímat vstupních dat, se vyjadřuje vahami vstupního vzoru.
- Oklasifikuji trénovací data všemi doposud naučenými modely a vzory, na kterých jsem udělal chybu, přidám do trénovací množiny následujícího modelu.
- Z toho vyplývá, že se modely učí jeden po druhém.
- Výstup ensemblu se spočítá jako vážený průměr (vážená majorita).
- Váhy pro majoritu jsou úměrné přesnosti jednotlivých modelů.

Boosting (2)


Dataset (T)
A,T(D1)
A,T(D2)
...
A,T(DN)
h1
h2
...
hN
(Meta)Learning


q
h1
h2
...
hN
Vote
Classifying

Adaboost

- Nejznámější algoritmus pro Boosting se nazývá Adaboost.
- Základní algoritmus předpokládá klasifikaci do dvou tříd (+1 / -1).
- Značení n je počet vzorů v trénovací množině. h_t je model (klasifikátor).

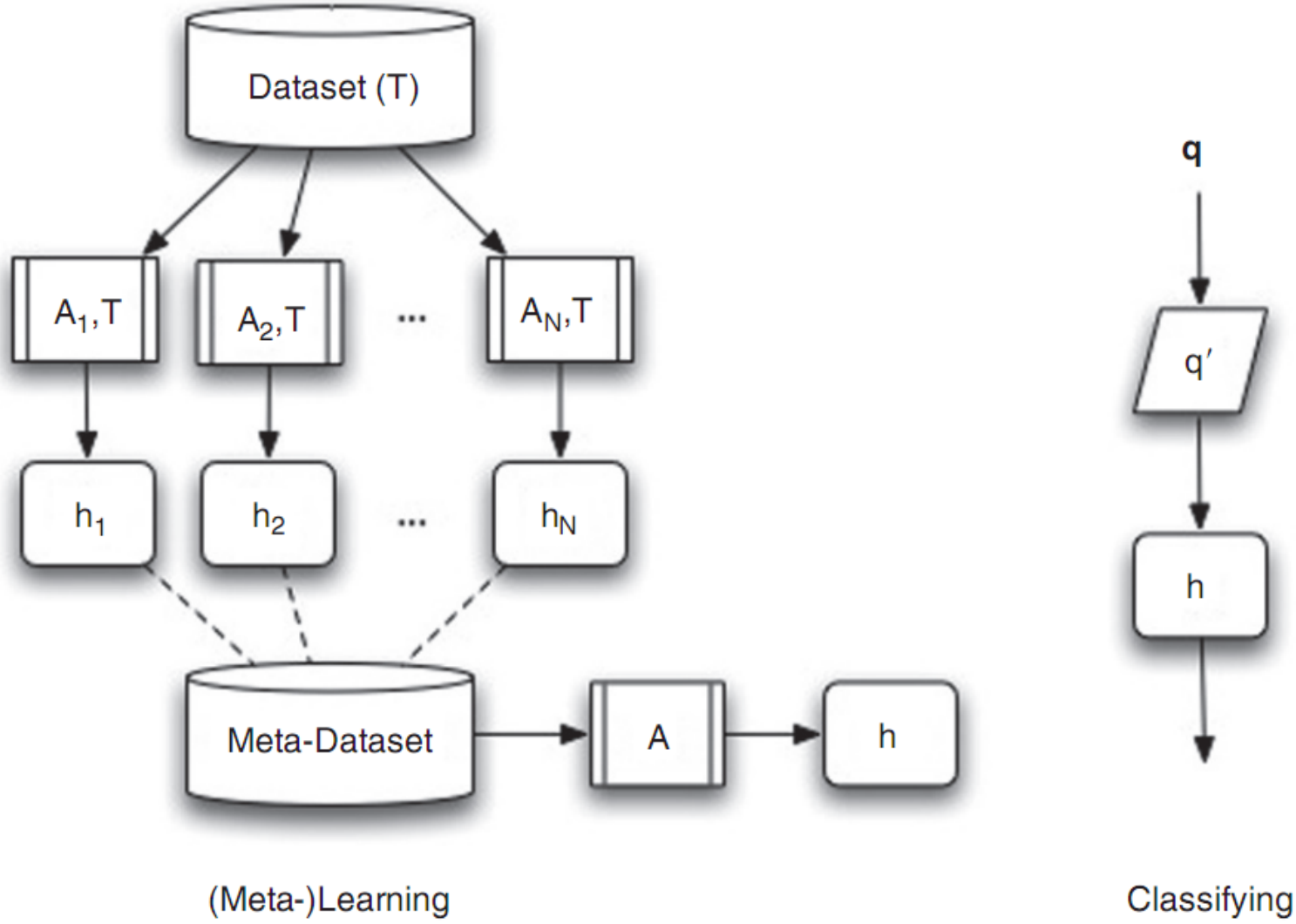
1. Nastav konstantní váhů všech vzorů v trénovací množině na $D_1(i) = \frac{1}{n}$ a nastav $t = 1$.
2. Nauč klasifikátor h_t.
3. Spočítej globální chybu na trénovacích datech $\eta_t = \sum_{\forall i, h_t(x_i) \neq y_i} D_t(i)$
4. Změň váhy všech vstupních vzorů, u kterých klasifikátor h_t udělal chybu. $D_{t+1}(i) = \frac{D_t(i)}{Z_i} \frac{\eta_t}{1-\eta_t}$. $\forall i$, kde $h_t(x_i) \neq y_i$.
5. Pokud globální chyba η_t klesla pod stanovenou hranici, skonči. Jinak pokračuj bodem 2.

- Prezentace věnovaná přímo algoritmu Adaboost
  http://www.robots.ox.ac.uk/~az/lectures/cv/adaboost_matas.pdf

Stacking

- Nezávisle naučím skupinu modelů, použiji odlišné algoritmy učení
  - různé implicitní předpoklady, různé hypotézy, různé chyby.
- Pro určení finálního výstupu použiji místo majority další model
  - říkáme mu meta-model, často jde o logistickou regresi,
  - výstupy jednotlivých modelů slouží jako vstupy meta-modelu,
  - ve srovnání s majoritou větší možnosti pro kombinaci výstupů.

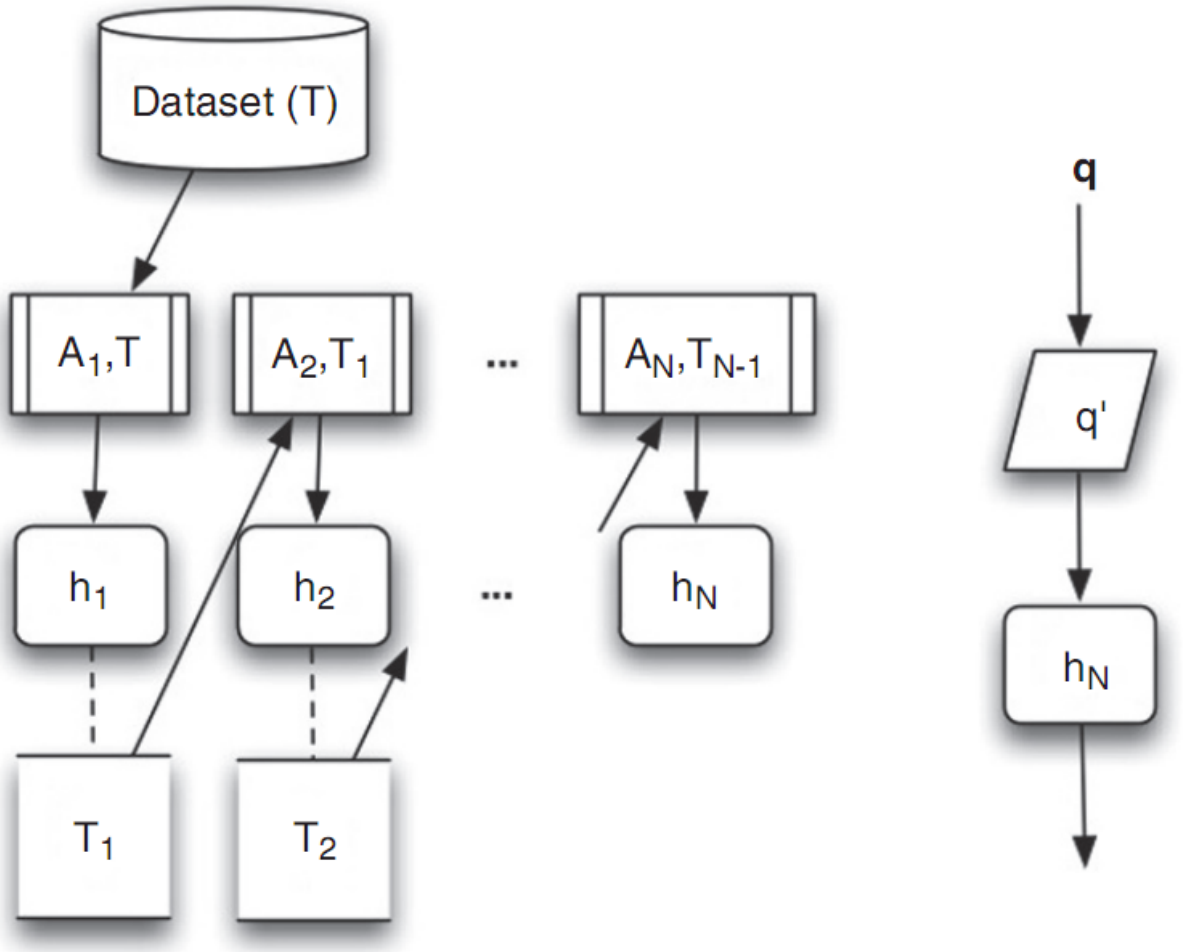
Stacking (2)


Dataset (T)
A1,T
A2,T
...
AN,T
h1
h2
...
hN
Meta-Dataset
A
h
(Meta-)Learning
q
q′
h
Classifying

Cascade generalization

- Modely v ensemblu tvořím postupně a závisle
  - ke vstupním proměnným doplním výstupy předchozích modelů,
  - dochází ke změně v algoritmu učení.
- Vstupem i-tého modelu jsou proměnné $(x_1, x_2, \ldots, x_n, y_1, \ldots, y_{i-1})$
  - kde $x_1, \ldots, x_n$ jsou příznaky ze vstupních dat,
  - kde $y_1, \ldots, y_{i-1}$ jsou výstupy předchozích modelů.
- Modely se učí jeden po druhém a výstupem ensemblu je výstup posledního modelu.

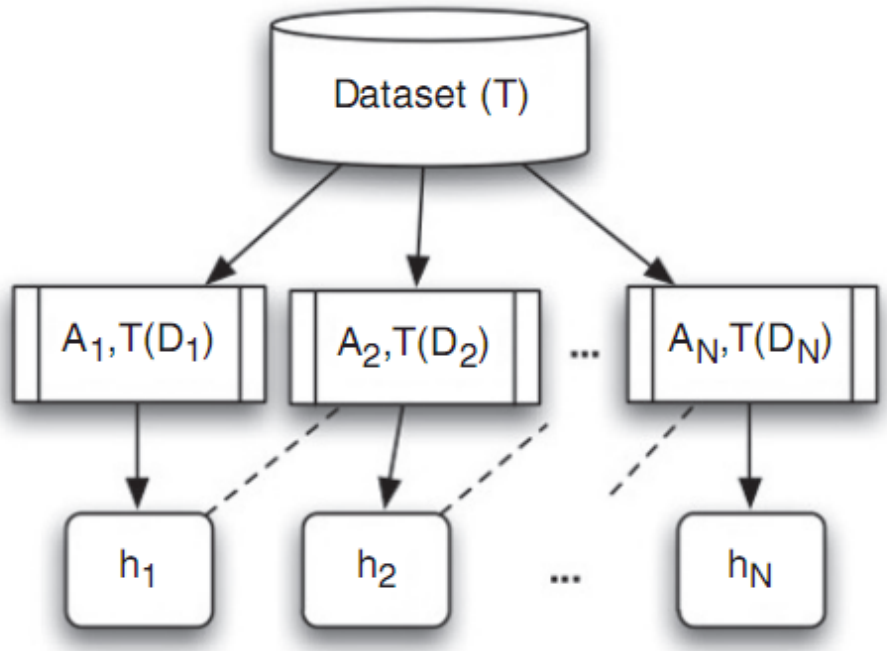
Cascade generalization (2)



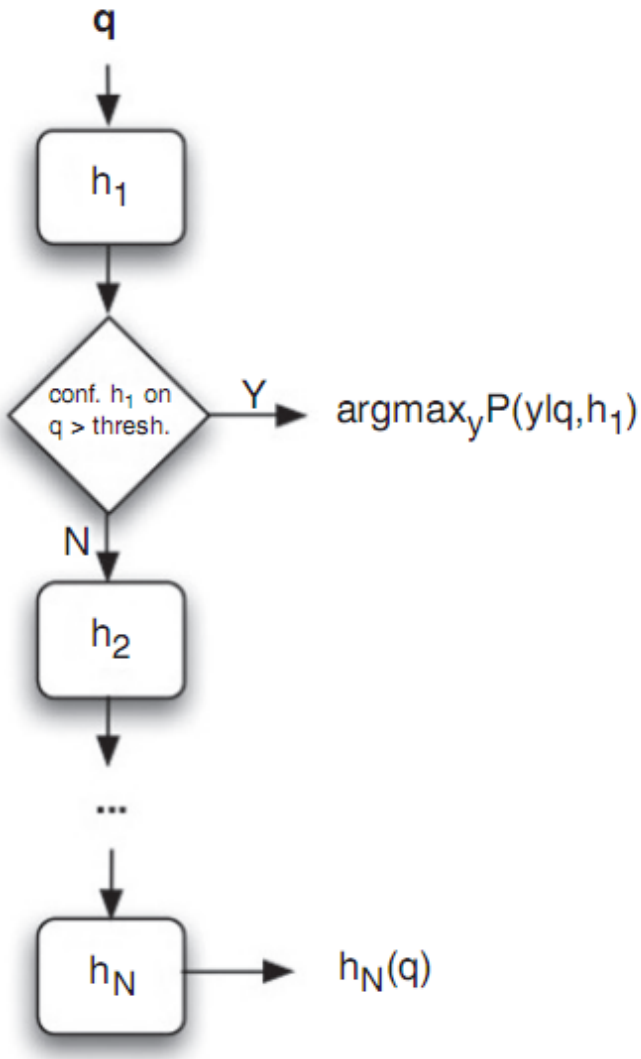
Cascading

- Podobně jako u Boostingu se další modely specializují na vzory, které předchozí modely klasifikovaly špatně – které indikovaly nízkou pravděpodobnost přiřazení vzoru do dané třídy.
- Při počítání výstupu ensemblu se použije výstup modelu, který udává dostatečně vysokou ppst výstupní třídy.

Cascading (2)



(Meta)Learning

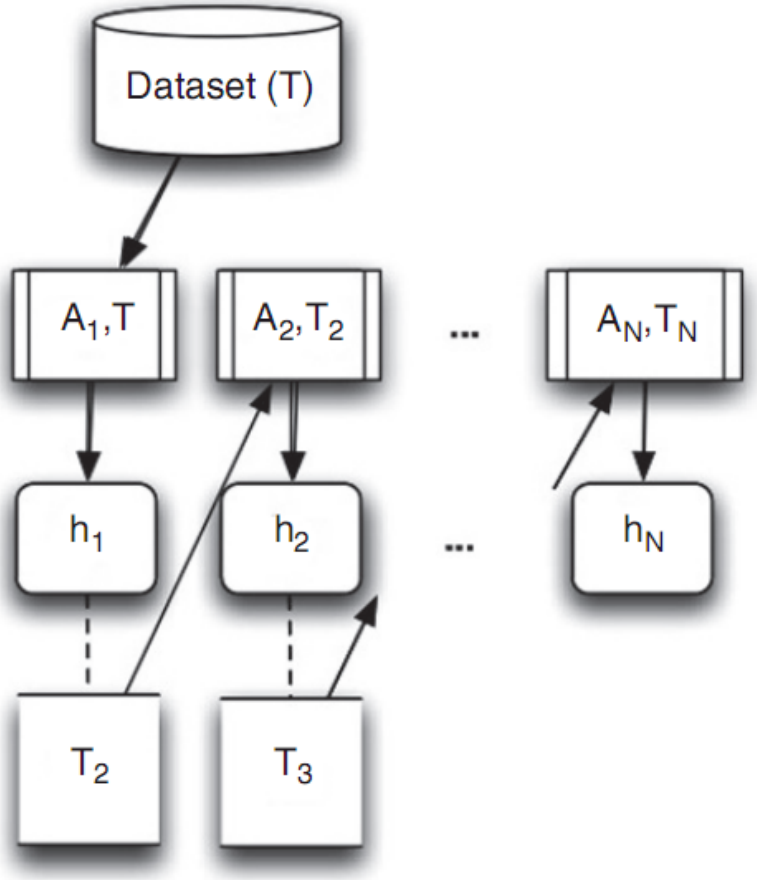


Classifying

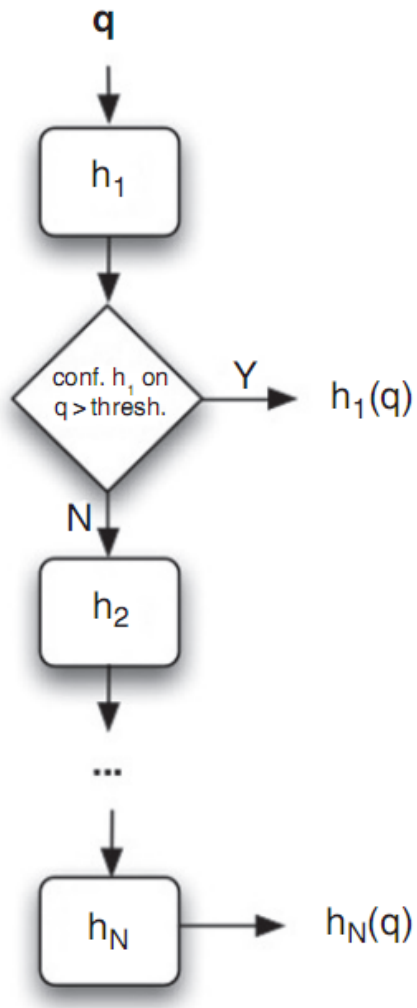
Delegating

- Trénovací množina prvního modelu je celá trénovací množina.
- Do trénovací množiny dalšího klasifikátoru přiřadím vstupní vzory, které byly klasifikovány špatně nebo ppst jejich zařaení do správné třídy je menší než určený práh.
- Výstup ensemblu je výstup modelu, který indikuje dostatečně vysokou ppst přiřazení do dané třídy.

Delegating (2)



(Meta)Learning

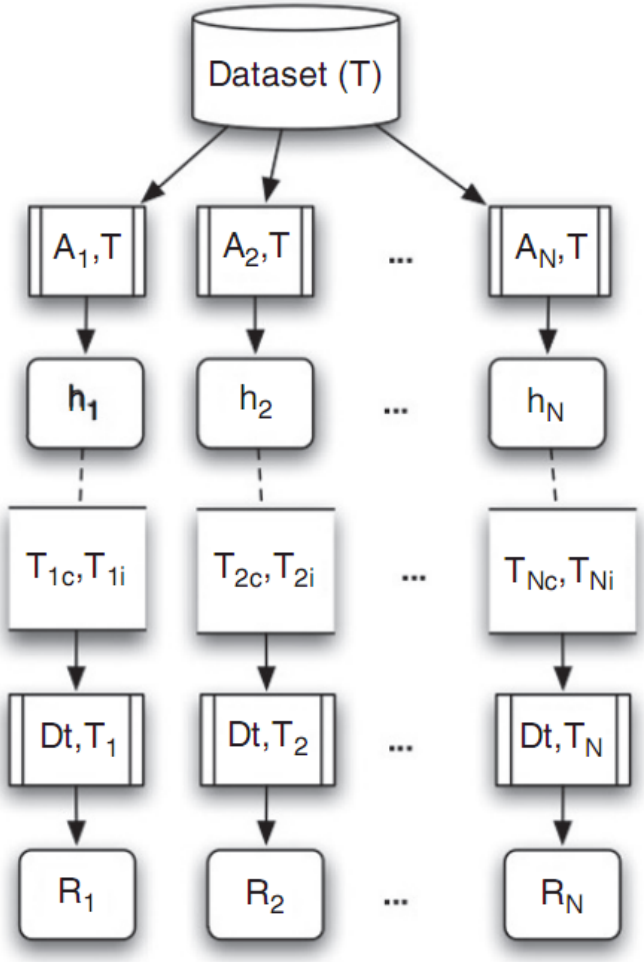


Classifying

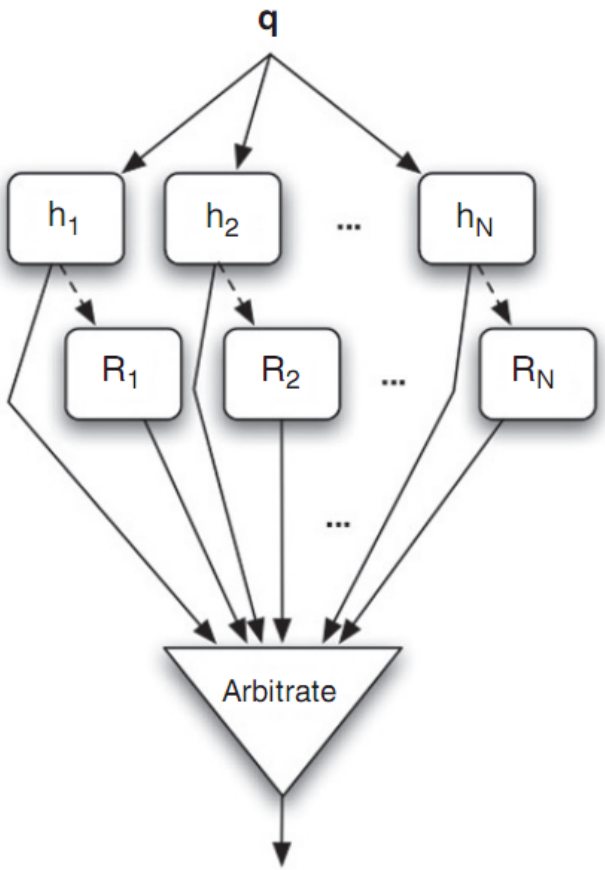
Arbitrating

- Trochu zvláštní metoda, kde jsou dva typy modelů
  - standardní modely, predikující cílovou proměnnou,
  - rozhodčí modely, které predikují úspěšnost standardních modelů.
- Každý stadardní model má svůj rozhočí model.
- Každá dvojice stardní+rozhodčí model je učena nezávisle.
- Výstupem ensemblu je model, jehož rozhodčí predikuje nejvyšší míru úspěchu.

Arbitrating (2)



(Meta)Learning



Classifying

Výběr relevatních příznaků

- Poslední téma tohoto kurzu – opravdu všechny vstupní proměnné potřebuji ke klasifikaci?
- Při klasifikaci zdravých a nemocných lidí asi bude hrát větší roli jejich teplota a tlak, než barva vlasů.
- Techniky, které vybírají vhodné vstupní proměnné, se označují jako feature selection (případně feature ranking) metody.
- A dělí se do dvou hlavních kategorií:
  - feature selection – tyto metody dodají seznam vstupních proměnných (atributů), které považují za důležité,
  - feature ranking – tyto metody přiřadí každému atributu skóre, který indikuje vliv atributu na výstupní třídu.

Feature selection

- Typicky hledají podmnožinu atributů, na které model ještě funguje dobře. Dělí se do 3 hlavních kategorií:
  - Wrappers – vyberou skupinu atributů, nad ní naučí nějaký model, spočítají jeho přesnost a podle přesnosti upraví skupinu atributů, atd...
  - Filters – fungují dost podobně, jen místo modelů se vyhodnocují tzv. filtry.
    - Filtry se v této souvislosti rozumí například korelace mezi vybranou skupinou vstupů a výstupem nebo vzájemná informace, ...
  - Embedded techniques – tento způsob je zabudován do učícího algoritmu modelu a podle toho, které proměnné model využívá, se sestavuje seznam důležitých atributů.

Feature selection (2)

- Při hledání vhodné kombinace se často uplatňuje ”hladový” přístup.
- Nejprve hledám množinu s jedním atributem, která má nejvyšší skóre (například nejvyšší přesnost modelu).
- K této jednoprvkové množině zkouším přidávat další atribut a hledám, který přinese největší zlepšení modelu.
- Pak hledám třetí, a tak dále, dokud se model nepřestane zlepšovat.

Feature ranking

- Přiřazuje každé vstupní proměnné skóre, které určuje její významnost.
- Často se používají stejné metody, které se na předchozím slajdu označovaly jako filters:
  - vzájemná informace mezi jednotlivými atributy a výstupem,
  - korelace,
  - informační entropie,
  - přesnost perceptronu s jedním vstupem.
- Je pak na člověku, jak těchto informací využije.

Shrnutí

- Kombinovování modelů je dnes standardním postupem
  - diverzita slabých modelů spojená s jejich přesností vede k úspěchu,
  - za přesnost platíme časem a někdy i srozumitelností.
- Jako každá jiná metoda podléhá “no free lunch” teorému
  - zlepšení přesnosti nedosáhneme “automaticky a mechanicky”.
- Literatura mj. Dietterich: Ensemble Methods in Machine Learning, Multiple classifier systems, Springer, 2000.