Vytěžování dat, přednáška 12: Kombinování modelů

Miroslav Čepek

Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti

Fakulta elektrotechnická, ČVUT

Ensembles (Kombinování modelů)

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

Motivace

- Soutěž – Netflix prize (predikce, které filmy by se zákazníkovi mohli 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.
- A cenu (2 000 000 USD) získá ten, kdo dokáže na testovací množině zlepšit přesnost o 5%.

Motivace (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 ensembly modelů.

Kombinace modelů

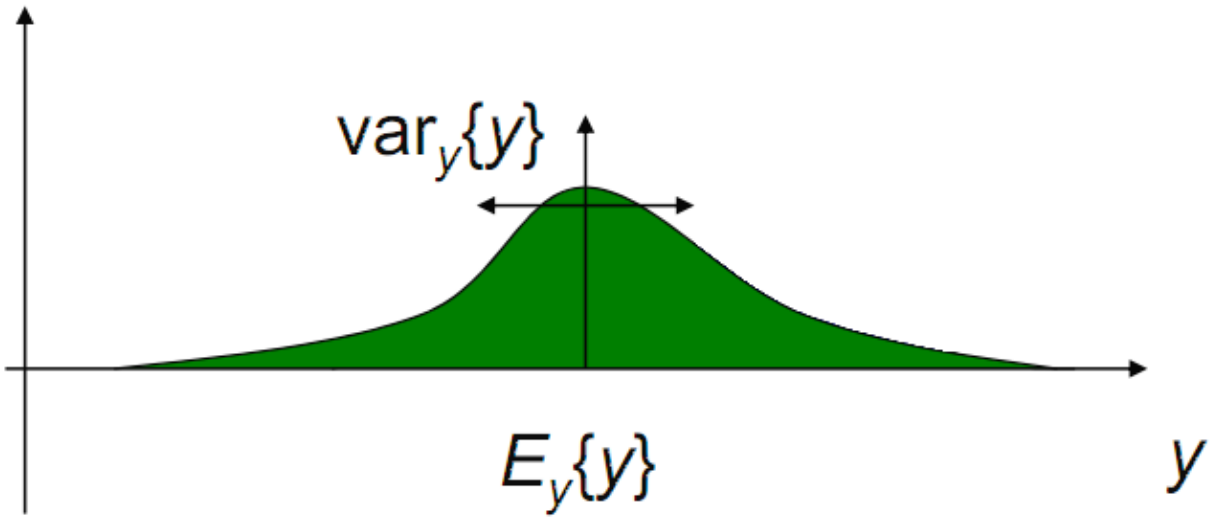
- Kombinace modelů je cestou, jak získat lepší přesnost, než je přesnost nejlepšího z modelů.
- Každý model dělá chyby pro trochu jiná data. A trochu jiné chyby.
- Čili, když se zkombinují dohromady, možná modely v ensemblu své chyby eliminují.
- Také tím, že zkombinuji různé modely, dokáže výsledný model aproximovat rozhodovací hranici, kterou by jinak samostatné modely nedokázaly proložit.
- Analogie z idealizované zkoušky – každý student se naučí látku (každý s jinými chybami) a společně rozhodují o odpovědích na otázky. A v ideálním případě by měli u každé otázky odpovědět správně :).

Variance modelů (rozptyl chyby modelů)

- Vraťme se ještě k chybám modelů.
- Jak jsem povídal posledně, když vytvořím různé modely (i na stejných datech), nebudou mít stejnou chybu, ale jejich chyby se budou lišit.
  - Například použiji jiné počáteční hodnoty,
  - nebo vezmu jinou podmnožinu trénovací množiny.

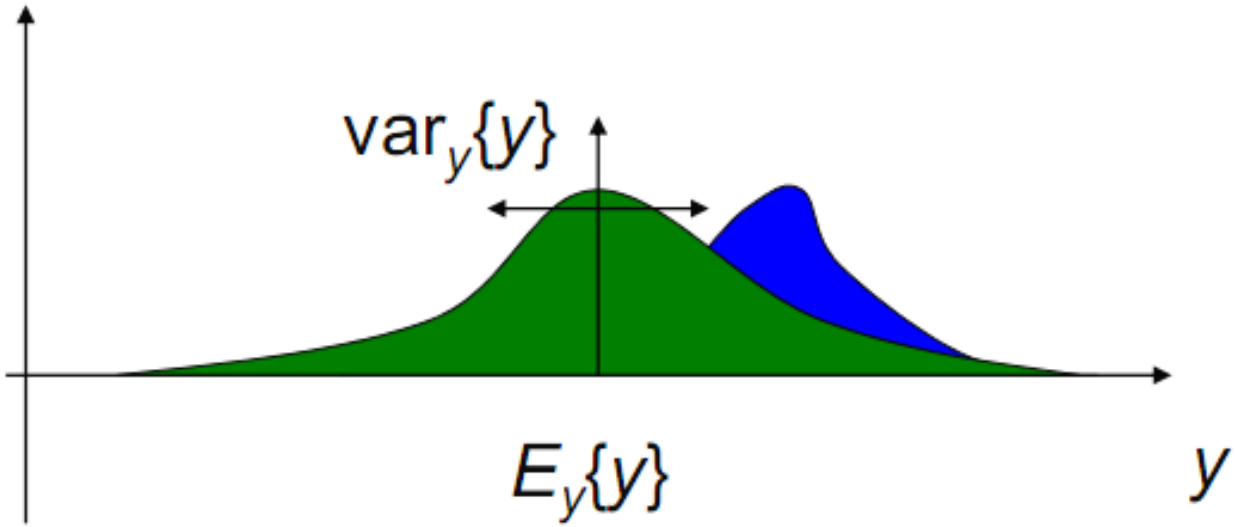
Variance modelů (rozptyl chyby modelů) (2)

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



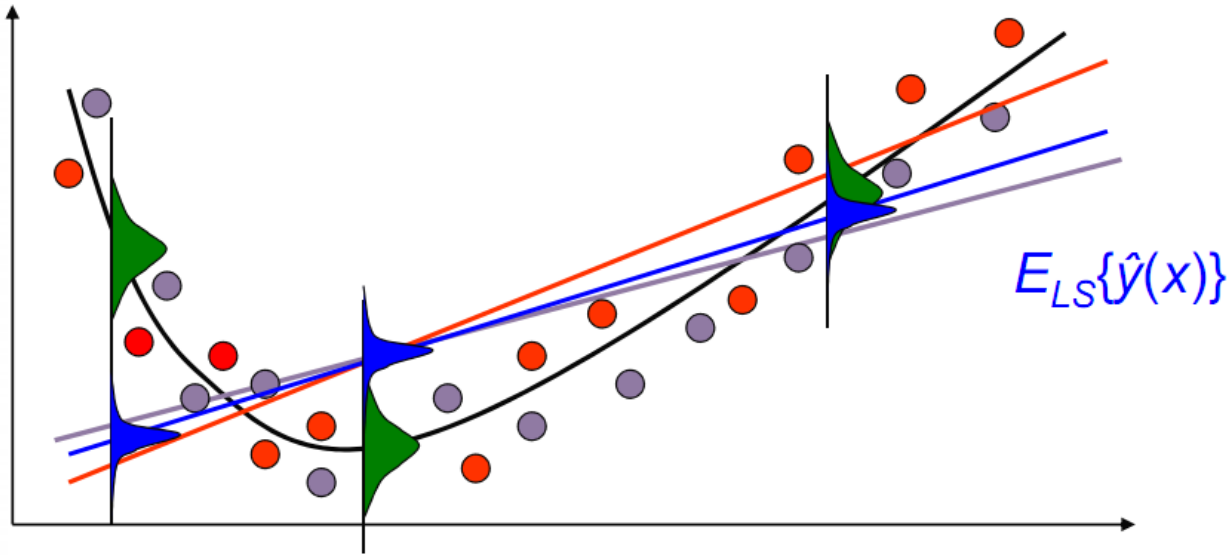
Bias modelů (Zaujetí modelů)

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



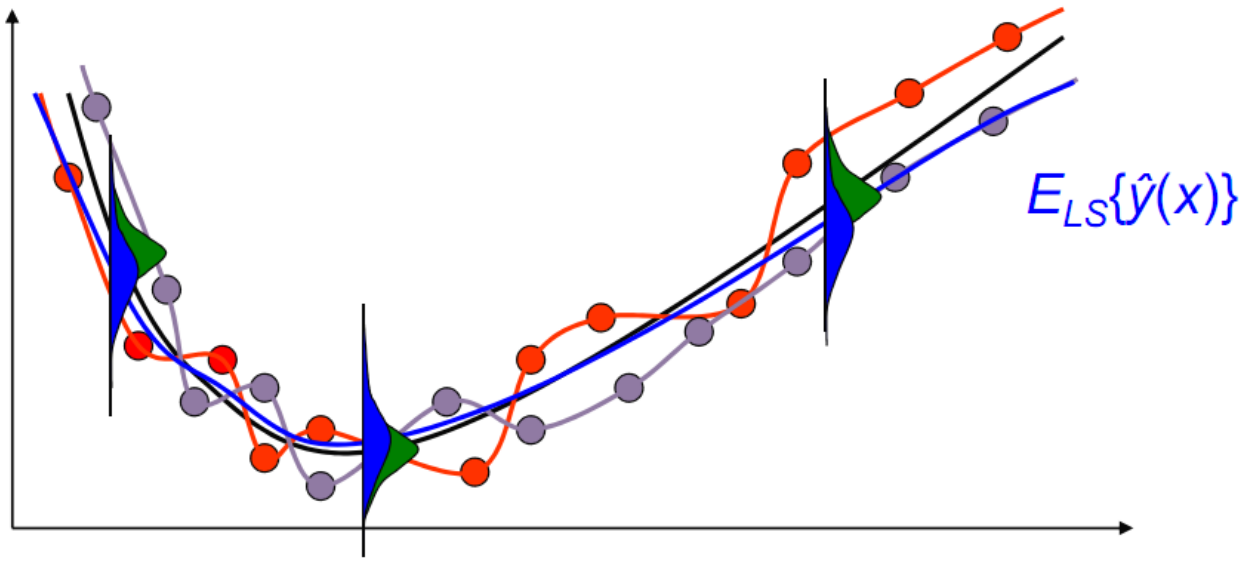
Nedoučení modelu (underfitting)

- 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 proti původním datům.



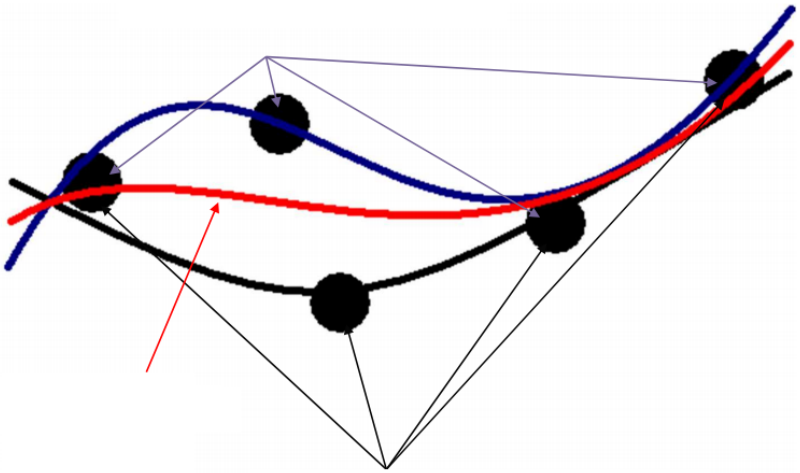
Přeučení modelu (overfitting, už zase)

- Model je příliš ohebný a naučil se i šum, který v datech ve skutečnosti není.
- Modely budou mít nízký bias, ale vysokou varianci, ale nízký bias.
- Co to znamená?
- Modely sice budou mít nízkou chybu na datech, ale jednotlivé modely budou hodně rozdílné.



Kombinování modelů – prevence přeučení

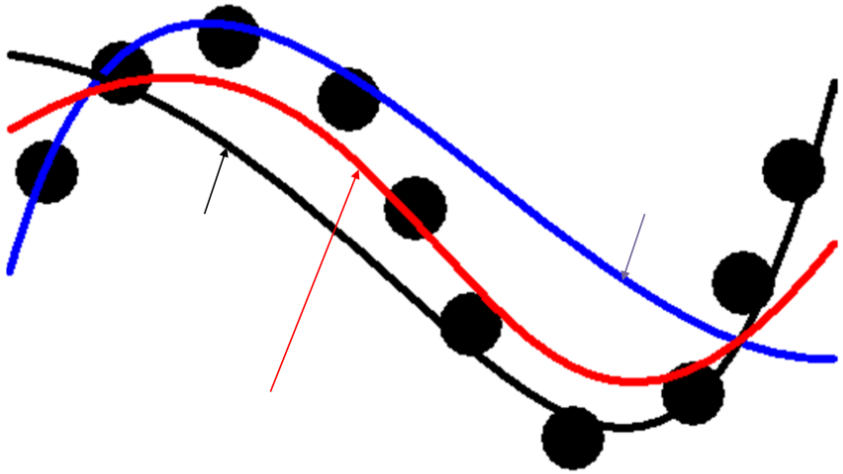
- Přeučení – model se naučí i šum v datech a na testovacích datech vykazuje velké chyby.
- Mám skupinu modelů, a každý jsem naučil na jiné podmnožině trénovacích dat.
- A každý model jsem možná přeučil. Ale můžu něčeho dosáhnout jejich kombinací?



- 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.



- Jejich kombinací opět dokáži získat mnohem "ohebnější" hranici a tedy i menší chybu.

Skupiny modelů

- Síla seskupování modelů tkví v diverzitě (různorodosti) modelů.
- Diverzity modelů můžu dosáhnout dvěma cestami:
  - Vytvořit modely pomocí různých modelovacích technik.
  - Vytvořit modely nad různými podmnožinami trénovací množiny.
- Základní použití všech ensemblovacích algoritmů má následující schéma:
  1. Vyber stavební jednotky ensemblu (vhodné modely) a vytvoř pro každý trénovací množinu.
  2. Natrénuj všechny modely v ensemblu (učení jednotlivých modelů může být závyslé na učení ostatních modelů v ensemblu).
  3. Výstup skupiny modelů – spočítej výstup všech modelů v ensemblu a jejich výstup skombinuj do výsledného výstupu.

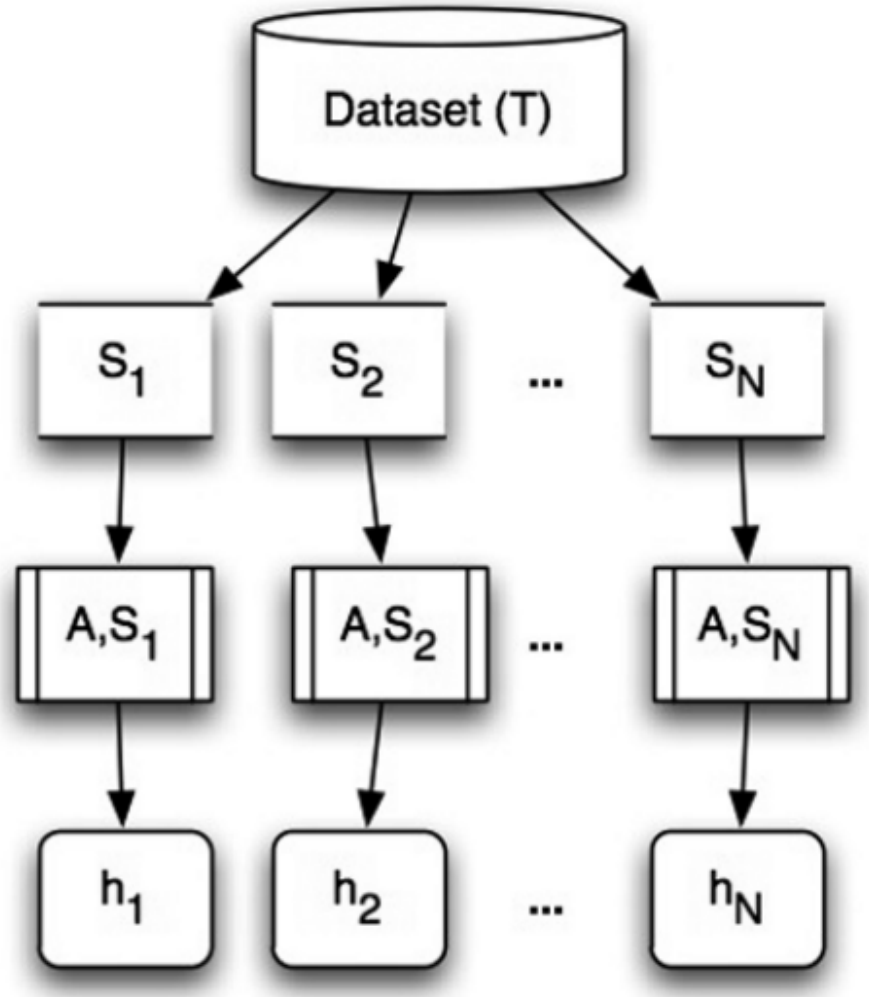
Základní ensemblovací algoritmy

- Pro klasifikační a regresní úlohy se dají použít:
  - Bagging
  - Boosting
  - Stacking
  - Cascade generalization
- Pouze pro klasifikační úlohy se také dají použít:
  - Cascading
  - Delegating
  - Arbitrating

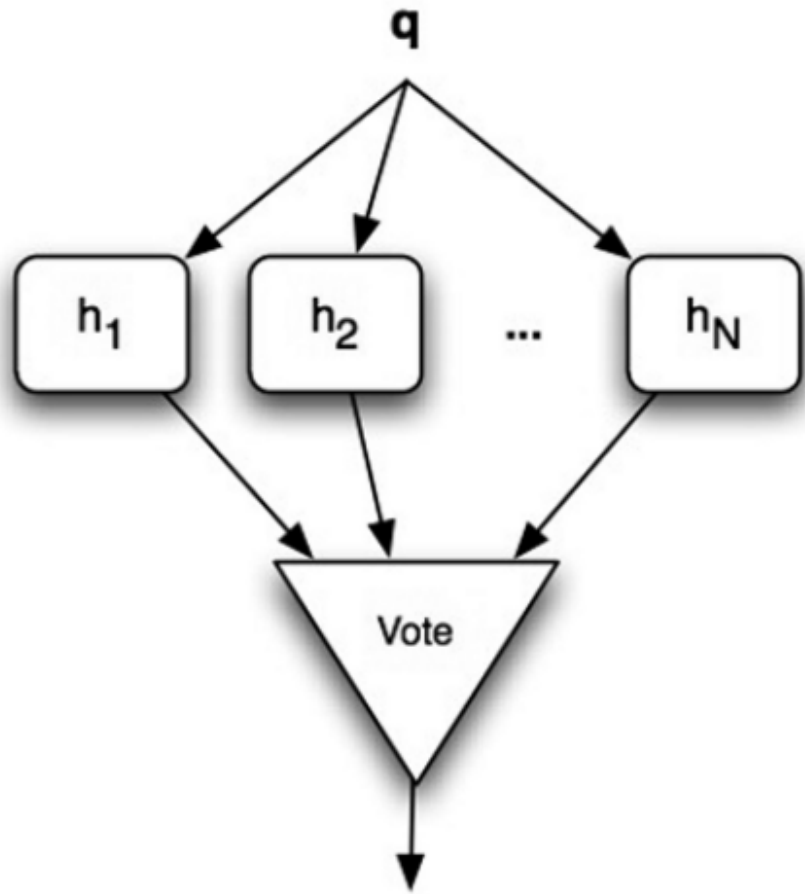
Bagging

- Nejjednodušší ensemblovací metoda.
- Nezávysle naučím skupinu modelů.
- 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

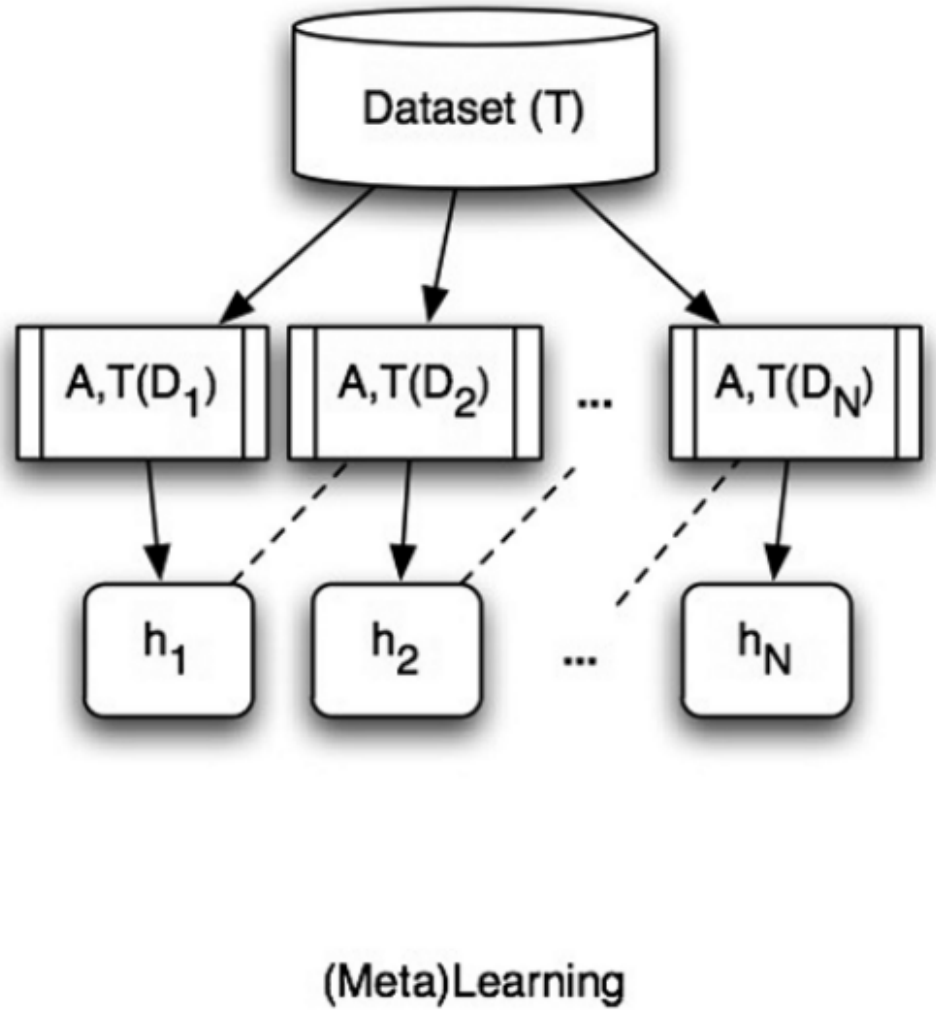


Classifying

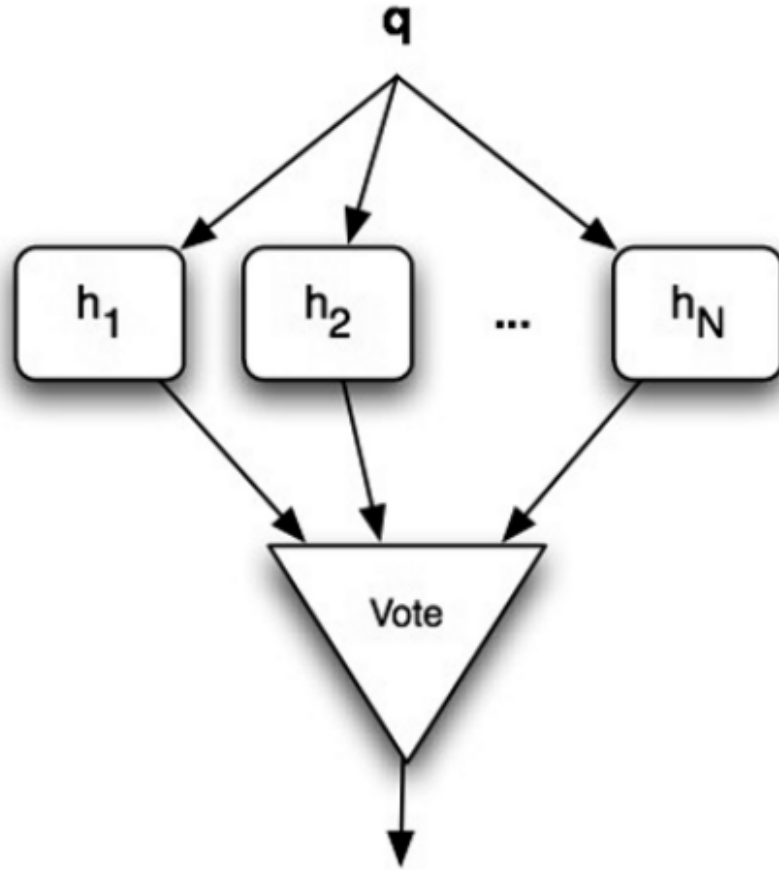
Boosting

- Naučím posloupnost modelů, každý další model se bude všímat té část vstupních dat, ve které předchozí modely chybovaly.
- To, jak moc si bude model všímat vstupních dat se vyjadřuje vahami vstupního vzoru.
- Oklasifikuji trénovací data všemy doposud naučenými modely a vzorům, 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 se určí na základě přesnosti jednotlivých modelů.

Boosting (2)



(Meta)Learning



Classifying

Adaboost

- Nejznámější algoritmus pro Boosing 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).

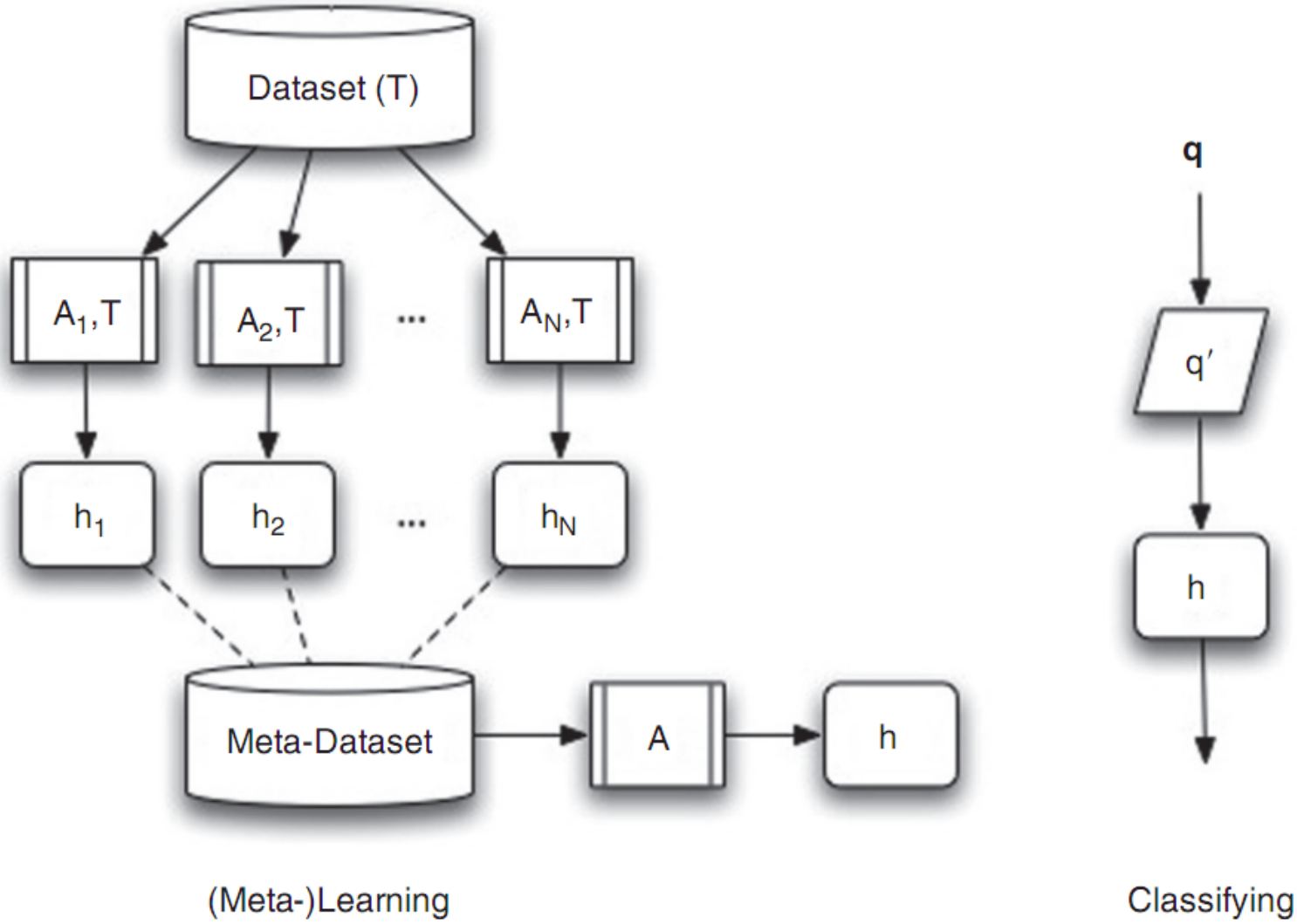
1. Nastav konstantní váhů všech vzorů v trénovací množině na $D_1(i) = \frac{1}{n}$ a nastav $t = 1$.
2. Nauč klasifikátor h_t.
3. Spočítej globální chybu na trénovacích datech $\eta_t = \sum_{\forall i, h_t(x_i) \neq y_i} D_t(i)$
4. Změň váhy všech vstupních vzorů, u kterých klasifikátor h_t udělal 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

- Nezávysle naučím skupinu modelů. Pro určení finálního výstupu použiji místo majority další model (meta model).
- Získám tím větší možnosti pro kombinaci výstupů jednotlivých modelů.
- Výstupy jednotlivých modelů slouží vstupy meta modelu.
- Při počítání výstupu spočítám výstupy jednolivých modelů, které pak pustím do meta modelu, který spočítá skutečný výstup.

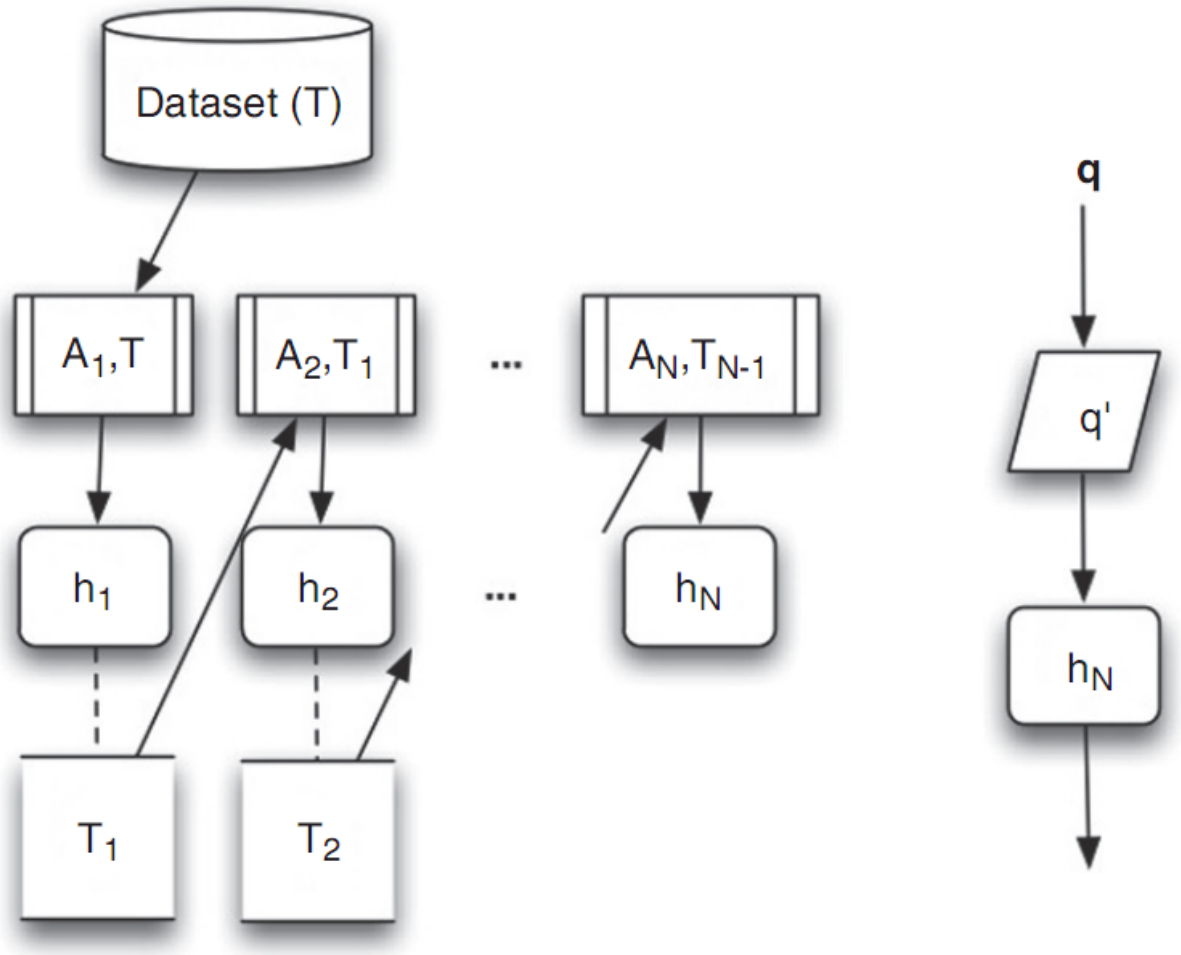
Stacking (2)


Dataset (T)
A_1,T
A_2,T
...
A_N,T
h_1
h_2
...
h_N
Meta-Dataset
A
h
(Meta-)Learning
q
q′
h
Classifying

Cascade generalization

- Modely v ensemblu tvořím postupně, ke vstupním proměnným postupně přidávám výstupy předchozích modelů.
- Vstupem i-tého modelu jsou tedy vstupní proměnné $(x_1, x_2, \ldots, x_n, y_1, \ldots, y_{i-1})$
- Kde $y_1, \ldots, y_{i-1}$ jsou výstupy předchozích modelů.
- Modely je tedy učí jeden po druhém a výstupem ensemblu je výstup posledního modelu.

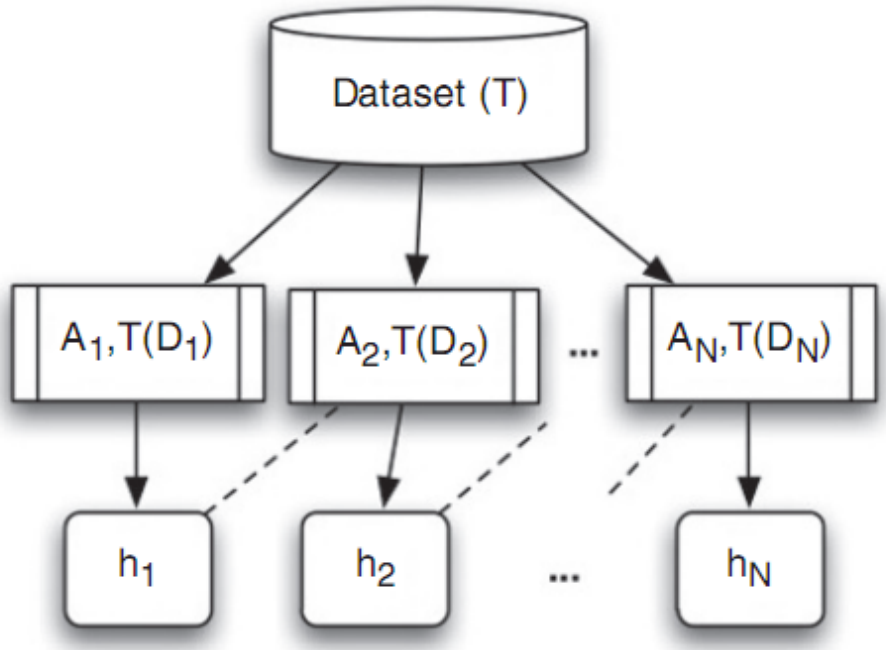
Cascade generalization (2)



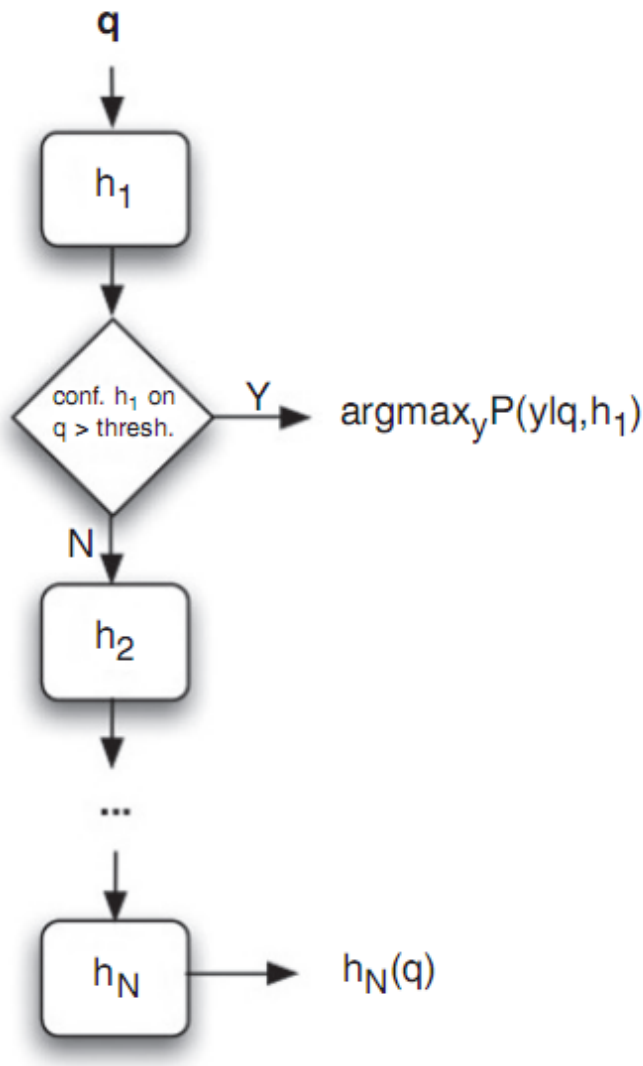
Cascading

- 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

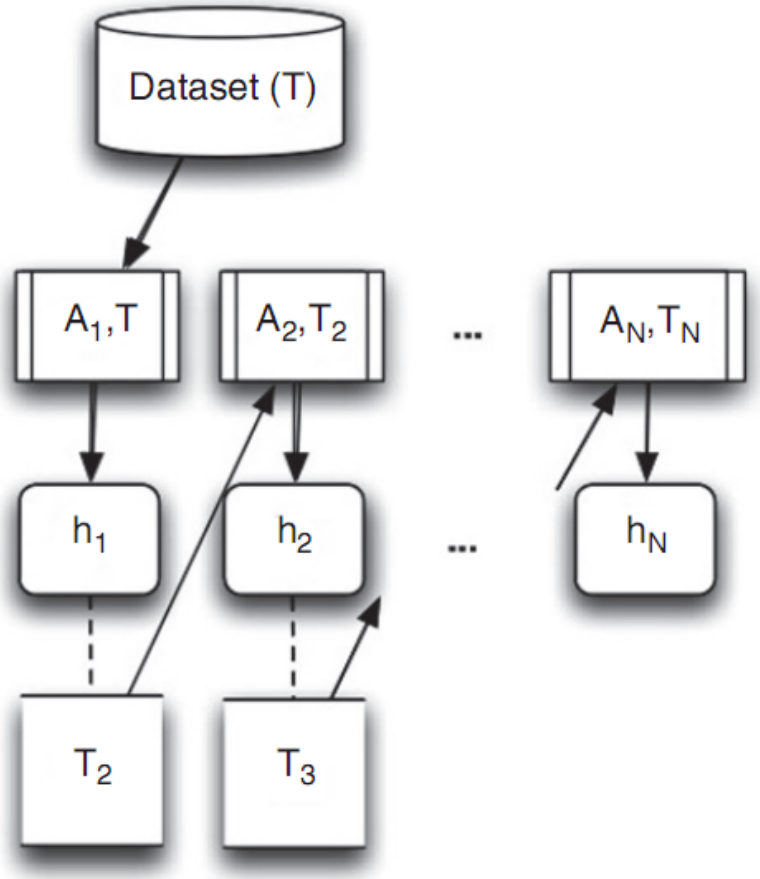


Classifying

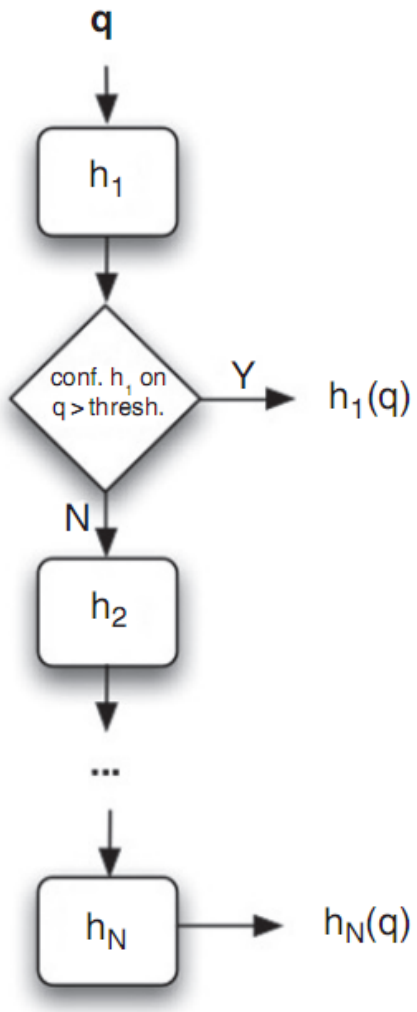
Delegating

- Trénovací množina prvního modelu je celá trénovací množina.
- Do trénovací množiny dalšího klasifikátoru se přiřadím stupní 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

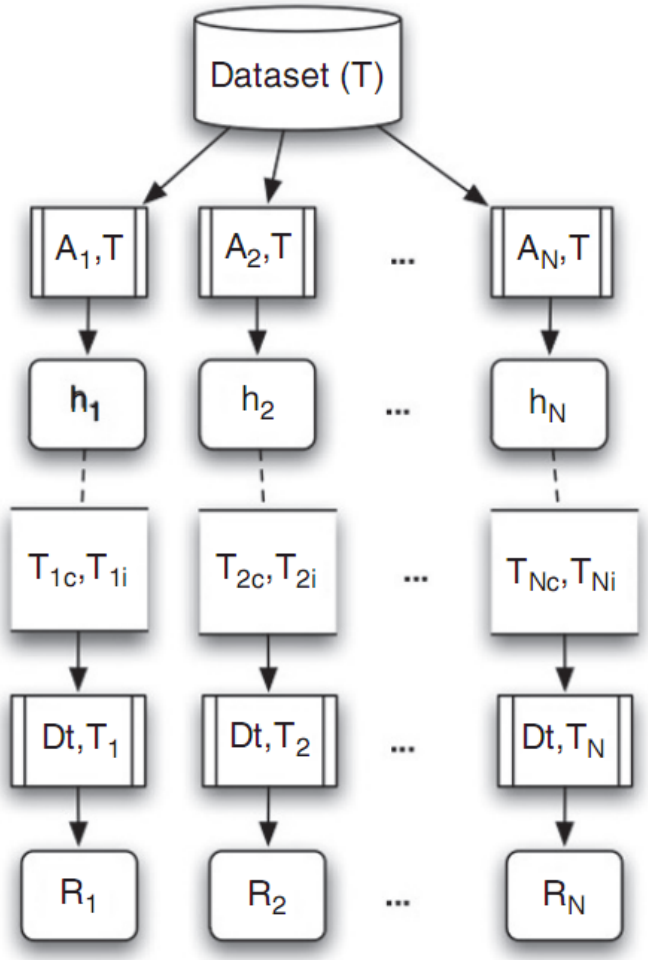


Classifying

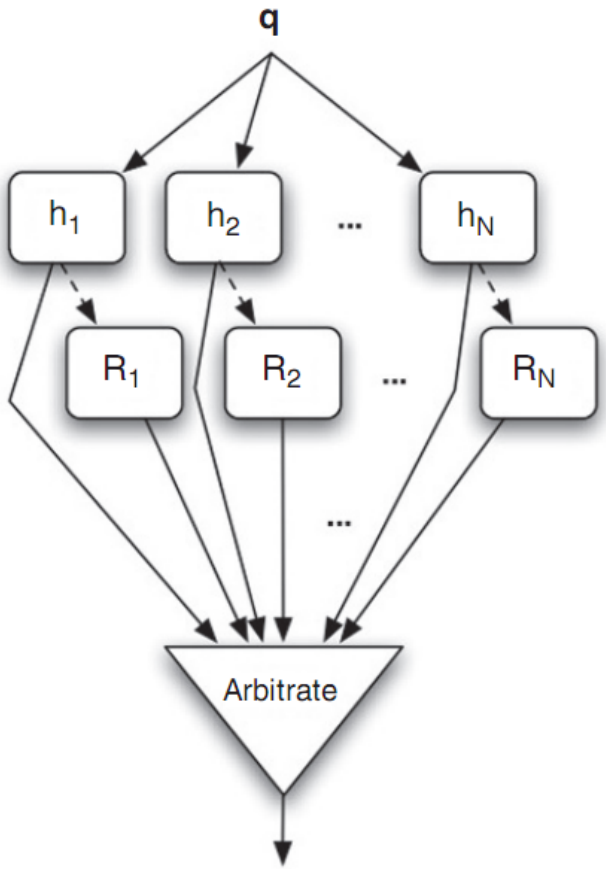
Arbitrating

- 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.