Výpočetní teorie učení.
PAC učení. VC dimenze.

Petr Pošík

Czech Technical University in Prague
Faculty of Electrical Engineering
Dept. of Cybernetics

Výpočetní teorie učení
(Computational Learning Theory, COLT)

Úvod a základní pojmy

Koncept

Příklady:

- ✔ sudé číslo, dvoustopé vozidlo, aktivní politik, krásný člověk, korektní hypotéza

Proč má smysl koncepty zavádět?

- ✔ Čím se liší sudá čísla od lichých? Aktivní politik od ostatních politiků?

Definiční obor X je množina všech možných instancí objektů:

- ✔ množina všech celých čísel, všech možných vozidel, všech politiků ...

Objekt $x \in X$ je popsán hodnotami atributů:

- ✔ číslo {hodnota}
- ✔ vozidlo {výrobce, typ motoru, počet náprav, ...}
- ✔ politik {počet hlasování ve sněmovně, počet předložených návrhů zákonů, počet pozměňovacích návrhů, počet interpelací na členy vlády, ...}

Cílový koncept $c \in C$ odpovídá nějaké podmnožině množiny X, $c \subseteq X$:

- ✔ každá instance $x \in X$ je buď *příkladem* nebo *protipříkladem* konceptu c
- ✔ charakteristická funkce $f : X \to \{0, 1\}$
  - ✘ když $f(x) = 1$, x je pozitivním příkladem konceptu
  - ✘ když $f(x) = 0$, x je negativním příkladem (protipříkladem) konceptu
- ✔ Konceptem c může být libovolná bool. funkce f nad X!

Hypotéza

Úloha induktivního učení: najdi hypotézu (model) h, která co nejlépe odpovídá cílovému (modelovanému) konceptu c, znáš-li

- ✔ pouze podmnožinu $D \subset X$ příkladů (a protipříkladů) cílového konceptu (trén. data) a
- ✔ prostor H všech možných hypotéz.

Hypotéza je pokus o popis cílového konceptu.

- ✔ H je prostor všech možných přípustných hypotéz
- ✔ v nejobecnějším případě i hypotéza h může být libovolná boolovská funkce $h : X \to \{0, 1\}$
- ✔ hypotéza h je také podmnožina množiny X: $h \subseteq X$

Cíl učení:

- ✔ najít hypotézu h, která je korektní pro všechny příklady z X, t.j.

$$\forall x \in X : h(x) = c(x).$$

COLT: Cíle

COLT se snaží teoreticky charakterizovat

1. *obtížnost problémů* strojového učení
   - ✔ Za jakých podmínek je učení vůbec možné?
2. *schopnosti algoritmů (modelů)* strojového učení
   - ✔ Za jakých podmínek je jistý algoritmus učení schopen učit se *úspěšně*?

COLT se snaží odpovídat na otázky typu:

- ✔ Existují nějaké třídy složitosti problémů nezávisle na použitém modelu/algoritmu?
- ✔ Jaký typ modelu (třídu hypotéz) máme použít? Je některý algoritmus konzistentně lepší než jiný?
- ✔ Kolik trénovacích příkladů je třeba, aby se model (hypotéza) úspěšně naučil?
  (Jsou i jiné prostředky než křivka učení?)
- ✔ Je-li prostor hypotéz rozsáhlý, je vůbec možné najít v rozumném čase nejlepší hypotézu?
- ✔ Jak složitá by výsledná hypotéza (model) měla být?
- ✔ Najdeme-li hypotézu, která je korektní pro trénovací data $D \subset X$,
  jak si můžeme být jistí, že hypotéza bude korektní i pro ostatní data $X \setminus D$???
- ✔ Kolik chyb algoritmus udělá předtím, než se úspěšně naučí (než najde dostatečně úspěšnou hypotézu)?

Generalizace

Schopnost *generalizace*:

- ✔ schopnost algoritmu vytvořit model, který umí klasifikovat správně i příklady, jež nebyly v trénovací sadě D.
- ✔ měří se chybovostí na $X \setminus D$.

Nevíme-li nic o problému, je nějaký důvod preferovat jeden model (algoritmus učení) nad jiným?

Označme:

- ✔ $P_A(h)$: apriorní pravděpodobnost, že algoritmus A vygeneruje hypotézu h
- ✔ $P_A(h|D)$: pravděpodobnost, že algoritmus A vygeneruje h, zná-li trénovací data D:
  - ✘ v případě deterministických algoritmů (nejbližší soused, rozhodovací stromy), $P_A(h|D)$ je všude nulové kromě jediné hypotézy
  - ✘ v případě stochastických algoritmů (neuronová síť trénovaná s náhodnými počátečními vahami) je rozdělení $P_A(h|D)$ nenulové pro větší počet hypotéz
- ✔ $P(c|D)$: rozdělení konceptů konzistentních s trénovacími daty D

Neznáme-li cílový koncept c, je přirozenou mírou generalizační schopnosti algoritmu jeho očekávaná chyba přes všechny koncepty při dané množině D:

$$E(\mathrm{Err}_A|D) = \sum_{h,c} \sum_{x \in X \setminus D} P(x) \cdot I(c(x) \neq h(x)) \cdot P(h|D) \cdot P(c|D)$$

Bez znalosti $P(c|D)$ nelze 2 algoritmy porovnat na základě jejich generalizační chyby!!!

Příklad

Mějme

- ✔ objekty popsané 3 binárními atributy,
- ✔ 1 určitý koncept c,
- ✔ 2 deterministické algoritmy a k nim příslušné hypotézy
  h_1 a h_2: pamatují si trénovací data, nová data zařazuje jeden do třídy 1, druhý do třídy -1.

	$\mathbf{x}$	c	h_1	h_2
D	000	1	1	1
	001	-1	-1	-1
	010	1	1	1
$X \setminus D$	011	-1	1	-1
	100	1	1	-1
	101	-1	1	-1
	110	1	1	-1
	111	1	1	-1

Pro daný koncept c:

- ✔ $E(\mathrm{Err}_{A_1}|c, D) = 0.4$, $E(\mathrm{Err}_{A_2}|c, D) = 0.6$,
- ✔ algoritmus A_1 je jasně lepší než algoritmus A_2.

Při tvorbě modelu ale neznáme cílový koncept c!

- ✔ Předpokládáme, že nemáme žádnou apriorní informaci o konceptu c, t.j. že všechny cílové koncepty jsou stejně pravděpodobné.
- ✔ Trénovací množina D
  - ✘ umožní eliminovat nekonzistentní hypotézy (v našem případě 224),
  - ✘ ale neumožní vybrat tu správnou mezi hypotézami konzistentními s D (zbývá jich 32),
  - ✘ protože průměrováno přes všechny koncepty c konzistentní s D, obě hypotézy jsou stejně úspěšné!

No Free Lunch

"No Free Lunch" teorém: Pro každé 2 algoritmy A_1 a A_2 (představované rozděleními pravděpodobnosti $P_{A_1}(h|D)$ a $P_{A_2}(h|D)$) platí následující, a to nezávisle na vzorkovacím rozdělení $P(\mathbf{x})$ a na konkrétní trénovací množině D:

1. Průměrováno přes všechny koncepty c, $E(\mathrm{Err}_{A_1}|c, D) = E(\mathrm{Err}_{A_2}|c, D)$.
2. Průměrováno přes všechna rozdělení $P(c)$, $E(\mathrm{Err}_{A_1}|c, D) = E(\mathrm{Err}_{A_2}|c, D)$.

Důsledky NFL:

- ✔ Ať se jakkoli snažíte navrnout 1 skvělý a 1 příšerný algoritmus, jsou-li všechny koncepty stejně pravděpodobné, pak budou oba algoritmy stejně úspěšné.
- ✔ Je-li alg. 1 na některých úlohách (konceptech) lepší než alg. 2, musí být na jiných úlohách horší.
- ✔ Všechna tvrzení stylu "alg. 1 je lepší než alg. 2" netvrdí nic o algoritmech samotných, ale spíše o aplikační oblasti (o množině konceptů, na nichž byly algoritmy testovány).
- ✔ V praxi pro jistou aplikační oblast hledáme algoritmus, který
  - ✘ funguje hůř na úlohách, které v dané oblasti neočekáváme, a
  - ✘ funguje lépe na úlohách, které jsou velmi pravděpodobné.
- ✔ **Generalizace není možná, pokud model není nějakým způsobem předpojatý (mnohdy implicitně), a je tím lepší, čím víc se předpoklady modelu shodují se skutečně platnými zákonitostmi aplikační oblasti!!!**

Bias

Inductive bias (předpojatost, zaujetí modelu):

- ✔ souhrn všech (i implicitních) předpokladů, které model (algoritmus učení) o aplikační oblasti činí
- ✔ využití těchto předpokladů dovoluje modelu generalizovat, t.j. poskytovat správné predikce pro dosud neznámá data, pokud předpoklady souhlasí se skutečností

Možné zdroje zaujetí modelu: využití apriorních znalostí o aplikační oblasti

- ✔ Jazyková předpojatost (language bias):
  - ✘ jazyk pro popis hypotéz nesouhlasí s jazykem pro popis možných konceptů
  - ✘ některé koncepty není možné ve zvoleném jazyce hypotéz vůbec vyjádřit
  - ✘ jiný jazyk ale může umožnit efektivní učení
- ✔ Preferenční předpojatost (preference bias):
  - ✘ algoritmus preferuje některou z hypotéz konzistentních s D
  - ✘ algoritmus preferuje mírně nekonzistentní hypotézu na úkor konzistentních hypotéz
  - ✘ Occamova břitva
- ✔ ...

PAC učení 10 / 22

PAC učení

Pravděpodobně skoro správné učení (Probably Approximately Correct, PAC):

- ✔ charakterizuje třídy konceptů, které je/není možné se naučit pomocí jisté třídy hypotéz
  - ✘ z "rozumného" počtu trénovacích příkladů
  - ✘ s "rozumnou" výpočetní náročností,

  a to
  - ✘ pro konečný prostor hypotéz i
  - ✘ pro nekonečný prostor hypotéz (kapacita, VC dimenze).
- ✔ definuje přirozenou míru složitosti pro prostory hypotéz (VC dimenze), která nám umožní omezit potřebnou velikost trénovací sady pro induktivní učení.

Předpoklady PAC učení:

- ✔ *Nezávislost:* Příklady $E_i = (x_i, c_i)$ jsou vzorkovány nezávisle, tj. $P(E_i|E_{i-1}, E_{i-2}, \ldots) = P(E_i)$.
- ✔ *Stacionarita:* Budoucí příklady jsou vzorkovány ze stejného rozdělení $P(E_i) = P(E_{i-1}) = \ldots$ jako minulé příklady.
- ✔ Příklady splňující tyto požadavky se často označují jako "i.i.d." (independent and identically distributed).

(Pro jednoduchost v dalším předpokládejme, že koncept c je deterministický a že je prvkem třídy hypotéz H.)

Chybovost hypotézy

Skutečná chybovost hypotézy h:

- ✔ vzhledem k cílovému konceptu c a
- ✔ vzhledem k rozdělení příkladů $P(X)$

$$\mathrm{Err}(h) = \sum_{x \in X} I(h(x) \neq c(x)) \cdot P(x)$$

- ✔ pravděpodobnost, že hypotéza zaklasifikuje příklad x špatně (viz křivka učení)

Hypotéza h je *skoro správná* nebo *ϵ-skoro správná*,

- ✔ pokud $\mathrm{Err}(h) \leq \epsilon$,
- ✔ kde ϵ je malá konstanta.

Je možné určit počet trénovacích příkladů potřebný k tomu, abychom se naučili koncept c zcela správně?

- ✔ Je-li množina trénovacích příkladů D pouze podmnožinou X, stále existuje více hypotéz konzistentních s D (viz NFL).
- ✔ Trénovací příklady se vybírají náhodně a mohou být zavádějící.

PAC model

PAC model definuje, co vlastně znamená *úspěšně se naučit* ve vztahu ke konceptům.

- ✔ Nevyžaduje se možnost naučit se *jakýkoli koncept* definovatelný nad X:
  - ✘ Zajímají nás jisté podmnožiny všech konceptů $C \subseteq 2^X$. (Některé koncepty se naučit nelze, viz např. případ, kdy X (a tudíž i C) je nekonečný a H konečný.)
  - ✘ Podobně, algoritmus A bude hledat hypotézu h v jisté třídě hypotéz H.
  - ✘ Může a nemusí platit, že $C = H$.
- ✔ Nevyžaduje se nulová chyba naučených hypotéz h.
  - ✘ Místo toho ji omezíme jistou malou konstantou ϵ.
- ✔ Nevyžaduje se, aby algoritmus vrátil hypotézu s akceptovatelnou chybou *vždy*.
  - ✘ Místo toho omezíme pravděpodobnost tohoto jevu malou konstantou δ.

Třída konceptů C se dá **PAC-naučit** (is **PAC-learnable**) pomocí třídy hypotéz H, jestliže

- ✔ pro všechny koncepty $c \in C$, všechna rozdělení $P(X)$, $X = \{0,1\}^n$, libovolné $0 < \epsilon, \delta < 1$
- ✔ existuje polynomiální algoritmus A, který vrátí s pravděpodobností alespoň $1 - \delta$ hypotézu h s $\mathrm{Err}(h) \leq \epsilon$
- ✔ s využitím nanejvýš polynomiálního množství trénovacích příkladů $(x_i, c(x_i))$ vzorkovaných z $P(X)$.
- ✔ "Polynomiální": rostoucí nanejvýš polynomiálně s $\frac{1}{\epsilon}$, $\frac{1}{\delta}$ a n.

Konzistentní PAC učení

Konzistentní algoritmus učení

- ✔ pro jakýkoli i.i.d. vzorek D (trénovací data) konceptu $c \in C$
- ✔ vrátí hypotézu $h \in H$ konzistentní s D.

Složitost vzorku (sample complexity)

- ✔ velikost m trénovací množiny D potřebná pro PAC-učení konceptu c pomocí H
- ✔ roste s dimenzí problému n
- ✔ představuje mez velikosti trénovací množiny pro konzistentní algoritmy učení

Kolik trénovacích příkladů potřebujeme, abychom mohli *s vysokou pravděpodobností říct*, že *všechny konzistentní hypotézy jsou skoro správné*?

- ✔ Označme množinu vážně špatných hypotéz $H_B = \{h \in H : \mathrm{Err}(h) > \epsilon\}$, $h_B \in H_B$.
- ✔ $\Pr(h_B \text{ souhlasí s 1 trénovacím příkladem}) \leq 1 - \epsilon$
- ✔ $\Pr(h_B \text{ souhlasí se všemi trénovacími příklady}) \leq (1-\epsilon)^m$ (Příklady jsou nezávislé.)
- ✔ $\Pr(H_B \text{ obsahuje nějakou konzistentní hypotézu}) \leq |H_B|(1-\epsilon)^m \leq |H|(1-\epsilon)^m$
  Pravděpodobnost, že některá z konzistentních hypotéz není skoro správná.
- ✔ Omezme pravděpodobnost tohoto jevu konstantou δ: $|H|(1-\epsilon)^m \leq \delta$.
- ✔ S využitím $1 - \epsilon \leq e^{-\epsilon}$:

$$m \geq \frac{1}{\epsilon}(\ln \frac{1}{\delta} + \ln |H|)$$

Je-li h konzistentní s m případy, pak $\mathrm{Err}(h) \leq \epsilon$ *s pravděpodobností alespoň* $1 - \delta$.

Sample complexity

Složitost vzorku:

$$m \geq \frac{1}{\epsilon}(\ln \frac{1}{\delta} + \ln |H|)$$

Je-li H třída všech boolovských funkcí nad n atributy:

- ✔ $|H| = 2^{2^n}$
- ✔ Složitost vzorku m roste jako $\ln |H|$, tj. jako 2^n.
- ✔ Jenže počet možných příkladů je taky 2^n.
- ✔ PAC-učení ve třídě všech boolovských funkcí vyžaduje trénování na všech (nebo téměř všech) příkladech!
- ✔ Důvod:
  - ✘ H obsahuje dostatek hypotéz, abychom mohli klasifikovat jakoukoli množinu příkladů jakýmkoli způsobem
  - ✘ Pro jakoukoli trénovací množinu m příkladů, počet s nimi konzistentních hypotéz, které příklad $\mathbf{x}_{m+1}$ zaklasifikují jako pozitivní, je stejný jako počet konzistentních hypotéz, které tento příklad zaklasifikují jako negativní.
  - ✘ Viz NFL: *aby byla možná generalizace, je třeba omezit prostor hypotéz H*.

Pozorování: m je funkcí $|H|$:

- ✔ Podaří-li se nám získat nějakou doplňkovou informaci (omezení na tvar přípustných hypotéz) a vtělit ji do algoritmu (zavést bias), pak vystačíme s menším počtem trénovacích příkladů!!! Významnou roli hraje **doménová znalost**.

Příklad: Rozhodovací seznam

Připomeňme: Rozhodovací seznam (DL)

- ✔ je sekvence testů, každý test je konjunkce literálů.
- ✔ Uspěje-li test, vrátí k němu přiřazenou třídu. Jinak se pokračuje dalším testem.
- ✔ *Neomezený* DL může reprezentovat libovolnou boolovskou funkci!

Omezme prostor hypotéz H na jazyk k-DL:

- ✔ rozhodovací seznamy, kde každý test může být konjunkcí nejvýše k literálů
- ✔ jazyk k-DL obsahuje jako svou podmnožinu jazyk k-DT (množinu všech rozhodovacích stromů s hloubkou nejvýš k)
- ✔ konkrétní instance jazyka k-DL závisí na množině atributů (na použité reprezentaci)
- ✔ označme k-DL(n) jazyk k-DL nad n boolovskými atributy

Jak ukázat, že třídu hypotéz k-DL se lze PAC-naučit?

1. Ukázat, že složitost vzorku je nejvýše polynomiální (viz další slide).
2. Ukázat, že existuje učicí algoritmus s polynomiální výpočetní složitostí. (Neuvedeno, např. nějaká varianta CN2.)

Příklad: Rozhodovací seznam (pokr.)

Ukažme, že jakoukoli hypotézu z k-DL lze přesně aproximovat učením na rozumně velké trénovací sadě:

- ✔ Potřebujeme odhadnout počet hypotéz v daném jazyku.
- ✔ Označme množinu testů (konjunkcí nejvýše k literálů nad n atributy) jako $Conj(n,k)$.
- ✔ Každý test může mít k sobě přiřazenou výstupní hodnotu "Ano", "Ne", nebo se v seznamu nemusí vyskytnout, takže existuje nejvýše $3^{|Conj(n,k)|}$ různých množin testů.
- ✔ Každá z těchto množin testů může být v libovolném pořadí, takže $|k{-}\mathrm{DL}(n)| \leq 3^{|Conj(n,k)|} \cdot |Conj(n,k)|!$.
- ✔ Počet konjunkcí nejvýše k literálů s n atributy: $|Conj(n,k)| = \sum_{i=0}^{k} \binom{2n}{i} = \mathcal{O}(n^k)$.
  $2n$, protože každý jednotlivý test atributu se v konjunkci může objevit i v negaci.
- ✔ Po úpravách: $k{-}\mathrm{DL}(n) = 2^{\mathcal{O}(n^k \log_2(n^k))}$
- ✔ Substituujeme-li tento výsledek za $|H|$ do vztahu pro složitost vzorku: $m \geq \frac{1}{\epsilon}\left(\ln\frac{1}{\delta} + \mathcal{O}\left(n^k \log_2(n^k)\right)\right)$
- ✔ m je polynomiální v n
- ✔ Jakýkoli algoritmus, který vrátí k-DL konzistentní s trénovacími daty, se PAC-naučí k-DL koncept s rozumným počtem trénovacích příkladů.

Příklad: Formule v DNF

Disjunktivní normální forma (DNF):

- ✔ Objekty popsány n boolovskými atributy: $a_1, \ldots, a_n$
- ✔ Formule v DNF: disjunkce konjunkcí, např. $(a_1 \wedge \neg a_2 \wedge a_5) \vee (\neg a_3 \wedge a_4)$

Jak velká je množina hypotéz H:

- ✔ 3^n možných konjunkcí
- ✔ $|H| = 2^{3^n}$ možných disjunkcí těchto konjunkcí
- ✔ $\ln H = 3^n \ln 2$ není polynomiální v n
- ✔ Nepodařilo se nám ukázat, že formule v DNF se lze PAC-naučit. (Ale také jsme to nevyvrátili.)

PAC-naučitelnost formulí v DNF je otevřený problém.

Ukázky výsledků pro PAC učení

1. Konjunktivní koncepty lze PAC-učit, ale
   koncepty ve formě disjunkce 2 konjunkcí PAC-učit nelze.
2. Lineární prahované koncepty (perceptrony) lze PAC-učit jak v boolovských, tak reálných prostorech. Ale
   konjunkce 2 perceptronů nelze PAC-učit, podobně jako disjunkce 2 perceptronů a vícevrstvé perceptrony se dvěma skrytými jednotkami. Pokud navíc omezíme váhy na hodnoty 0 a 1, pak ani perceptrony v boolovském prostoru nelze PAC-učit.
3. Třídy k-CNF, k-DNF a k-DL lze PAC-učit pro zvolené k. Ale
   nevíme, zda lze PAC-učit DNF formule, CNF formule nebo rozhodovací stromy.

VC dimenze

Nevýhoda použití $|H|$ ve složitosti vzorku:

- ✔ je to "worst-case" odhad
- ✔ mnohdy velmi nadhodnocuje počet potřebných trénovacích příkladů
- ✔ $|H|$ nelze použít pro nekonečné hypotézové prostory

Kapacita, Vapnik-Chervonenkisova dimenze $VC(H)$

- ✔ jiná míra flexibility (složitosti) třídy hypotéz H: kvantifikuje předpojatost vlastní jisté omezené třídě hypotéz H
- ✔ aplikovatelná i na nekonečné prostory H
- ✔ může poskytnout těsnější mez pro velikost vzorku
- ✔ **Definice:** $VC(H)$ je maximální počet d příkladů $x \in X$ takový, že pro kterýkoli z 2^d způsobů označení příkladů x za pozitivní a negativní existuje v H hypotéza, která je s těmito příklady konzistentní.

Složitost vzorku pomocí VC dimenze:

- ✔ Mějme prostor hypotéz H a prostor konceptů C, $C \subseteq H$.
- ✔ Pak jakýkoli konzistentní algoritmus pro učení C pomocí H bude mít složitost vzorku

$$m \geq \frac{1}{\epsilon}\left(4\log_2\frac{2}{\delta} + 8 \cdot VC(H) \cdot \log_2\frac{13}{\epsilon}\right)$$

VC dimenze (pokr.)

Příklady VC dimenze některých H:

- ✔ VC dimenze lineární diskriminační funkce v 1D prostoru? 2.
  Lin. disk. funkce není schopna správně reprezentovat všechny možné koncepty definované nad 3 body v 1D prostoru.
- ✔ VC dimenze lineární diskriminační funkce ve 2D prostoru? 3.
  Lin. diskr. funkce není schopna správně reprezentovat všechny možné koncepty definované nad 4 body ve 2D prostoru.
- ✔ Obecně pro lin. disk. funkci $f_n(\mathbf{x}) = w_0 + w_1x_1 + \ldots + w_nx_D$ platí, že $VC(f_n) = n + 1$
- ✔ Příklad 1D diskr. funkce f s $VC(f) = \infty$: $f(x) = \sin(\alpha x)$
  Lze ukázat, že $\sin(\alpha x)$ v 1D prostoru dokáže správně klasifikovat jakoukoli množinu bodů.
- ✔ VC dimenze SVM s RBF jádrem, kde penalizační člen v kritériu může být jakýkoli: $VC(f_{SVM-RBF}) = \infty$

Další využití VC dimenze:

- ✔ odhad skutečné (testovací) chyby klasifikátoru jen na základě trénovacích dat
- ✔ "structural risk minimization", základní princip SVM

Shrnutí

- ✔ **Generalizace vyžaduje předpoklady!!!**
- ✔ NFL: Všechny modely/algoritmy jsou průměrně stejně dobré
  - ✘ pokud na nějaké třídě problémů fungují nadprůměrně, na jiné musí fungovat podprůměrně
  - ✘ chceme najít modely, které
    - ✓ fungují podprůměrně na úlohách, které se v praxi moc nevyskytují
    - ✓ fungují nadprůměrně na úlohách, které jsou pro nás důležité
- ✔ Probably Approximately Correct (PAC) učení
  - ✘ specifikace pojmu "model se učí správně"
  - ✘ tolerance ve velikosti chyby ϵ a v pravděpodobnosti, že tato chyba bude překročena δ
  - ✘ umožňuje odhadnout potřebnou velikost trénovací množiny
- ✔ VC dimenze
  - ✘ míra flexibility (třeba i nekonečné) třídy hypotéz
  - ✘ obvykle poskytuje těsnější odhady potřebné velikosti trénovací sady