Základy teórie strojového učenia (v.0)

Bui Truc Lam Tomáš Vinař

Fakulta matematiky, fyziky a informatiky Univerzity Komenského v Bratislave,
Mlynská dolina, 842 48 Bratislava

Obsah

1 Štatistický model učenia, základné definície a predpoklady

Množinu všetkých možných vstupov označíme X, niekedy jej budeme hovoriť aj *množina inštancií*. Množinu všetkých možných výstupov označíme Y. Proces, ktorým získavame dáta (x, y), vieme formalizovať pomocou pravdepodobnostného rozdelenie P nad priestorom $X \times Y$, ktoré každému možnému pozorovaniu priradí nejakú pravdepodobnosť (resp. hustotu pravdepodobnosti v prípade spojitého rozdelenia).

Trénovacie príklady získame ako t nezávislých vzoriek z rozdelenia P. Množinu trénovacích príkladov budeme volať *trénovacia množina* a označíme ju T. (Z matematického hľadiska to ale nie je množina, kedže jedna dvojica (x, y) sa medzi trénovacími príkladmi môže vyskytovať viackrát. Pre jednoduchosť to ale budeme nazývať množinou.)

Poznámka 1 (Skrátené zápisy stredných hodnôt.). Vzhľadom k tomu, že pravdepodobnostné rozdelenie P nám bude slúžiť v tomto texte ako "univerzálny" model sveta, zápisy stredných hodnôt budeme zjednodušovať nasledovne:

- ak sa stredná hodnota berie cez príklady z rozdelenia P:

$$\mathrm{E}_{(x,y)\sim P} \equiv \mathrm{E}_{x,y}$$

- ak sa stredná hodnota berie cez všetky možné trénovacie množiny veľkosti t:

$$\mathrm{E}_{T\in P^t} \equiv \mathrm{E}_T$$

Podobné skrátené zápisy budeme používať aj pri pravdepodobnostiach a pod.

Postup, ktorým zostrojíme funkciu h, vieme formalizovať ako algoritmus L, ktorý na vstupe dostane niekoľko trénovacích príkladov $(x_1, y_1), \ldots, (x_t, y_t)$, a na výstupe vráti funkciu $h : X \to Y$, ktorú budeme volať *hypotéza*. Táto hypotéza bude pochádzať z takzvanej *množiny hypotéz*, ktorú označíme H.

1.1 Chyba hypotézy, očakávaná trénovacia a testovacia chyba

Chybu hypotézy budeme vyhodnocovať pomocou *chybovej funkcie* $\mathrm{err} : Y \times Y \to \mathbb{R}_0^+$, pričom $\mathrm{err}(y, y')$ vyjadruje, ako veľmi sa od seba líšia výstupy y a y'.

Pri klasifikácii sa naša hypotéza vždy buď trafí, alebo netrafí do správnej odpovede. Správnej odpovedi priradíme chybu 0, nesprávnej odpovedi chybu 1:

$$\mathrm{err}(y, y') = \begin{cases} 0, & \text{ak } y = y' \\ 1, & \text{inak} \end{cases} .$$

Pri regresii budeme v nasledujúcom texte používať funkciu $\mathrm{err}(y, y') = (y - y')^2$ (takzvaná L_2 chybová funkcia), často sa používa aj funkcia $\mathrm{err}(y, y') = |y - y'|$ (takzvaná L_1 chybová funkcia).

Chyba na jednom príklade. Keď dostaneme nejaký príklad (x, y), vieme zmerať, ako veľmi sa naša hypotéza h na tomto vstupe pomýlila, ako $\mathrm{err}(h(x), y)$. *Trénovacia chyba* je chyba, ktorú hypotéza nadobúda na trénovacej množine. Budeme ju označovať Err, a vypočítame ju nasledovne:

$$\mathrm{Err}_T(h) = \frac{1}{t} \cdot \sum_{i=1}^{t} \mathrm{err}(h(x_i), y_i).$$

Očakávaná testovacia chyba hypotézy je očakávaná chyba na náhodne vybranom príklade z rozdelenia P. Túto chybu budeme tiež označovať Err, a vypočítame ju nasledovne:

$$\mathrm{Err}_P(h) = \mathrm{E}_{(x,y)\sim P} \left[\mathrm{err}(h(x), y)\right].$$

Rozdelenie P bude obvykle jasné z kontextu, v takom prípade budeme skrátene písať $\mathrm{Err}(h)$.

1.2 Predpoklady o trénovacom algoritme

Doteraz sme neriešili, ako má vyzerať trénovací algoritmus L. Vo všeobecnosti, takýto algoritmus dostane na vstupe trénovaciu množinu T pozostávajúcu z t trénovacích príkladov a množinu hypotéz H, ktorá charakterizuje všetky možné "použiteľné" hypotézy. Výstupom takéhoto algoritmu má byť jedna funkcia $h \in H$, ktorú nazveme natrénovaná hypotéza.

Vo všeobecnosti, uvažovať vlastnosti ľubovoľného takého algoritmu L je veľmi ťažké. Intuitívne je jasné, že hypotéza, ktorú dá algoritmus na výstupe, by mala určite mať malú trénovaciu chybu pre trénovaciu množinu T. Kým pre niektoré množiny hypotéz H môže byť nájdenie takej hypotézy pomerne priamočiare (ako napríklad nájdenie funkcie minimalizujúcej L_2 trénovaciu chybu na množine všetkých lineárnych funkcií), v iných prípadoch to môže byť výpočtovo ťažký optimalizačný problém (napríklad NP-ťažký, či dokonca nevypočítateľný). Takisto, algoritmus L môže byť deterministický (pre tú istú trénovaciu množinu dá vždy ten istý výsledok), no môže byť aj pravdepodobnostný (používa náhodné čísla a preto sa výsledok môže pri rôznych behoch líšiť, aj napriek tomu, že na vstupe je tá istá trénovacia množina).

Aj keď úvahy o rôznych typoch trénovacích algoritmov sú zaujímavými problémami a tvoria podstatnú časť výskumu v teórii strojového učenia, pre jednoduchosť v tomto texte budeme od všetkých týchto problémov abstrahovať a budeme uvažovať jediný typ trénovacieho algoritmu: bude to algoritmus, ktorý nájde "najlepšiu" hypotézu z množiny hypotéz H, t.j. takú, ktorá minimalizuje trénovaciu chybu:

$$\hat{h}_T = \arg\min_{h \in H} \{\mathrm{Err}_T(h)\}.$$

Nebudeme pri tom uvažovať, akým spôsobom, či v akej časovej zložitosti sa to nášmu trénovaciemu algoritmu podarí. Takýto algoritmus sa nazýva aj *minimalizácia empirického rizika* (empirical risk minimization, alebo ERM).

Pre ERM algoritmus teraz vieme zadefinovať a neskôr aj analyzovať *očakávanú trénovaciu chybu* (ďalej OTrCh) a *očakávanú testovaciu chybu* (ďalej OTeCh). Uvedomme si, že v našom modeli je trénovacia množina T jednoducho množina t príkladov, ktoré sú nezávisle na sebe vyberané z pravdepodobnostnej distribúcie P. Preto T možno chápať ako náhodnú premennú a v tomto kontexte aj funkcia $\hat{h}_T$ je náhodnou premenou. *Očakávaná trénovacia chyba* preto bude definovaná ako nasledujúca stredná hodnota:

$$\mathrm{OTrCh} = E_T[\mathrm{Err}_T(\hat{h}_T)] = E_T\left[\frac{1}{t}\sum_{(x_i,y_i)\in T} \mathrm{err}(\hat{h}_T(x_i), y_i)\right].$$

OTrCh teda meria, ako dobre ERM algoritmus dokáže modelovať typickú trénovaciu množinu. Nás však bude zaujímať najmä to, ako dobre taký algoritmu funguje na nových príkladoch, ktoré neboli nutne obsiahnuté v trénovacej množine. Preto okrem trénovacej množiny T budeme uvažovať aj ďalší testovací príklad, takisto náhodne vybratý z pravdepodobnostnej distribúcie P. Pomocou tohto nového príkladu zadefinujeme *očakávanú testovaciu chybu* nasledovne:

$$\mathrm{OTeCh} = E_T E_{x,y}[\mathrm{err}(\hat{h}_T(x), y)].$$

Takto zadefinovaná chyba vystihuje celkovú chybu nakumulovanú v rámci celého učiaceho procesu: od chyby, ktorá môže vzniknúť výberom nereprezentatívnej trénovacej množiny T, cez chyby spôsobené výberom nevhodnej množiny hypotéz H, až po chyby vzniknuté pri záverečnom používaní natrénovanej hypotézy.

1.3 Teoretické limity

Intuitívne by sme očakávali, že ak by sme mali dostatočne veľkú množinu hypotéz H a pracovali by sme s dostatočne veľkým počtom trénovacích príkladov, tak sa nám eventuálne podarí docieliť, že OTeCh bude nulová. Problémom však je, že výsledkom trénovania musí byť funkcia, ktorá

pre každú hodnotu $x \in X$ vráti jednu konkrétnu cieľovú hodnotu $y \in Y$. Rozdelenie P ale môže pre jedno x pripúšťať viaceré hodnoty y, pre ktoré je pravdepodobnosť nenulová; napríklad P môže reprezentovať zašumené dáta.

Teda ani najlepšia možná hypotéza-funkcia nemusí mať nulovú chybu. Označme takúto hypotézu $h^{\square}$. Ak by sa nám podarilo vyjadriť jej testovaciu chybu, získali by sme akýsi ireducibilný komponen chyby, ktorý má každá hypotéza; môžeme sa snažiť znížiť iba tú časť chyby, ktorá je oproti tomuto "navyše". Uvažujúc L_2 chybu, z definície

$$h^{\square} = \arg\min_h \{\mathrm{Err}(h)\} = \arg\min_h \{\mathrm{E}_{x,y}\left[(h(x) - y)^2\right]\}.$$

Chybu ľubovoľnej hypotézy h vieme upraviť nasledovne:

$$\begin{aligned} \mathrm{Err}(h) &= \mathrm{E}_{x,y}\left[(h(x) - y)^2\right] && (1)\\ &= \mathrm{E}_x\left[\mathrm{E}_{y|x}\left[(h(x) - y)^2\right]\right]. && (2) \end{aligned}$$

Pozrime sa na vnútornú strednú hodnotu ($\mathrm{E}_{y|x}$). V nej je x konštanta, a teda aj $h(x) = c$ je konštanta. Nie je ťažké vidieť (napríklad zderivovaním), že minimum dosiahneme pre $c = \mathrm{E}_{y|x}[y]$. Takže

$$h^{\square}(x) = \mathrm{E}_{y|x}[y],$$

a (ireducibilná) očakávaná testovacia chyba najlepšej možnej hypotézy teda je

$$\mathrm{Err}(h^{\square}) = \mathrm{E}_x\left[\mathrm{E}_{y|x}\left[(y - \mathrm{E}[y])^2\right]\right] = \mathrm{E}_x\left[\underset{y|x}{\mathrm{Var}}(y)\right].$$

2 Hľadanie rovnováhy medzi výchylkou a rozptylom

V tejto časti sa podrobnejšie pozrieme na to, ako závisia vyššie uvedené metriky (t.j. OTeCh a OTrCh) od veľkosti trénovacej množiny t a od zložitosti množiny hypotéz H. V celej časti budeme predpokladať, že úloha je regresného charakteru a chyba sa meria pomocou funkcie L_2.

2.1 Rozklad OTeCh na výchylku a rozptyl

V tomto odseku si ukážeme zaujímavý výsledok, ktorý nám za určitých predpokladov umožňuje vyjadriť chyby pomocou iných, jasnejších veličín: tzv. *výchylky* a *rozptylu*. Označme najlepšiu hypotézu z množiny H ako $h^{\star}$, teda

$$h^{\star} = \arg\min_{h \in H} \left(\mathrm{Err}(h)\right).$$

Budeme upravovať výraz reprezentujúci OTeCh:

$$\begin{aligned} \mathrm{OTeCh} &= \mathrm{E}_T\left[\mathrm{Err}(\hat{h}_T)\right] && (3)\\ &= \mathrm{E}_T\left[\mathrm{E}_{x,y}\left[(\hat{h}_T(x) - y)^2\right]\right] && (4)\\ &= \mathrm{E}_T\left[\mathrm{E}_{x,y}\left[\left((\hat{h}_T(x) - h^{\star}(x)) + (h^{\star}(x) - y)\right)^2\right]\right] && (5)\\ &= \mathrm{E}_T\left[\mathrm{E}_{x,y}\left[(\hat{h}_T(x) - h^{\star}(x))^2\right]\right] + \mathrm{E}_T\left[\mathrm{E}_{x,y}\left[(h^{\star}(x) - y)^2\right]\right] && (6) \end{aligned}$$

Posledná rovnosť vychádza z netriviálneho technického kroku, ktorý si vyžaduje dodatočné predpoklady. Tieto technické detaily však prenecháme na kapitolu 2.2. Druhý zo sčítancov sa dá ešte

zjednodušiť. Kedže $h^\star$ ani y nezávisia od trénovacích dát, môžeme sa zbaviť vonkajšej strednej hodnoty. Dostávame tak výslednú rovnosť

$$\text{OTeCh} = \underbrace{\mathrm{E}_T\left[\mathrm{E}_{x,y}\left[(\hat{h}_T(x) - h^\star(x))^2\right]\right]}_{\text{rozptyl}} + \underbrace{\mathrm{E}_{x,y}\left[(h^\star(x) - y)^2\right]}_{\text{výchylka}}. \tag{7}$$

Prvý zo sčítancov budeme volať *rozptyl* (angl. variance). Trénovací algoritmus s malým rozptylom vracia funkcie, ktoré sú blízko optima v množine H. Tým, že mu zväčšíme množinu trénovacích dát, si veľmi neprilepšíme. Naopak, algoritmus s veľkým rozptylom vracia funkcie ďaleko od optima, vieme sa teda k optimu priblížiť tým, že zväčšíme množstvo trénovacích dát.

Druhý zo sčítancov budeme volať *výchylka* (angl. bias). Vyjadruje chybu, ktorá je spôsobená tým, že sa náš algoritmus obmedzil na nejakú konkrétnu množinu hypotéz H. Čím väčšia množina hypotéz, tým menšia výchylka (nakoľko $h^\star$ je najlepšia hypotéza v množine H, jej zväčšením si môžeme iba prilepšiť). Zložitejšia množina hypotéz sa ale ľahšie "napasuje" na ľubovoľné trénovacie dáta. To zvyšuje riziko toho, že výsledná hypotéza bude špecifická pre trénovacie dáta a nebude schopná generalizácie. Teda čím zložitejšia množina hypotéz, tým väčší rozptyl.

Výchylku vieme upraviť ďalej, čo nám umožní vyjadriť vzťah medzi OTrCh a OTeCh. Všimnime si, že hypotéza $h^\star$ ani y nezávisia od trénovacej množiny T. Z ich pohľadu sú teda testovacie dáta x, y a trénovacie dáta $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)\}$ nerozlíšiteľné. Takže na meranie chyby $h^\star$ môžeme použiť aj trénovacie dáta, berúc v úvahu ich náhodný výber:

$$\text{výchylka} = \mathrm{E}_T\left[\frac{1}{t}\sum_{(x_i,y_i)\in T}(h^\star(x_i) - y_i)^2\right] \tag{8}$$

$$= \mathrm{E}_T\left[\frac{1}{t}\sum_{(x_i,y_i)\in T}\Big((h^\star(x_i) - \hat{h}_T(x_i)) + (\hat{h}_T(x_i) - y_i)\Big)^2\right] \tag{9}$$

Použitím technického kroku obdobného k predchádzajúcemu dostaneme:

$$\text{výchylka} = \underbrace{\mathrm{E}_T\left[\frac{1}{t}\sum_{(x_i,y_i)\in T}(h^\star(x_i) - \hat{h}_T(x_i))^2\right]}_{\text{trénovací rozptyl}} + \underbrace{\mathrm{E}_T\left[\frac{1}{t}\sum_{(x_i,y_i)\in T}(\hat{h}_T(x_i) - y_i)^2\right]}_{\text{OTrCh}} \tag{10}$$

Prvý zo sčítancov budeme volať *trénovací rozptyl*. Uvedomme si, že pre ľubovoľné trénovacie dáta T platí $\mathrm{Err}_T(\hat{h}_T) \leq \mathrm{Err}_T(h^\star)$, nakoľko $\hat{h}_T$ je optimálna hypotéza pre danú množinu trénovacích dát. Hypotéza $h^\star$ síce je najlepšia pre H, trénovacie dáta sú ale len malá vzorka z H. Trénovací rozptyl teda môžeme chápať ako mieru toho, akej veľkej chyby sa trénovací algoritmus dopustí z dôvodu toho, že dostal obmedzený počet trénovacích príkladov.

Druhý zo sčítancov je OTrChAlg. Je to stredná hodnota chyby, ktorej sa dopustí výstup z algoritmu $\hat{h}_T$ na tých istých dátach, pomocou ktorých sme $\hat{h}_T$ zostrojili.

Z uvedených vzťahov ľahko vidíme, že platí:

$$\text{OTrCh} \leq \text{výchylka} \leq \text{OTeCh}.$$

Na konkrétnych trénovacích dátach ale nemusí platiť, že trénovacia chyba je menšia ako testovacia chyba: mohli sme si (síce s malou pravdepodobnosťou, ale predsa) vytiahnuť zlé trénovacie dáta, na ktorých sa žiadnej hypotéze z H nedarí. Hlbšie dôsledky rozkladov (7) a (10) si rozoberieme v kapitole 2.3.

Poznámka 2. Význam pojmov výchylka a rozptyl, tak ako sme ich používali v tejto kapitole, sa odlišuje od toho, ako sa tieto pojmy používajú v štatistike. Slovo výchylka tu používame v zmysle "časť chyby, ktorá je ireducibilná vzhľadom ku danej množine hypotéz" a časť komunity ho volá aj *chyba aproximácie* (angl. *approximation error*), aby sa predišlo zámene s rovnako nazvaným štatistickým pojmom. Podobne, slovo rozptyl tu používame ako "časť chyby, ktorá vyplýva z nedokonalého trénovacieho algoritmu (konkrétne ERM na fixnom počte trénovacích príkladov)" a zvykne sa nazývať aj *chyba odhadu* (angl. *estimation error*). Výsledok prezentovaný ako rovnica (7) tak môžeme nájsť aj pod názvom *rozklad na chybu aproximácie a chybu odhadu* (angl. *approximation–estimation error decomposition* resp. *trade-off between approximation and decomposition error.*)

Napriek tomu sa tvrdenie (7) a (10) a pojmy zadefinované ako výchylka a rozptyl v tejto kapitole bežne používajú v kontexte záverov prezentovaných v kapitole 2.3, preto sa aj tu pridŕžame tejto interpretácie a mierne nešťastnej terminológie. Na druhej strane, existuje aj rozklad OTeCh na výchylku a rozptyl v štatistickom slova zmysle; tento výsledok, ako aj jeho dôsledky, si zbežne ukážeme v kapitole 2.7, tento výsledok nám však neumožňuje odvodiť závery prezentované v kapitole 2.3.

2.2 *** Technické detaily

V tejto časti dokážeme tvrdenie, ktoré sme použili pri odvodení vzťahu (6).

Pripomeňme si, že $h^{\square}(x)$ je cieľová funkcia, ktorá je definovaná ako $h^{\square}(x) = \mathrm{E}_{y|x}[y]$ a táto cieľová funkcia nemusí byť súčasťou triedy hypotéz H. Funkcia $h^{\star}(x)$ je taká funkcia z H, ktorá má najmenšiu možnú očakávanú chybu, t.j. $h^{\star} = \arg\min_{h\in H}\left(\mathrm{E}_{x,y}[(h(x)-y)^2]\right)$. Túto definíciu vieme vztiahnuť ku funkcii $h^{\square}$ pomocou nasledujúcej lemy.

Lema 1. *Pre funkciu $h^{\star}(x)$ platí:*

$$h^{\star} = \arg\min_{h\in H}\left(\mathrm{E}_{x}[(h(x)-h^{\square}(x))^2]\right).$$

Dôkaz. Úpravou výrazu z definície h^{*} získame:

$$\begin{aligned}
\mathrm{E}_{x,y}[(h(x)-y)^2] &= \mathrm{E}_{x,y}[((h(x)-h^{\square}(x))+(h^{\square}(x)-y))^2] \\
&= \mathrm{E}_{x,y}[\underbrace{(h(x)-h^{\square}(x))^2}_{\text{nezávisí od } y}] + \mathrm{E}_{x,y}[(h^{\square}(x)-y)^2] \\
&\quad +2\mathrm{E}_{x,y}[((h(x)-h^{\square}(x)(h^{\square}(x)-y))] \\
&= \mathrm{E}_{x}[(h(x)-h^{\square}(x))^2] + \mathrm{E}_{x,y}[(h^{\square}(x)-y)^2] \\
&\quad +2\mathrm{E}_{x}[(h(x)-h^{\square}(x))(h^{\square}(x)-\mathrm{E}_{y|x}[y])]
\end{aligned}$$

Druhý sčítanec je konštanta vzhľadom na h, preto ho pri hľadaní minima nemusíme uvažovať. Z definície $h^{\square}(x) = \mathrm{E}_{y|x}[y]$, preto tretí sčítanec bude nulový. Zostáva nám teda:

$$h^{\star} = \arg\min_{h\in H}\left(\mathrm{E}_{x}[(h(x)-h^{\square}(x))^2]\right),$$

čo bolo treba dokázať. □

Z predchádzajúcej lemy teda vyplýva, že funkcia $h^{\star}$ je priemetom funkcie $h^{\square}$ do H. Pre zjednodušenie si v tejto kapitole zavedieme označenie $\langle f, g\rangle = \mathrm{E}_{x}[f(x)g(x)]$. Takisto si zavedieme pojem *vzdialenosti funkcií* $\mathrm{dist}^2(f,g) = \mathrm{E}_{x}[(f(x)-g(x))^2] = \langle f(x)-g(x), f(x)-g(x)\rangle$. Funkcia $h^{\star}(x)$ je tak vlastne definovaná ako funkcia z H s najmenšou vzdialenosťou od $h^{\square}$.

Lema 2. *Nech množina hypotéz H je uzavretá na lineárne kombinácie. Potom pre ľubovoľnú funkciu $h \in H$ platí:*

$$\mathrm{E}_x\left[h(x)(h^\star(x) - h^\square(x))\right] = \langle h, h^\star - h^\square\rangle = 0.$$

Dôkaz. Nech h je ľubovoľná funkcia z H. Uvažujme triedu funkcií F, ktoré sú tvaru $f_\Delta = h^\star + \Delta h$, kde Δ je ľubovoľné reálne číslo. Všimnime si, že $F \subseteq H$, lebo H je uzavretá na lineárne kombinácie.

Vzhľadom ku leme 1, ktorá charaktizuje $h^\star$, musí byť zo všetkých funkcií z F najbližšou k $h^\square$ funkcia $f_0 = h^\star$, z čoho vyplýva, že derivácia funkcie $D(\Delta) = \mathrm{dist}^2(f_\Delta, h^\square)$ musí byť v bode $\Delta = 0$ nulová.

Upravme funkciu $D(\Delta)$:

$$\begin{aligned} D(\Delta) &= \mathrm{dist}^2(f_\Delta, h^\square) && (11)\\ &= \langle h^\star + \Delta h - h^\square, h^\star + \Delta h - h^\square\rangle && (12)\\ &= \langle (h^\star - h^\square) + \Delta h, (h^\star - h^\square) + \Delta h\rangle && (13)\\ &= \langle h^\star - h^\square, h^\star - h^\square\rangle + \Delta^2\langle h, h\rangle + 2\Delta\langle h^\star - h^\square, h\rangle && (14)\end{aligned}$$

Vypočítame teraz deriváciu $D(\Delta)$ podľa Δ:

$$\begin{aligned} D'(\Delta) &= \frac{d\langle h^\star - h^\square, h^\star - h^\square\rangle}{d\Delta} + \frac{d\Delta^2\langle h, h\rangle}{d\Delta} + \frac{d(2\Delta\langle h^\star - h^\square, h\rangle)}{d\Delta} && (15)\\ &= 2\Delta\langle h, h\rangle + 2\langle h^\star - h^\square, h\rangle && (16)\end{aligned}$$

Teraz položíme $\Delta = 0$, a keďže derivácia musí byť v tomto bode nulová, dostávame rovnicu:

$$0 = d'(0) = 2\langle h^\star - h^\square, h\rangle, \qquad (17)$$

čo bolo treba dokázať. □

Vráťme sa teraz k rovnici (6). Chceme ukázať, že

$$\mathrm{E}_T\left[\mathrm{E}_{x,y}\left[\left((\hat{h}_T(x) - h^\star(x)) + (h^\star(x) - y)\right)^2\right]\right] = \mathrm{E}_T\left[\mathrm{E}_{x,y}\left[(\hat{h}_T(x) - h^\star(x))^2\right]\right] + \mathrm{E}_T\left[\mathrm{E}_{x,y}\left[(h^\star(x) - y)^2\right]\right],$$

čiže potrebujeme ukázať, že

$$\mathrm{E}_T\left[\mathrm{E}_{x,y}\left[(\hat{h}_T(x) - h^\star(x)) \cdot (h^\star(x) - y)\right]\right] = 0$$

Platí:

$$\begin{aligned} \mathrm{E}_{x,y}\left[(\hat{h}_T(x) - h^\star(x)) \cdot (h^\star(x) - y)\right] &= \mathrm{E}_{x,y}\left[(\hat{h}_T(x) - h^\star(x)) \cdot (h^\star(x) - h^\square(x) + h^\square(x) - y)\right]\\ &= \mathrm{E}_{x,y}[\underbrace{(\hat{h}_T(x) - h^\star(x)) \cdot (h^\star(x) - h^\square(x))}_{\text{nezávisí od } y}]\\ &\quad + \mathrm{E}_{x,y}[(\hat{h}_T(x) - h^\star(x)) \cdot (h^\square(x) - y)]\\ &= \mathrm{E}_x[(\hat{h}_T(x) - h^\star(x)) \cdot (h^\star(x) - h^\square(x))]\\ &\quad + \mathrm{E}_x[(\hat{h}_T(x) - h^\star(x)) \cdot (h^\square(x) - \mathrm{E}_{y|x}[y])].\end{aligned}$$

Keďže z definície $h^\square(x) = \mathrm{E}_{y|x}[y]$, druhý sčítanec sa vynuluje a dostávame:

$$\begin{aligned} \mathrm{E}_{x,y}\left[(\hat{h}_T(x) - h^\star(x)) \cdot (h^\star(x) - y)\right] &= \mathrm{E}_x[(\hat{h}_T(x) - h^\star(x)) \cdot (h^\star(x) - h^\square(x))]\\ &= \langle \hat{h}_T - h^\star, h^\star - h^\square\rangle.\end{aligned}$$

Uvedomme si, že funkcia $h = \hat{h}_T - h^\star$ je lineárnou kombináciou dvoch funkcií z H, a preto tiež patrí do H. Preto môžeme použiť Lemu 2 a dostávame pre ľubovoľnú trénovaciu množinu T:

$$\mathrm{E}_{x,y}\left[(\hat{h}_T(x) - h^\star(x)) \cdot (h^\star(x) - y)\right] = \langle h, h^\star - h^\square \rangle = 0,$$

a teda aj:

$$\mathrm{E}_T\mathrm{E}_{x,y}\left[(\hat{h}_T(x) - h^\star(x)) \cdot (h^\star(x) - y)\right] = 0,$$

čo sme chceli dokázať.

Poznámka 3. Označenie $\langle f, g \rangle$ sa vizuálne nepodobá na skalárny súčin vektorov náhodou. V skutočnosti je toto označenie priamočiarym rozšírením skalárneho súčinu vektorov na funkcie a množstvo vlastností skalárneho súčinu vektorov možno analogicky preniesť aj na takto definované skalárne súčiny funkcií. Analogicky ku kolmosti vektorov možno zadefinovať aj kolmosť funkcií $f \perp g$, ak $\langle f, g \rangle = 0$. Lema 2 je v takom prípade ekvivalentom tvrdenia o kolmosti v prípade projekcií vektorov do podpriestoru. Napríklad pre priamku p a bod x platí, že ak y je najbližší bod priamky p ku x, tak vektor $x - y$ je kolmý na p. V tomto texte je H analógom priamky p, $h^\star$ je analógom bodu y a $h^\square$ je analógom bodu x.

Poznámka 4. Všetky tvrdenia doposiaľ predpokladali, že v množine hypotéz H existuje funkcia h^* najbližšia ku H. Pri niektorých množinách hypotéz však tento predpoklad nemusí byť splnený. Môže sa nám totiž stať, že vieme zostrojiť nekonečnú postupnosť funkcií, kde každá ďalšia funkcia je bližšie ku $h^\square$, no táto postupnosť funkcií konverguje ku funkcii mimo H. Aby sme zaručili, že takýto prípad nenastane, musíme do predpokladov zahrnúť aj požiadavku, že H je uzavreté na limity.

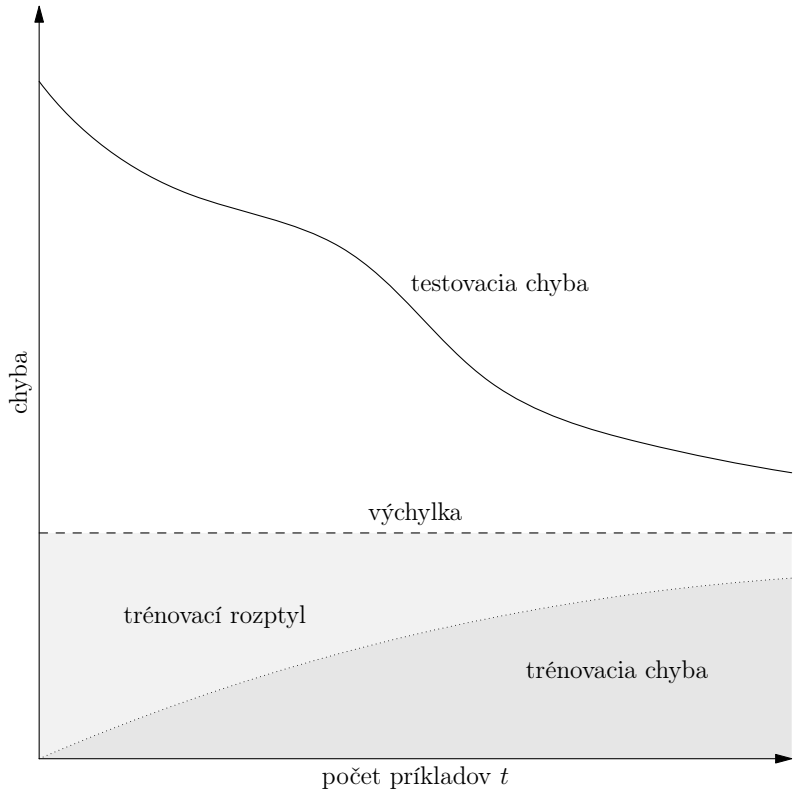
2.3 Krivky učenia, podučenie a preučenie

Na základe vzťahov (7) a (10) vieme graficky znázorniť, ako sa zhruba správajú OTeCh, rozptyl, výchylka, trénovací rozptyl a OTrCh v závislosti od veľkosti trénovacej množiny (obrázok 1) a od zložitosti množiny hypotéz (obrázok 2). Tieto obrázky sa nazývajú aj *krivky učenia*.

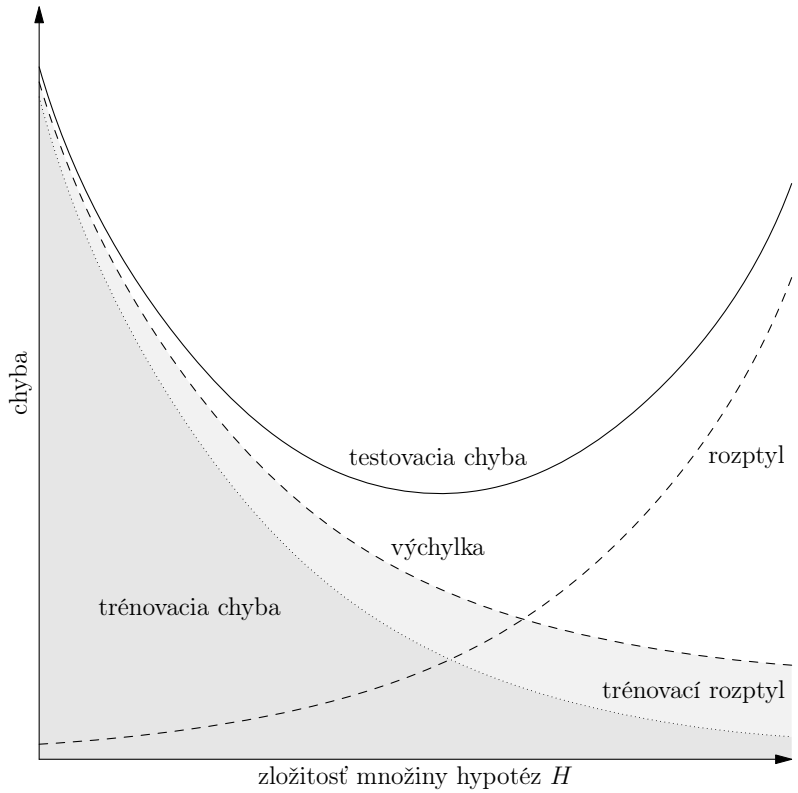
Obrázok 1 vychádza z toho, že ak zvyšujeme počet trénovacích príkladov, tak trénovací rozptyl aj rozptyl sa budú znižovať (všimnite si, že výchylka závisí len od zložitosti množiny hypotéz, ale nie od počtu trénovacích príkladov). Na základe vzťahu (10) vieme teda povedať, že OTeCh sa bude so zvyšujúcim počtom trénovacích príkladov približovať k výchylke zhora, kým OTrCh sa bude postupne približovať k výchylke zdola. Preto pri fixnej množine hypotéz nie je najzaujímavejším faktorom absolútna trénovacia či testovacia chyba (ktorá obsahuje aj výchylku, ktorá je pri fixnej množine hypotéz ireducibilná), ale rozdiel medzi trénovacou a testovacou chybou.

Obrázok 2 ilustruje situácie, ak máme fixné trénovacie dáta T a môžeme meniť zložitosť množiny hypotéz H. Obrázok demonštruje, že kľúčom ku dobre vyladenému modelu je nájdenie rovnováhy medzi výchylkou a rozptylom (angl. *bias-variance tradeoff*), ktoré dohromady dávajú testovaciu chybu. Zložité H bude mať malú výchylku ale veľký rozptyl, čo vedie k tzv. *preučeniu* (angl. *overfitting*. V takom prípade je potrebné zvýšiť množstvo trénovacích príkladov alebo zjednodušiť množinu hypotéz. Jednoduché H bude mať malý rozptyl, ale veľkú výchylku a nastane tzv. *podučenie* (angl. *underfitting*). V takom prípade nám zvýšenie počtu trénovacích príkladov nepomôže, dominujúcim faktorom v testovacej chybe je totiž malá reprezentačná sila množiny hypotéz H.

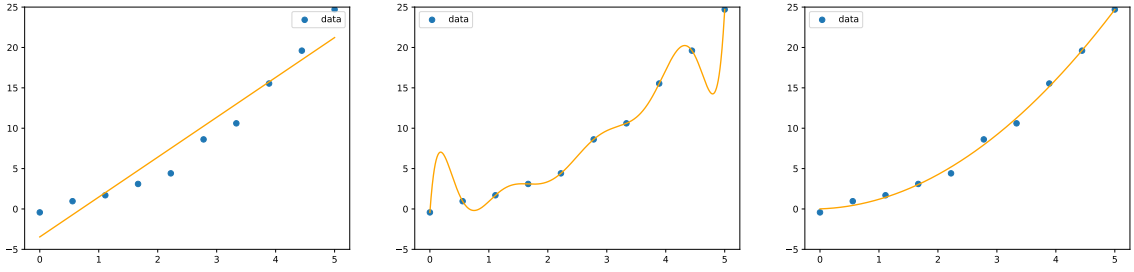
Na obrázku 3 ilustrujeme oba koncepty. Úlohou je modelovať kvadratickú funkciu, ku ktorej sme pridali malé množstvo šumu. Rozdelenie P teda vracia nejaké x (napríklad z intervalu $\langle 0, 10 \rangle$) a $y = x^2 + \varepsilon$, kde ε je zvolené náhodne z intervalu $\langle -1, 1 \rangle$. Ak za množinu hypotéz zvolíme lineárne funkcie, vďaka ich jednoduchosti už pri malých trénovacích množinách bude trénovacia chyba blízko očakávanej testovacej chyby (nízky rozptyl). Za to ale budú všetky chyby vysoké (vysoká výchylka). Ak za množinu hypotéz zvolíme polynómy nejakého vysokého stupňa,



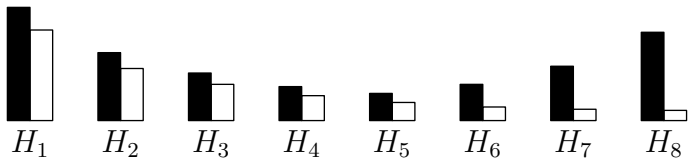
Obr. 1: Závislosť chyby algoritmu od počtu trénovacích príkladov t.



Obr. 2: Závislosť chyby algoritmu od zložitosti množiny hypotéz H.



Obr. 3: Podučenie, preučenie, akurát.



Obr. 4: Testovacie (čierne) a trénovacie (biele) chyby v čím ďalej, tým zložitejších množinách hypotéz H.

ľahko nájdeme polynóm presne prechádzajúci cez trénovacie dáta (nízka trénovacia chyba), avšak mimo nich bude dávať výsledky úplne mimo (vysoká testovacia chyba, a teda vysoký rozptyl).

2.4 Regularizácia

Predstavme si, že máme na výber z viacerých množín hypotéz $H_1, H_2, \ldots$, čím ďalej tým zložitejších. Teda $H_1 \subseteq H_2 \subseteq H_3 \subseteq \ldots$. Ak by sme si graficky znázornili testovacie a trénovacie chyby najlepších hypotéz z jednotlivých množín, vyzeralo by to zhruba ako na obrázku 4.

Z ktorej množiny hypotéz chceme vybrať? Pri trénovaní sa snažíme nájsť hypotézu h, ktorá minimalizuje chybu na trénovacích dátach. Táto chyba sa ale od očakávanej testovacej chyby líši zhruba o $\epsilon_{\text{est.}} + \epsilon^T_{\text{est.}}$ (v očakávanom prípade).

V prístupe zvanom *regularizácia* do minimalizovaného výrazu umelo pridáme *pokutu*, ktorá aproximuje $\epsilon_{\text{est.}} + \epsilon^T_{\text{est.}}$. Tento člen označíme $\Lambda(h)$. Z čím zložitejšej množiny hypotéza h je, tým väčšia bude pokuta. Výstupom algoritmu potom je

$$\hat{h}_T = \underset{h \in H_1 \cup H_2 \cup \ldots}{\arg\min} \left(\text{Err}_T(h) + \Lambda(h) \right).$$

Uvedomme si, že vrámci jednej množiny H_i ostáva ako najlepšia hypotéza stále tá istá, ako pred zavedením pokuty. V jednej množine sú totiž všetky hypotézy penalizované rovnako, nerobí to teda rozdiel. Penalizácia nám ale umožňuje "férovejšie" porovnávať hypotézy z rôznych množín, nakoľko bez pokuty by na tom boli (neprávom) lepšie zložitejšie hypotézy.

Množiny H_i nemusia byť explicitné, môžu byť implicitne skryté v tom, aký tvar má výraz $\Lambda(h)$. Do jednej množiny patria tie hypotézy, ktoré majú rovnakú penalizáciu.

Vo všetkých prípadoch je pokuta parametrizovaná reálnym parametrom λ hovoriacim, ako veľké pokuty chceme udeľovať. Veľké λ (v porovnaní s trénovacou chybou) nám hovorí, že sa snažíme hlavne dosiahnuť jednoduché hypotézy; s malým λ zas kladieme dôraz na hypotézy s menšou trénovacou chybou.

Uvedieme si niekoľko príkladov výrazov, ktoré sú bežne používané ako pokuta. Budeme predpokladať, že celá množina hypotéz, z ktorej vyberáme (tj. $H_1 \cup H_2 \cup \ldots$) je množina lineárnych funkcii $\mathbb{R}^n \to \mathbb{R}$. Hypotézy majú teda tvar

$$h(x) = a_1 x_1 + a_2 x_2 + \ldots + a_n x_n.$$

- L_2 regularizácia (známa aj ako *ridge regression*).

$$\Lambda(h) = \lambda \cdot \|(a_1, a_2, \ldots, a_n)\|^2 = \lambda \cdot (a_1^2 + a_2^2 + \ldots + a_n^2)$$

  Táto pokuta "tlačí" váhy nepotrebných atribútov do nuly, pričom väčšie váhy tlačí viac. Takže čím dôležitejší atribút, tým väčšiu váhu si môže dovoliť mať.

- L_1 regularizácia (známa aj ako *lasso*).

$$\Lambda(h) = \lambda \cdot (|a_1| + |a_2| + \ldots + |a_n|)$$

  Opäť "tlačíme" nepotrebné atribúty do nuly, pričom ale všetky váhy tlačíme rovnako. To nám vie vynulovať nepotrebné atribúty, čo nám vie znížiť výpočtové nároky: nemusíme pri výpočte uvažovať tie členy, ktoré majú nulový koeficient.

2.5 Holdout testing

V tomto prístupe si rozdelíme dostupné dáta na dve časti: trénovaciu množinu a *validačnú množinu*. Pomocou validačnej množiny budeme odhadovať testovacie chyby pre jednotlivé množiny hypotéz, na základe ktorých zistíme, ktorá množina hypotéz je pre náš problém najvhodnejšia. Konkrétnejšie:

1. Trénovaciu množinu použijeme na natrénovanie hypotéz z jednotlivých množín.

2. Ako odhad testovacej chyby jednotlivých hypotéz použijeme ich chybu na validačnej množine. Podľa týchto odhadov zistíme, ktorá množina hypotéz je pre náš problém najvhodnejšia.

3. Použijeme všetky dáta, ktoré máme k dispozícii (tj. z trénovacej aj validačnej množiny), na natrénovanie najlepšej možnej hypotézy. Berieme samozrejme do úvahy iba hypotézy z tej množiny hypotéz, ktorú sme identifikovali ako najvhodnejšiu. Výsledná hypotéza je výstupom algoritmu.

V kroku 2 je dôležité, aby bola validačná množina nezávislá od trénovacej. Prečo je to dôležité? Môžeme uvažovať extrémny prípad, keď je validačná množina totožná s trénovacou. Potom ale ako náš "odhad" dostaneme trénovaciu chybu, ktorá rozhodne nie je dobrým odhadom testovacej chyby. Nezávislosť nám teda zaručuje, že odhad získaný na validačnej množine je dobrý.

k-fold evaluation. Pri holdout testovaní je dôležité mať dobrý odhad testovacej chyby pre jednotlivé množiny hypotéz. Dát ale môže byť málo, a v takom prípade môže byť odhad nestabilný/nepresný. Môžeme ale experiment zopakovať niekoľkokrát: v každej iterácii teda zvolíme inú trénovaciu a inú validačnú množinu, a dostaneme iný odhad testovacej chyby. Keď tieto odhady spriemerujeme, dostaneme oveľa presnejší odhad, ako keby sme vykonali iba jednu iteráciu. V tomto konkrétnom prístupe je k iterácii, a množiny sa volia nasledovne: všetky dáta sa rozdelia na k zhruba rovnako veľkých a navzájom nezávislých množín $K_1, K_2, \ldots, K_k$. Následne, v iterácii i sa ako validačné dáta použije množina K_i. Všetko ostatné budú trénovacie dáta.

Leave-one-out. Ak chceme zmerať testovaciu chybu výstupnej hypotézy, musíme si na to rezervovať ďalšiu časť dát: *testovaciu množinu*. Tú nepoužívame ani pri trénovaní, ani pri validácii. Iba úplne na konci celého procesu na nej vypočítame chybu výslednej hypotézy.

2.6 Cvičenia

V nasledujúcich dvoch cvičeniach môžete predpokladať, že trénovací algoritmus vždy vráti nejakú funkciu (nemusí byť len jedna) s minimálnou chybou na trénovacích dátach.

1. Je rozumné predpokladať, že s väčším množstvom trénovacích dát sa nám bude testovacia chyba zmenšovať. Sú ale zostrojiteľné situácie, kedy tomu tak nie je. Nájdite jednu takú situáciu.

Konkrétne, nájdite takú množinu hypotéz H funkcii $\mathbb{R}^n \to \mathbb{R}$ a rozdelenie P, pre ktoré sa bude $\mathrm{Err}(\hat{h}_T)$ so zvyšujúcim sa počtom trénovacích chýb *zvyšovať*. Na množinu hypotéz je kladená jedna podmienka: pre každú možnú trénovaciu množinu T musí existovať hypotéza v H, ktorá minimalizuje trénovaciu chybu. (Teda vždy musí existovať minimum, môže ich byť ale viac. Pre všeobecné H vieme povedať iba to, že existuje infimum.)

2. Za určitých podmienok ale skutočne platí, že viac trénovacích dát nám vo veľkom merítku neuškodí. Nech množina hypotéz H je konečná a všetky jej funkcie ($\mathbb{R}^n \to \mathbb{R}$) sú ohraničené. Dokážte, že pre $t \to \infty$ sa bude OTeCh hypotézy $\hat{h}_T$ blížiť k OTeCh najlepšej možnej hypotézy $h^\star$. Inak zapísané, dokážte

$$\lim_{t\to\infty} \mathrm{E}_T \left[\mathrm{Err}(\hat{h}_T) - \mathrm{Err}(h^\star)\right] = 0.$$

3. Jednou výhodou L_2 regularizácie oproti L_1 regularizácie je, že sa ľahšie minimalizuje výsledný výraz. Ako príklad uvedieme lineárnu regresiu. V nej je hypotéza parametrizovaná stĺpcovým vektorom $\theta = (\theta_1, \ldots, \theta_n)^T$. Výstupom pre vstup $x = (x_1, \ldots, x_n)$ je $x \cdot \theta$.

Označme X maticu, ktorej riadkami sú vstupy jednotlivých trénovacích príkladov. Ďalej nech y je stĺpcový vektor cieľových výstupov na jednotlivých príkladoch. Je známe, že optimálnymi parametrami lineárnej hypotézy je taký stĺpcový vektor θ, ktorý je riešením rovnice

$$X^T X \cdot \theta = X^T y.$$

Dokáže, že keď k minimalizovanej hodnote pridáme pokutu vo forme $\lambda \cdot \|\theta\|^2$, tak sa optimálnymi parametrami stane θ riešiaca rovnicu

$$(X^T X + \lambda I) \cdot \theta = X^T y.$$

Rozmyslite si tiež, že takéto explicitné vyjadrenie nie je možné priamočiaro získať pre L_1 regularizáciu.

2.7 *** Rozklad na výchylku a rozptyl (ako štatistické pojmy)

V literatúre pod názvom *bias-variance tradeoff* vystupuje odlišný výsledok, ako vyššie spomenutý bias-complexity tradeoff. Uvádzame ho, pretože sa často zamieňajú a je v tom zmätok. Pokúsime sa vyjasniť, kde sú rozdiely medzi týmito dvomi výsledkami.

Veta 3. *Zamerajme sa na jeden konkrétny vstup x, a merajme chybu výslednej hypotézy $\hat{h}_T$ iba na tomto vstupe. (Stále ale môžeme dostať rôzne y.) Na meranie chyby použijeme kvadratickú odchýlku. Označme očakávanú hodnotu tejto chyby (cez všetky možné trénovacie množiny T a výstupy y) ako* $\mathrm{Err}_{|x=x}(L)$. *Tvrdíme, že sa dá vyjadriť nasledovne:*

$$\mathrm{Err}_{|x=x}(L) = \underbrace{\operatorname{Var}_T\left(\hat{h}_T(x)\right)}_{\text{rozptyl}} + \underbrace{\left(\mathrm{E}_T\left[\hat{h}_T(x)\right] - h^{\square}(x)\right)^2}_{\text{výchylka}^2} + \underbrace{\mathrm{E}_y\left[\varepsilon^2\right]}_{\text{šum}},$$

kde $\varepsilon := y - h^{\square}(x)$.

Zastavme sa najprv nad tým, ako toto tvrdenie interpretovať. Pre každú možnú vzorku trénovacích dát T dostaneme nejakú inú hypotézu $\hat{h}_T$. Rozptyl je nízky práve vtedy, keď budú všetky tieto hypotézy dávať podobné výsledky. Ak je vysoký, znamená to, že výsledná hypotéza je veľmi citlivá na trénovacie dáta; oplatí sa preto zväčšiť ich množstvo. (Ak chceme byť veľmi skeptický, nič nezaručuje, že zväčšením trénovacej množiny sa rozptyl zníži. Možno existuje iný dôvod, kvôli ktorému je vysoký; takmer určite sa dajú skonštruovať takéto umelé protipríklady.)

Keď už uvažujeme všetky možné výsledné hypotézy $\hat{h}_T$, môžme sa pozrieť na to, aký je ich "priemerný odhad": ak by sme spriemerovali všetky ich výsledky, čo by sme dostali? Presne $\mathrm{E}_T[\hat{h}_T(x)]$; výchylka-na-druhú potom meria, o koľko sa takáto priemerná hypotéza líši od najlepšej možnej funkcie $h^\square$. Ak je výchylka vysoká ale variancia nízka, znamená to, že takmer všetky $\hat{h}_T$ majú problém na vstupe x, treba teda zvážiť voľbu zložitejšej množiny hypotéz.

Nakoniec, šum zodpovedá ireducibilnej chybe, ktorú bude mať každá hypotéza. Je to v dôsledku toho, že jednému x môže pripadať viacero rôznych y. Presne túto chybu nadobúda najlepšia hypotéza-funkcia $h^\square$ (pre ktorú sú rozptyl aj výchylka-na-druhú nulové).

Dôkaz. Upravujme pôvodný výraz.

$$\begin{aligned}
\mathrm{Err}_{|x=x}(L) &= \mathrm{E}_{T,y}\left[(\hat{h}_T(x) - y)^2\right] && (18)\\
&= \mathrm{E}_{T,y}\left[((\hat{h}_T(x) - h^\square(x)) - \varepsilon)^2\right] && (19)\\
&= \mathrm{E}_{T,y}\left[(\hat{h}_T(x) - h^\square(x))^2 + \varepsilon^2 - 2\cdot\varepsilon\cdot(\hat{h}_T(x) - h^\square(x))\right] && (20)\\
&= \mathrm{E}_T\left[(\hat{h}_T(x) - h^\square(x))^2\right] + \mathrm{E}_y\left[\varepsilon^2\right] - \mathrm{E}_{T,y}\left[2\cdot\varepsilon\cdot(\hat{h}_T(x) - h^\square(x))\right] && (21)\\
&= \mathrm{E}_T\left[(\hat{h}_T(x) - h^\square(x))^2\right] + \mathrm{E}_y\left[\varepsilon^2\right] - 2\cdot\mathrm{E}_y\left[\varepsilon\right]\cdot\mathrm{E}_T\left[(\hat{h}_T(x) - h^\square(x))\right] && (22)\\
&= \mathrm{E}_T\left[(\hat{h}_T(x) - h^\square(x))^2\right] + \mathrm{E}_y\left[\varepsilon^2\right] && (23)
\end{aligned}$$

Výraz sme upravili, potom v kroku 20 roznásobili a v kroku 21 využili linearitu strednej hodnoty. Ďalej v kroku 22 sme využili, že stredná hodnota súčinu nezávislých premenných je súčinom stredných hodnôt tých premenných. Nakoniec v kroku 23 využívame $\mathrm{E}[\varepsilon] = 0$. Zamerajme sa ďalej na prvý sčítanec.

$$\begin{aligned}
&= \mathrm{E}_T\left[(\hat{h}_T(x) - h^\square(x))^2\right] && (24)\\
&= \mathrm{E}_T\left[\hat{h}_T(x)^2 + h^\square(x)^2 - 2\cdot\hat{h}_T(x)\cdot h^\square(x)\right] && (25)\\
&= \mathrm{E}_T\left[\hat{h}_T(x)^2\right] + \mathrm{E}_T\left[h^\square(x)^2\right] - \mathrm{E}_T\left[2\cdot\hat{h}_T(x)\cdot h^\square(x)\right] && (26)\\
&= \left(\operatorname*{Var}_T\left(\hat{h}_T(x)\right) + \mathrm{E}_T\left[\hat{h}_T(x)\right]^2\right) + \left(\operatorname*{Var}_T\left(h^\square(x)\right) + \mathrm{E}_T\left[h^\square(x)\right]^2\right) - \mathrm{E}_T\left[2\cdot\hat{h}_T(x)\cdot h^\square(x)\right] && (27)\\
& && (28)
\end{aligned}$$

V kroku 25 sme výraz roznásobili a potom v kroku 26 využili linearitu strednej hodnoty. V poslednom kroku (27) sme využili vzťah $\mathrm{Var}(a) = \mathrm{E}\left[a^2\right] - \mathrm{E}\left[a\right]^2$. Pokračujme ďalej v úpravách.

$$\begin{aligned}
&= \left(\operatorname*{Var}_T\left(\hat{h}_T(x)\right) + \mathrm{E}_T\left[\hat{h}_T(x)\right]^2\right) + h^\square(x)^2 - 2\cdot\mathrm{E}_T\left[\hat{h}_T(x)\right]\cdot h^\square(x) && (29)\\
&= \operatorname*{Var}_T\left(\hat{h}_T(x)\right) + \left(\mathrm{E}_T\left[\hat{h}_T(x)\right]^2 + h^\square(x)^2 - 2\cdot\mathrm{E}_T\left[\hat{h}_T(x)\right]\cdot h^\square(x)\right) && (30)\\
&= \operatorname*{Var}_T\left(\hat{h}_T(x)\right) + \left(\mathrm{E}_T\left[\hat{h}_T(x)\right] - h^\square(x)\right)^2 && (31)
\end{aligned}$$

V kroku 29 sme využili, že x je vopred dané, a teda $h^\square(x)$ je konštanta. Má teda nulový rozptyl a jeho stredná hodnota je identická s jeho hodnotou. Ďalej sme už len upravovali. Keď to celé dáme do jednej rovnice, dostaneme

$$\mathrm{Err}_{|x=x}(L) = \underbrace{\operatorname*{Var}_T\left(\hat{h}_T(x)\right)}_{\text{rozptyl}} + \underbrace{\left(\mathrm{E}_T\left[\hat{h}_T(x)\right] - h^\square(x)\right)^2}_{\text{výchylka}^2} + \underbrace{\mathrm{E}_y\left[\varepsilon^2\right]}_{\text{šum}}.$$

□

Literatúra