

Môžeš ku každému bodu napísať nejakú odpoveď:

1. Regresia. Lineárna regresia, riešenie pomocou normálnych rovníc a gradientovej metódy, generalizovaná lineárna regresia, regularizácia.
2. Neurónové siete. Rozdiely medzi logistickou regresiou a jednoduchým perceptrónom. Skryté vrstvy v neurónových sieťach. Konvolučné neurónové siete. Metóda spätnej propagácie.
3. Support vector machines. Základná formulácia, duálna formulácia. Kernelová metóda.
4. Hlasovacie schémy. Bagging a boosting. Aplikácia na rozhodovacie stromy. Náhodné lesy.
5. Teória strojového učenia. Matematický model strojového učenia. Výchylka a rozptyl. Holdout testing. PAC (probably approximately correct) učenie, VC (Vapnik-Cervonenkis) dimenzia
6. **Regresia: Lineárna regresia, riešenie pomocou normálnych rovníc a gradientovej metódy, generalizovaná lineárna regresia, regularizácia**
 - *Lineárna regresia* je metóda modelovania vzťahu medzi jednou alebo viacerými vstupnými premennými (nezávislé premenné) a výstupnou premennou (závislá premenná), pričom sa predpokladá lineárna závislosť. Cieľom je nájsť také parametre θ , aby bola chyba medzi predikovanými a skutočnými hodnotami čo najmenšia ^{[1] [2] [3]}.
 - *Normálne rovnice* poskytujú analytické riešenie lineárnej regresie: $\theta = (X^T X)^{-1} X^T y$, kde X je matica vstupov a y vektor výstupov. Toto riešenie je vhodné pre malé až stredne veľké úlohy, kde nie je problém s inverziou matice ^{[1] [3]}.
 - *Gradientová metóda* (gradient descent) je iteratívny algoritmus, ktorý postupne upravuje parametre v smere najväčšieho znižovania chyby. Je vhodný pre veľké datasety, kde je výpočet normálnych rovníc neefektívny alebo nemožný ^{[1] [4]}.
 - *Generalizovaná lineárna regresia* rozširuje lineárnu regresiu pomocou báзовých (často nelineárnych) funkcií, ktoré umožňujú modelovať aj nelineárne vzťahy medzi vstupmi a výstupom ^{[3] [5]}.
 - *Regularizácia* (napr. Lasso, Ridge) pridáva do chybovej funkcie penalizáciu za veľké hodnoty parametrov, čím znižuje riziko preučenia a zvyšuje robustnosť modelu, najmä pri veľkom počte vstupných premenných alebo šume v dátach ^{[2] [3]}.
2. **Neurónové siete: rozdiely medzi logistickou regresiou a perceptrónom, skryté vrstvy, konvolučné siete, spätná propagácia**
 - *Logistická regresia* je jednoduchý model na binárnu klasifikáciu, ktorý používa sigmoidálnu aktiváciu a výstup interpretuje ako pravdepodobnosť. *Perceptrón* je základný model neurónovej siete, ktorý tiež klasifikuje vstupy, ale používa prahovú (step) funkciu a nie je schopný riešiť nelineárne separovateľné úlohy ^[6].
 - *Skryté vrstvy* v neurónových sieťach umožňujú modelovať zložité nelineárne vzťahy medzi vstupom a výstupom. Každá skrytá vrstva transformuje dáta do vyššej reprezentácie, čo

zvyšuje schopnosť siete učiť sa komplexné funkcie. Viac skrytých vrstiev znamená hlboké učenie (deep learning) [17].

- *Konvolučné neurónové siete* (CNN) sú špeciálne architektúry určené na spracovanie dát s priestorovou štruktúrou, najmä obrázkov. Používajú konvolučné a subsamplingové (pooling) vrstvy, ktoré extrahujú príznaky z lokálnych oblastí vstupu, čím sú veľmi efektívne pri rozpoznávaní obrazových vzorov [8].
- *Spätná propagácia* (backpropagation) je algoritmus na trénovanie viacvrstvových neurónových sietí, ktorý využíva gradientovú metódu na úpravu váh siete podľa chyby na výstupe. Umožňuje efektívne učiť aj hlboké siete [5].

3. **Support Vector Machines (SVM): základná a duálna formulácia, kernelová metóda**

- *Základná formulácia* SVM hľadá optimálnu deliacu hyperplochu medzi dvoma triedami tak, aby maximálny počet bodov bolo správne rozdelených a zároveň bola maximalizovaná vzdialenosť (margin) medzi najbližšími bodmi oboch tried (tzv. podporné vektory) [9] [10].
- *Duálna formulácia* SVM preformuluje optimalizačný problém tak, že riešenie je vyjadrené ako lineárna kombinácia trénovacích príkladov, čo umožňuje efektívne využitie kernelových funkcií a riešiť aj nelineárne separovateľné úlohy [11] [10].
- *Kernelová metóda* (kernel trick) umožňuje implicitne transformovať dáta do vyššej dimenzie, kde sú lineárne separovateľné, bez explicitného výpočtu transformácie. Najčastejšie používané kernely sú RBF (gaussovský), polynomiálny a sigmoidálny [9] [11] [10].

4. **Hlasovacie schémy: bagging, boosting, aplikácia na rozhodovacie stromy, náhodné lesy**

- *Bagging* (bootstrap aggregating) je technika, kde sa trénuje viacero modelov (napr. rozhodovacích stromov) na rôznych náhodných podmnožinách dát a ich predikcie sa priemerujú alebo hlasujú. Znižuje rozptyl modelu a zvyšuje stabilitu [12] [13] [14].
- *Boosting* trénuje modely sekvenčne, pričom každý ďalší model sa zameriava na príklady, ktoré predchádzajúce modely klasifikovali nesprávne. Výsledné predikcie sú váženým hlasovaním všetkých modelov, pričom väčšiu váhu majú presnejšie modely [12] [15].
- *Rozhodovacie stromy* sú často používané základné modely v baggingu aj boostingu, pretože sú náchylné na vysoký rozptyl, ktorý ensemble metódy efektívne znižujú [13] [14].
- *Náhodné lesy* (random forests) kombinujú bagging s náhodným výberom podmnožiny vstupných atribútov pre každý strom, čím sa zvyšuje diverzita stromov a zlepšuje presnosť a robustnosť výsledného modelu [16].

5. **Teória strojového učenia: matematický model, výchylka a rozptyl, holdout testing, PAC učenie, VC dimenzia**

- *Matematický model strojového učenia* pozostáva z výberu hypotéz (modelov), ktoré sa učia na trénovacích dátach, pričom cieľom je minimalizovať chybu na neznámych dátach [17] [5].
- *Výchylka* (bias) je systematická chyba modelu, ktorá vzniká pri príliš jednoduchých modeloch. *Rozptyl* (variance) je chyba spôsobená prílišnou citlivosťou na trénovacie dáta. Cieľom je nájsť rovnováhu medzi výchylkou a rozptylom (bias-variance trade-off) [5].
- *Holdout testing* je metóda hodnotenia modelu, kde sa časť dát odloží (hold-out set) a použije až na testovanie modelu po natrénovaní, aby sa získal neskreslený odhad jeho

výkonnosti^[18].

- *PAC učenie* (probably approximately correct) je teoretický rámec, ktorý formálne definuje, kedy je učenie úspešné: model má byť s vysokou pravdepodobnosťou približne správny, pričom sa analyzuje, koľko tréningových príkladov je potrebných na dosiahnutie požadovanej presnosti a spoľahlivosti^{[5] [19]}.
- *VC dimenzia* (Vapnik-Červonenkis) je mierou kapacity modelu – určuje, koľko rôznych rozdelení dát dokáže model reprezentovať. Vyššia VC dimenzia znamená väčšiu flexibilitu, ale aj vyššie riziko preučenia. VC dimenzia je kľúčová pri odhadoch vzorkovacej zložitosti v PAC učení^{[5] [19]}.



1. Regresia: Lineárna regresia, normálne rovnice, gradientová metóda, generalizovaná lineárna regresia, regularizácia

Lineárna regresia

Lineárna regresia je štatistická metóda na modelovanie vzťahu medzi jednou alebo viacerými nezávislými premennými (regresormi) a jednou závislou premennou. Model má tvar $y = a_0 + a_1x + \epsilon$, kde a_0 je priesečník, a_1 smernica a ϵ náhodná chyba^{[20] [21] [22] [23]}. Cieľom je nájsť takú regresnú priamku, ktorá najlepšie vystihuje závislosť medzi premennými, pričom sa minimalizuje súčet štvorcov rozdielov medzi skutočnými a predikovanými hodnotami (metóda najmenších štvorcov)^{[21] [23]}.

Normálne rovnice

Normálne rovnice poskytujú analytické riešenie lineárnej regresie. Pre jednoduchú regresiu je možné vypočítať koeficienty priamo zo sústavy rovníc odvodených z minimalizácie súčtu štvorcov chýb. Vo viacrozmernom prípade je riešenie $\theta = (X^T X)^{-1} X^T y$, kde X je matica vstupov a y vektor výstupov^{[21] [22]}.

Gradientová metóda

Pri veľkých datasetoch alebo ak je inverzia matice neefektívna, využíva sa gradientový zostup (gradient descent), ktorý iteratívne upravuje parametre modelu v smere najväčšieho znižovania chyby. Tento prístup je škálovateľný a vhodný aj pre online učenie^{[21] [22]}.

Generalizovaná lineárna regresia

Generalizovaná lineárna regresia rozširuje základný model o možnosť nelineárnych vzťahov medzi vstupmi a výstupom pomocou bázových funkcií alebo transformácií vstupov. Príkladom je polynomiálna regresia, kde sa modelujú vyššie mocniny vstupných premenných^[22].

Regularizácia

Regularizácia (napr. Ridge, Lasso) pridáva do chybovej funkcie penalizačný člen, ktorý obmedzuje veľkosť koeficientov. Pomáha predchádzať preučeniu (overfitting), najmä pri vysokom počte premenných alebo šume v dátach^[22].

2. Neurónové siete: Logistická regresia vs. perceptrón, skryté vrstvy, CNN, spätná propagácia

Logistická regresia vs. perceptrón

Logistická regresia je štatistický model na binárnu klasifikáciu, ktorý využíva sigmoidálnu aktivačnú funkciu a výstup interpretuje ako pravdepodobnosť. Je vhodná na úlohy, kde je výstup dichotomický (0/1) ^[24]. Perceptrón je základný model neurónovej siete, ktorý používa prahovú (step) funkciu a dokáže riešiť iba lineárne separovateľné úlohy. Hlavný rozdiel je v aktivačnej funkcii a interpretácii výstupu ^{[25] [24]}.

Skryté vrstvy v neurónových sieťach

Skryté vrstvy (hidden layers) sa nachádzajú medzi vstupnou a výstupnou vrstvou. Ich úlohou je transformovať a spracovať vstupné dáta do reprezentácie, ktorú môže výstupná vrstva efektívne vyhodnotiť. Viac skrytých vrstiev umožňuje modelovať komplexné nelineárne vzťahy a vedie k hlbokému učeniu (deep learning) ^{[26] [27]}.

Konvolučné neurónové siete (CNN)

CNN sú špeciálne architektúry určené na spracovanie dát s priestorovou štruktúrou (najmä obrázkov). Hlavné komponenty sú:

- vstupná vrstva (prijíma obrazové dáta),
 - konvolučné vrstvy (extrahujú lokálne príznaky cez filtre),
 - pooling vrstvy (zmenšujú rozmer dát a zvýrazňujú dominantné príznaky),
 - plne prepojené vrstvy (mapujú extrahované príznaky na výstupné triedy) ^[28].
- CNN efektívne zachytávajú priestorové závislosti a sú štandardom v počítačovom videní.

Spätná propagácia (backpropagation)

Spätná propagácia je algoritmus na učenie viacvrstvových neurónových sietí. Využíva gradientový zostup, pričom chyba z výstupnej vrstvy sa postupne šíri späť do všetkých vrstiev siete a podľa nej sa upravujú váhy. Tento proces umožňuje efektívne učiť aj hlboké siete ^[29].

3. Support Vector Machines: Základná a duálna formulácia, kernelová metóda

Základná formulácia

SVM hľadá optimálnu deliacu hyperplochu medzi dvoma triedami tak, aby bola maximalizovaná vzdialenosť (margin) medzi najbližšími bodmi oboch tried (podporné vektory) ^{[30] [31]}. Pri ideálne separovateľných dátach existuje nekonečne veľa hyperplôch, SVM vyberá tú s najväčším marginom.

Duálna formulácia

Optimalizačný problém SVM možno prepísať do duálnej formy, kde sa riešenie vyjadruje ako lineárna kombinácia trénovacích príkladov (Lagrangeove multiplikátory). Táto formulácia umožňuje efektívne využitie kernelových funkcií a riešenie nelineárnych úloh ^{[32] [31]}.

Kernelová metóda

Kernelová metóda (kernel trick) umožňuje implicitne transformovať vstupné dáta do vyššej dimenzie, kde sú lineárne separovateľné, bez explicitného výpočtu transformácie. Najčastejšie

používané kernely sú RBF (gaussovský), polynomiálny a sigmoidálny. Kernelová funkcia musí spĺňať Mercerove podmienky^{[30] [32] [33] [31]}.

4. Hlasovacie schémy: Bagging, boosting, rozhodovacie stromy, náhodné lesy

Bagging (Bootstrap Aggregating)

Bagging trénuje viacero modelov (napr. rozhodovacích stromov) na rôznych náhodných podmnožinách dát. Ich predikcie sa kombinujú (hlasovaním alebo spriemerovaním), čím sa znižuje rozptyl modelu a zvyšuje stabilita. Modely sú trénované paralelne a nezávisle^{[34] [35]}.

Boosting

Boosting trénuje modely sekvenčne – každý ďalší model sa zameriava na príklady, ktoré predchádzajúce modely klasifikovali nesprávne (zvyšuje im váhu). Výsledné predikcie sú váženým hlasovaním všetkých modelov. Boosting znižuje výchylku a zvyšuje presnosť, ale je citlivý na šum^[34].

Rozhodovacie stromy

Rozhodovacie stromy sú hierarchické modely, ktoré rozdeľujú dáta na základe podmienok (if/else) podľa dôležitosti príznakov. Stromy sú ľahko interpretovateľné, ale náchylné na preučenie. Preto sa často používajú ako základné modely v baggingu a boostingu^[36].

Náhodné lesy (Random Forests)

Náhodný les je ansámblová metóda, ktorá kombinuje bagging s náhodným výberom podmnožiny vstupných atribútov pre každý strom. To zvyšuje diverzitu stromov a zlepšuje presnosť a robustnosť modelu. Výsledkom je spriemerovaná alebo hlasovaná predikcia viacerých stromov^[35].

5. Teória strojového učenia: Matematický model, výchylka a rozptyl, holdout testing, PAC učenie, VC dimenzia

Matematický model strojového učenia

Model strojového učenia pozostáva z množiny hypotéz, ktoré sa učia na tréningových dátach, pričom cieľom je minimalizovať chybu na neznámych dátach. Formálne: X – množina tréningových príkladov, C – cieľová funkcia, H – množina hypotéz, L – algoritmus učenia^[37].

Výchylka (bias) a rozptyl (variance)

Výchylka je systematická chyba modelu spôsobená jeho jednoduchosťou, rozptyl je chyba spôsobená prílišnou citlivosťou na tréningové dáta. Kľúčom je nájsť rovnováhu medzi výchylkou a rozptylom (bias-variance trade-off), aby model dobre generalizoval^[38].

Holdout testing

Holdout testing je spôsob hodnotenia modelu, kde časť dát (holdout set) sa odloží a použije až na testovanie modelu po natrénovaní. Umožňuje získať neskreslený odhad výkonnosti modelu na neznámych dátach^[39].

PAC učenie (Probably Approximately Correct)

PAC učenie je teoretický rámec, ktorý skúma, koľko tréningových príkladov je potrebných na to,

aby model s vysokou pravdepodobnosťou dosiahol približne správnu hypotézu. Analyzuje tiež výpočtovú zložitosť učenia a prípustný počet chybné klasifikovaných príkladov počas učenia^[37].

VC dimenzia (Vapnik-Červonenkis)

VC dimenzia je mierou kapacity modelu – udáva, koľko rôznych rozdelení dát dokáže model reprezentovať. Vyššia VC dimenzia znamená väčšiu flexibilitu, ale aj vyššie riziko preučenia. Je kľúčová pri odhadoch vzorkovacej zložitosti v PAC učení^[37].

✱✱

Hlbšia analýza vybraných metód strojového učenia

1. Regresné metódy a ich matematický základ

1.1 Lineárna regresia a normálne rovnice

Lineárna regresia modeluje vzťah medzi nezávislými premennými \mathbf{X} a závislou premennou \mathbf{Y} pomocou lineárnej kombinácie parametrov:

$$y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \epsilon_i$$

Normálne rovnice poskytujú analytické riešenie minimalizácie súčtu štvorcov chýb (RSS):

$$\beta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Tento prístup vyžaduje invertibilitu matice $\mathbf{X}^T \mathbf{X}$, čo obmedzuje jeho použiteľnosť pri vysokorozmerných dátach^{[40] [41]}. Pre matice s hodnotou menšou ako počet premenných sa používa pseudoinverzia.

1.2 Gradientové metódy

Iteratívne algoritmy ako **stochastický gradientový zostup** (SGD) riešia problémy s veľkými dátami:

$$\beta^{(k+1)} = \beta^{(k)} - \eta \nabla_{\beta} L(\beta^{(k)})$$

kde η je rýchlosť učenia. SGD používa mini-dávky (typicky 32-512 príkladov) pre stabilnejšie konvergencie^{[42] [43]}. Adaptívne metódy ako Adam kombinujú moment a RMSProp pre lepšiu konvergenciu.

1.3 Generalizovaná lineárna regresia

Rozširuje klasický model o:

1. Linkovú funkciu $g(\mu_i) = \mathbf{x}_i^T \beta$
2. Exponenciálnu rodinu rozdelení:

$$f(y_i|\theta_i) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right)$$

Pre logistickú regresiu platí logit link $g(\mu) = \ln(\mu/(1 - \mu))$ s binomickým rozdelením [44] [45]. Poissonova regresia používa log-link pre početné údaje.

1.4 Regularizačné techniky

Ridge regresia (L2) pridáva penalizáciu:

$$\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_2^2$$

LASSO (L1) vykonáva výber premenných:

$$\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_1$$

Elastic Net kombinuje obe penalizácie s parametrom $\alpha \in [0, 1]$ [46] [47]. Pre $\lambda > 0$ regularizácia znižuje podmienené číslo matice $\mathbf{X}^T \mathbf{X}$, čím zlepšuje numerickú stabilitu.

2. Architektúry neurónových sietí

2.1 Perceptrón vs. logistická regresia

Perceptrón používa prahovú funkciu:

$$f(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Logistická regresia aplikuje sigmoidu:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Kľúčový rozdiel je v diferencovateľnosti - sigmoida umožňuje gradientové metódy, čo je kľúčové pre učenie viacvrstvových sietí [48].

2.2 Skryté vrstvy a univerzálna aproximácia

Podľa Cybenkovej vety dokáže sieť s jednou skrytou vrstvou a dostatočným počtom neurónov aproximovať ľubovoľnú spojitú funkciu na kompaktnom podpriestore \mathbb{R}^n . Pre funkciu $f: \mathbb{R}^d \rightarrow \mathbb{R}$ existujú váhy w_i , b_i a v_i také, že:

$$f(\mathbf{x}) \approx \sum_{i=1}^N v_i \sigma(w_i^T \mathbf{x} + b_i)$$

Hĺbka siete zvyšuje kapacitu modelu exponenciálne v porovnaní so šírkou [49].

2.3 Konvolučné neurónové siete

CNN využívajú:

- **Konvolučné filtre** s váhovým zdieľaním
- **Pooling vrstvy** (max/average) pre invariantnosť k posunu
- **Batch normalizáciu** pre stabilizáciu distribúcie aktivácií

Matematicky, konvolúcia pre 2D vstup:

$$(F * K)(i, j) = \sum_m \sum_n F(i - m, j - n) K(m, n)$$

Moderné architektúry ako ResNet využívajú skip connections:

$$\mathcal{H}(\mathbf{x}) = \mathcal{F}(\mathbf{x}) + \mathbf{x}$$

2.4 Spätná propagácia

Algoritmus pozostáva z:

1. **Forward pass:** Výpočet stratovej funkcie
2. **Backward pass:** Reťazové pravidlo pre gradienty

Pre stratovú funkciu L a váhy $W^{(l)}$ vrstvy l :

$$\frac{\partial L}{\partial W^{(l)}} = \frac{\partial L}{\partial a^{(l)}} \cdot \frac{\partial a^{(l)}}{\partial z^{(l)}} \cdot \frac{\partial z^{(l)}}{\partial W^{(l)}}$$

kde $z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$ a $a^{(l)} = \sigma(z^{(l)})$. Pre ReLU aktiváciu:

$$\frac{\partial a^{(l)}}{\partial z^{(l)}} = \begin{cases} 1 & z^{(l)} > 0 \\ 0 & \text{inak} \end{cases}$$

3. Metódy podporných vektorov

3.1 Primal problém

Optimalizačný problém pre lineárne separovateľné dáta:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad \forall i \end{aligned}$$

Margin je daný $2/\|\mathbf{w}\|$. Pre nepresne separovateľné dáta sa zavádza slack premenná ξ_i s penalizačným parametrom C [50].

3.2 Duálna formulácia

Lagrangeove multiplikátory α_i vedú k duálnej forme:

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

s podmienkami $0 \leq \alpha_i \leq C$ a $\sum_i \alpha_i y_i = 0$. Podporné vektory majú $\alpha_i > 0$.

3.3 Kernelový trik

Nelineárne zobrazenie $\phi : \mathcal{X} \rightarrow \mathcal{H}$ do RKHS priestoru:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$$

Bežné kernele:

- **RBF:** $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$
- **Polynomiálny:** $K(\mathbf{x}, \mathbf{y}) = (\gamma \mathbf{x}^T \mathbf{y} + r)^d$
- **Sigmoid:** $K(\mathbf{x}, \mathbf{y}) = \tanh(\gamma \mathbf{x}^T \mathbf{y} + r)$

4. Ansámblové metódy

4.1 Bagging (Bootstrap Aggregating)

- Trénuje B modelov na bootstrapových vzorkách
- Redukuje rozptyl: $\text{Var}(\bar{f}) = \frac{1}{B} \text{Var}(f)$
- Efektívne pre nízko vychýlené modely s vysokým rozptylom

4.2 Boosting

AdaBoost algoritmus:

1. Inicializácia váh $w_i^{(1)} = 1/n$
2. Pre $t = 1$ až T :
 - Trénovanie klasifikátora h_t s váhami $w^{(t)}$
 - Výpočet chyby $\epsilon_t = \sum_{i=1}^n w_i^{(t)} \mathbf{1}(h_t(\mathbf{x}_i) \neq y_i)$
 - Aktualizácia váh $w_i^{(t+1)} = w_i^{(t)} \exp(-\alpha_t y_i h_t(\mathbf{x}_i))$

kde $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$

4.3 Náhodné lesy

Kombinuje:

- **Feature bagging:** Náhodný výber $m \approx \sqrt{p}$ premenných
- **Stromová diverzita:** Maximálna hĺbka, min. vzorky na rozdelenie

Out-of-bag chyba poskytuje interný odhad generalizácie:

$$\text{OOB-error} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(y_i \neq \hat{y}_i^{\text{OOB}})$$

5. Teoretické základy učenia

5.1 Bias-Variance Tradeoff

Rozklad strednej štvorcovej chyby:

$$\mathbb{E}[(y - \hat{f})^2] = \text{Bias}(\hat{f})^2 + \text{Var}(\hat{f}) + \sigma^2$$

kde:

- **Bias** = $\mathbb{E}[\hat{f}] - f$
- **Var** = $\mathbb{E}[(\hat{f} - \mathbb{E}[\hat{f}])^2]$
- σ^2 = neznížiteľný šum

5.2 VC dimenzia

Pre triedu funkcií \mathcal{H} :

$$\text{VC-dim}(\mathcal{H}) = \max\{m : \Pi_{\mathcal{H}}(m) = 2^m\}$$

kde $\Pi_{\mathcal{H}}(m)$ je rastová funkcia. Pre lineárne klasifikátory v \mathbb{R}^d : $\text{VC-dim} = d + 1$

5.3 PAC učenie

Triada $(\mathcal{X}, \mathcal{Y}, \mathcal{H}, \mathcal{D}, \mathcal{L})$ je PAC učiteľná, ak existuje algoritmus A a funkcia $m : (0, 1)^2 \rightarrow \mathbb{N}$ tak, že pre každé $\epsilon, \delta > 0$ a distribúciu \mathcal{D} :

$$\mathbb{P} \left[L(h_A) \leq \min_{h \in \mathcal{H}} L(h) + \epsilon \right] \geq 1 - \delta$$

kde $m = \Omega \left(\frac{\text{VC-dim}(\mathcal{H}) + \ln(1/\delta)}{\epsilon^2} \right)$

5.4 Validácia modelov

k-fold cross-validácia:

1. Rozdelenie dát na k častí
2. Postupné trénovanie na $k - 1$ foldoch
3. Validácia na zvyšnom folde
4. Spriemerovanie výsledkov

Holdout je špeciálny prípad s $k = 2$. Pre malé datasety sa odporúča $k = 5$ alebo 10 ^[51].

**

1. <http://compbio.fmph.uniba.sk/vyuka/ml/handouts/s2.pdf>
2. <https://dspace.cuni.cz/bitstream/handle/20.500.11956/182163/120445983.pdf?sequence=1&isAllowed=y>
3. <http://compbio.fmph.uniba.sk/vyuka/ml/old/2008/handouts/linreg.pdf>
4. <http://www2.fiit.stuba.sk/~kvasnicka/MathematicsPhD/LECTURE3.pdf>
5. <http://compbio.fmph.uniba.sk/vyuka/ml/handouts/zhrnutie2014.pdf>
6. https://is.muni.cz/th/wdnpv/R_Oprendek_Diplomova_Prace.pdf
7. <https://umelainteligencia.sk/uvod-do-neuronovych-sieti/>
8. https://www.karlin.mff.cuni.cz/~tuma/Aplikace16/Prace/Neuronove_siete_a_AlphaGo_Dominika_Kubanova.pdf
9. <https://umelainteligencia.sk/zaklady-support-vector-machines/>
10. http://www2.fiit.stuba.sk/~cernans/nn/nn_texts/neuronove_siete_priesvitky_svm.pdf
11. <https://ei.fhi.sk/index.php/EAI/article/view/306/303>
12. <https://www.nb-data.com/p/comparing-model-ensembling-bagging>
13. https://cw.fel.cvut.cz/old/_media/courses/y336vd/prednasky/p11-ensemble10.pdf
14. <http://smnd.sk/mcibula/alg/BA.html>
15. <https://www.aiacceleratorinstitute.com/boosting-and-bagging-powerful-ensemble-methods-in-machine-learning/>
16. https://cs.wikipedia.org/wiki/Náhodný_les
17. <https://umelainteligencia.sk/algoritmy-strojoveho-ucenia/>
18. <https://www.statsig.com/glossary/hold-out>
19. <https://www.cs.ox.ac.uk/people/varun.kanade/teaching/CLT-MT2018/lectures/lecture01.pdf>
20. <https://kurzy.kpi.fei.tuke.sk/nm/student/13.html>
21. <https://smnd.sk/mcibula/alg/linreg.html>
22. <https://www.unite.ai/sk/čo-je-lineárna-regresia/>
23. <https://www.sjf.stuba.sk/buxus/docs/katedry/Fyzika/Skripta/skripta11.pdf>
24. <https://dspace.vut.cz/bitstreams/3e9f0cf1-0551-4b03-8466-2cd240c1ec74/download>
25. https://is.muni.cz/th/wdnpv/R_Oprendek_Diplomova_Prace.pdf
26. <http://mathandit.blogspot.com/2016/03/neuronove-siete-cast-1.html>

27. <https://umelainteligencia.sk/uvod-do-neuronovych-sieti/>
28. <https://sk.eitca.org/umela-inteligencia/eitc-ai-dlhf-hlboké-učenie-s-tensorflow/konvolučné-neurónové-siete-v-tensorflow/základy-konvolučných-neurónových-sieti/skúška-prehľad-konvolučných-neurónových-sieti-základy/aké-sú-hlavné-komponenty-konvolučnej-neurónovej-siete-cnn-a-ako-prispievajú-k-rozpoznávaniu-obrazu/>
29. [https://sk.wikipedia.org/wiki/Spätná_väzba_\(všeobecne\)](https://sk.wikipedia.org/wiki/Spätná_väzba_(všeobecne))
30. <https://kristina.machova.website.tuke.sk/prezentacieSU/SVM.pptx>
31. <https://umelainteligencia.sk/zaklady-support-vector-machines/>
32. <https://ei.fhi.sk/index.php/EAI/article/view/306/303>
33. <https://www.zzv.h.sk/data/files/84.pdf>
34. <https://www.youtube.com/watch?v=0SgrtqdmV4M>
35. https://cs.wikipedia.org/wiki/Náhodný_les
36. <https://umelainteligencia.sk/nahlad-do-algoritmu-rozhodovaci-strom/>
37. <https://kristina.machova.website.tuke.sk/pdf/monoSUvSSL.pdf>
38. <http://compbio.fmph.uniba.sk/vyuka/ml/handouts/learning-theory-v0.pdf>
39. <https://posthog.com/tutorials/holdout-testing>
40. <https://kurzy.kpi.fei.tuke.sk/nm/student/13.html>
41. <http://compbio.fmph.uniba.sk/vyuka/ml/handouts/s2.pdf>
42. https://en.wikipedia.org/wiki/Stochastic_gradient_descent
43. <https://stanford.edu/~rezab/classes/cme323/S15/notes/lec11.pdf>
44. https://is.muni.cz/el/1431/jaro2008/Bi7490/um/02_regrese_glm.pdf
45. https://tomas-bacigal.quarto.pub/pokrocile-metody-statistickeho-modelovania/07_generalized_lm.html
46. <https://thesis.science.upjs.sk/~pchomic/clanok.pdf>
47. <https://www.33rdsquare.com/lasso-and-ridge-regularization-a-rescuer-from-overfitting/>
48. <https://dspace.vut.cz/items/56638aac-ed7b-450e-8786-4b28f07b0c81>
49. <https://www.cs.ubc.ca/~schmidtm/Courses/5XX-S20/S2.pdf>
50. <https://dspace.cuni.cz/bitstream/handle/20.500.11956/182163/120445983.pdf?sequence=1&isAllowed=y>
51. <https://ei.fhi.sk/index.php/EAI/article/view/18/33>