TSVD 10

Technológie spracovania veľkých dát

Peter Bednár, Martin Sarnovský

Obsah

- Dátové prúdy
- Konceptový drift
- Adaptácia na drift
- Vyhodnocovanie adaptívnych modelov

Dátové prúdy

- **Dátové prúdy** (*data streams*) - dáta nie sú spracovávané ako statický dataset, ale ako kontinuálne prichádzajúci tok, ktorý musí byť analyzovaný priebežne bez možnosti jeho kompletného uloženia do pamäte
- Dôsledok rastu objemu dát generovaných modernými systémami (IoT zariadenia, webové služby, finančné transakcie)
  - tradičné dávkové spracovanie nedokáže zabezpečiť dostatočne nízku latenciu
- Cieľ - schopnosť robiť rozhodnutia v reálnom čase alebo tzv. "*near real-time*", čo znamená, že model musí produkovať predikcie okamžite po príchode dát
- V mnohých aplikáciách má informácia krátku životnosť (napr. detekcia podvodov), preto oneskorené spracovanie vedie k strate hodnoty dát
- Tento posun mení základný prístup k strojovému učeniu – od offline učenia k online a adaptívnym metódam

Dátové prúdy - definícia

- Dátový prúd je definovaný ako **usporiadaná sekvencia** prvkov $S = \{x_1, x_2, ..., x_t\}$, kde každý prvok prichádza v čase a reprezentuje pozorovanie z generujúceho procesu
- Každý prvok môže obsahovať viacero atribútov a prípadne cieľovú premennú
  - Distribúcia generovania dát je často *neznáma* a môže sa *meniť v čase*
- Pri riešení prediktívnych úloh na streamoch pracujeme s dvojicami (x_t, y_t), pričom cieľom je aproximovať funkciu $f(x)$, ktorá mapuje vstupy na výstupy
- Rozdiel oproti "klasickému" učeniu - dáta **nie sú** dostupné naraz
  - model musí byť schopný pracovať s čiastočnou informáciou a postupne sa aktualizovať
- Takýto model reflektuje reálne prostredie, kde dáta vznikajú dynamicky a nie sú k dispozícii v úplnej forme

Vlastnosti streamov

- Streamy sú potenciálne **nekonečné**, čo znamená, že nie je možné uložiť celý dataset, a preto algoritmy musia pracovať s obmedzenou pamäťou a často využívať sumarizačné techniky
- Dáta prichádzajú **sekvenčne** a systém nemá kontrolu nad ich poradím ani nad tým, kedy presne budú dostupné, čo komplikuje ich spracovanie
- Spracovanie musí byť realizované jedným prechodom (**one-pass**), pretože **opakované** čítanie dát **nie je možné** alebo je príliš nákladné
- Rýchlosť príchodu dát môže byť **veľmi vysoká**, čo kladie nároky na výpočtovú efektivitu a latenciu algoritmov
- **Distribúcia dát** sa môže **meniť v čase**, čo znamená, že model musí byť schopný adaptácie na nové podmienky.

Typy streamov

- **Stacionárne streamy** - predpokladajú, že distribúcia dát *P(x,y)* je konštantná
  - v praxi zriedkavé a väčšina reálnych aplikácií tento predpoklad porušuje
- **Nestacionárne streamy** - charakteristické tým, že distribúcia dát *P(x,y)* sa mení v čase, čo vedie k problémom ako konceptový drift a k zhoršovaniu výkonnosti modelov
- Úlohy na streamoch môžu byť supervised, semi-supervised alebo unsupervised v závislosti od dostupnosti cieľovej premennej
  - Špecifikum týchto úloh - často sú labely **oneskorené** alebo **chýbajú**
- Dáta môžu byť jednoduché (vektorové) alebo komplexné (grafy, texty)
- V praxi sa často stretávame s kombináciou viacerých typov streamov a rôznymi úrovňami komplexity

Výzvy spracovania streamov

- Obmedzené výpočtové zdroje (pamäť, čas) - algoritmy musia byť navrhnuté tak, aby boli efektívne a škálovateľné
- Nemožnosť uložiť všetky dáta - potreba aproximácie a sumarizácie informácií
- Oneskorené alebo chýbajúce labely - komplikujú učenie a evaluáciu modelov
- Zmena distribúcie dát - spôsobuje degradáciu modelu a vyžaduje adaptívne prístupy
- Potreba kontinuálneho učenia znamená, že model musí byť aktualizovaný bez úplného pretrénovania

Konceptový drift

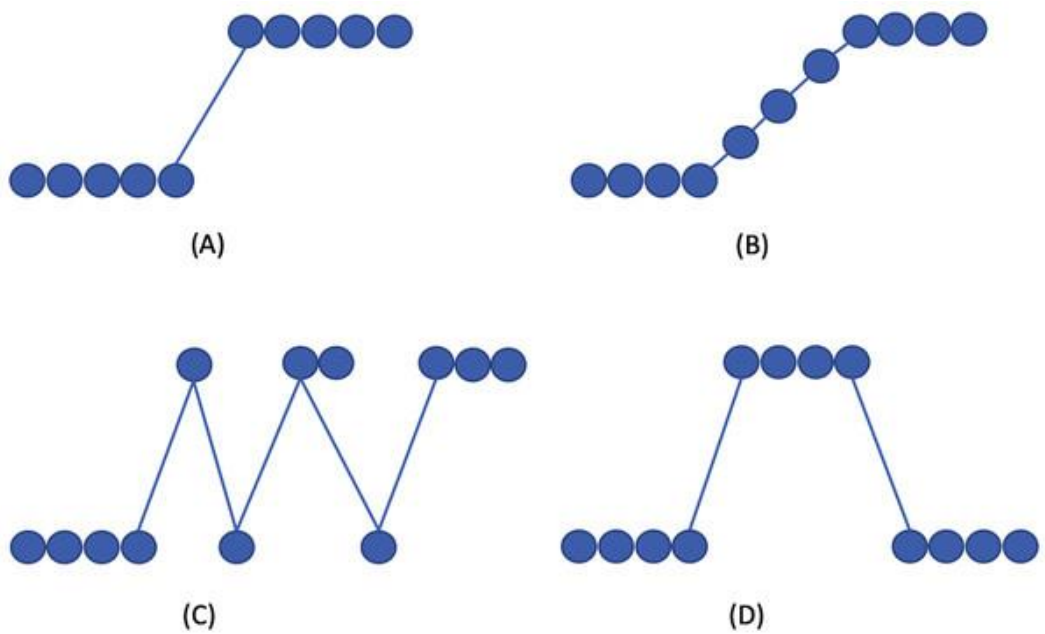
- Označuje zmenu v distribúcii dát v čase, ktorá spôsobuje, že model naučený na historických dátach prestáva byť aktuálny
- Formálne: $P_t(x,y) \neq P_{t+\Delta}(x,y)$
  - čo znamená, že vzťah medzi predikujúcimi a predikovanou premennou sa mení a naučené pravidlá prestávajú platiť
- Drift je typický pre dynamické prostredia, ako sú finančné trhy, používateľské správanie alebo senzory
- Bez adaptácie modelu vedie drift k zhoršovaniu presnosti predikcií
- Schopnosť detekovať a reagovať na drift je kľúčová vlastnosť modelov pre streamy

Zdroje driftu

- Zmeny správania používateľov (napr. nákupné preferencie) vedú k zmene distribúcie dát
- Sezónne efekty spôsobujú opakované zmeny v dátach (napr. sviatky, počasie)
- Technologické zmeny (napr. nový systém) môžu zmeniť charakter generovaných dát
- Externé faktory ako ekonomické alebo sociálne udalosti ovplyvňujú distribúciu dát
- Drift môže byť spôsobený aj degradáciou senzorov alebo zmenou meracích podmienok

Typy driftu

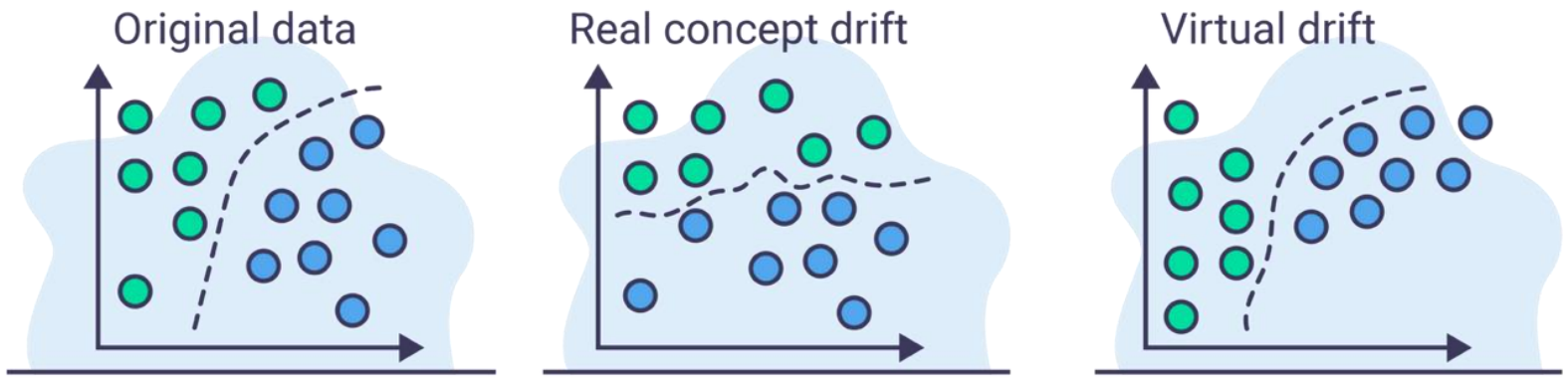
- *Náhly drift (sudden drift)* - náhla zmena distribúcie, kde starý koncept je okamžite nahradený novým
- *Inkrementálny drift (incremental drift)* - plynulá zmena distribúcie bez jasného bodu zlomu
- *Postupný drift (gradual drift)* - postupná zmena, kde starý a nový koncept koexistujú počas určitého obdobia
- *Opakujúci sa drift (re-occuring drift)* - staré koncepty opätovne objavujú v čase (napr. sezónnosť)
- Každý typ driftu vyžaduje odlišnú stratégiu detekcie a adaptácie



Concept drift types according to Gama et al. (2014).
(A) Sudden/Abrupt, (B) incremental, (C) gradual, (D) re-occuring.

Real vs. Virtual drift

- **Skutočný drift** - zmena distribúcie dát *P(x,y)*, ktorá vedie k zmene rozhodovacej funkcie *f(x)* a priamo ovplyvňuje predikcie modelu
  - má väčší dopad na výkon modelu, pretože mení samotný vzťah medzi vstupmi a výstupmi
- **Virtuálny drift** - predstavuje zmenu distribúcie vstupov *P(x)*, ktorá ale nemení rozhodovaciu funkciu
  - môže spôsobiť degradáciu modelu nepriamo, napr. zmenou hustoty dát v priestore
- Rozlíšenie týchto typov je dôležité pre návrh adaptačných mechanizmov



Detekcia driftu

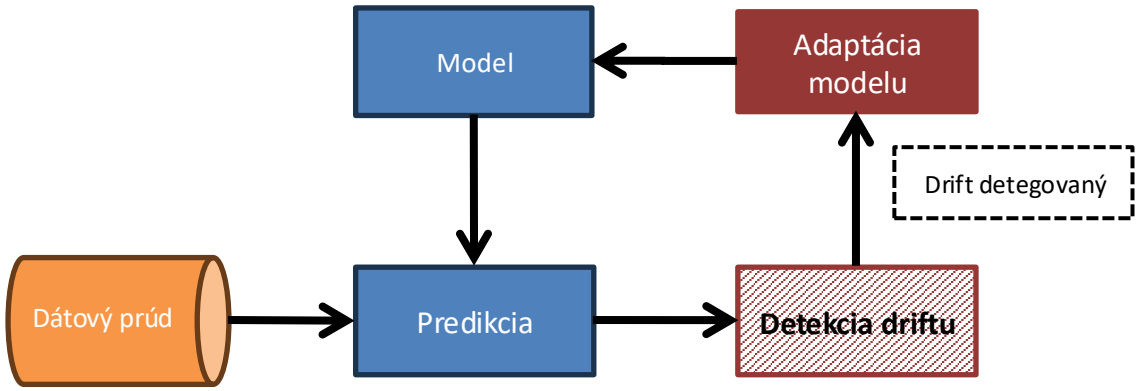
- Detekčné metódy sú založené na sledovaní výkonosti modelu (napr. error rate), alebo vlastností dát
- Metódy ako *DDM (Drift Detection Method)* sledujú zmeny v chybovosti modelu a signalizujú drift pri jej prudkom náraste
- *ADWIN (Adaptive Windowing)* využíva adaptívne okná a porovnáva distribúcie v rôznych časových úsekoch
- Detekcia driftu - kompromis medzi rýchlosťou detekcie a počtom falošných poplachov
- Nesprávna detekcia môže viesť k zbytočným aktualizáciám modelu alebo naopak k ignorovaniu zmien

Adaptácia na drift

- Výber prístupu závisí od charakteru dát a požiadaviek na presnosť a výpočtový výkon
- *Pasívne prístupy* - priebežne aktualizujú model bez explicitnej detekcie driftu, čím zabezpečujú kontinuálnu adaptáciu
- *Aktívne prístupy* - využívajú detekciu driftu a aktualizujú model len v prípade detegovanej zmeny
- *Hybridné prístupy* - kombinujú oba spôsoby s cieľom využiť ich výhody.

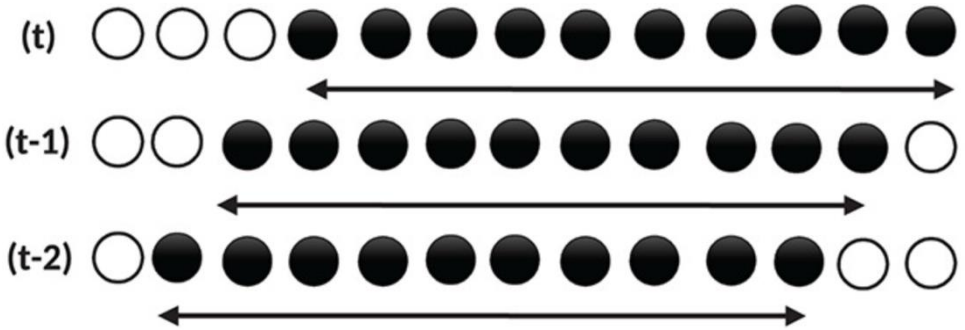
Explicitná detekcia pretrénovanie

- Adaptácia je riadená detekciou driftu pomocou štatistických testov alebo monitorovania chybovosti modelu
- Po detekcii driftu:
  - model sa úplne retrénuje
  - alebo sa aspoň čiastočne aktualizuje
- Typické metódy detekcie:
  - DDM (Drift Detection Method)
  - EDDM (Early Drift Detection Method)
  - ADWIN (Adaptive Windowing)
- Výhody - presná reakcia na zmeny
- Nevýhody - riziko falošných alarmov a oneskorenia detekcie
- Tento prístup je vhodný najmä pre náhly drift



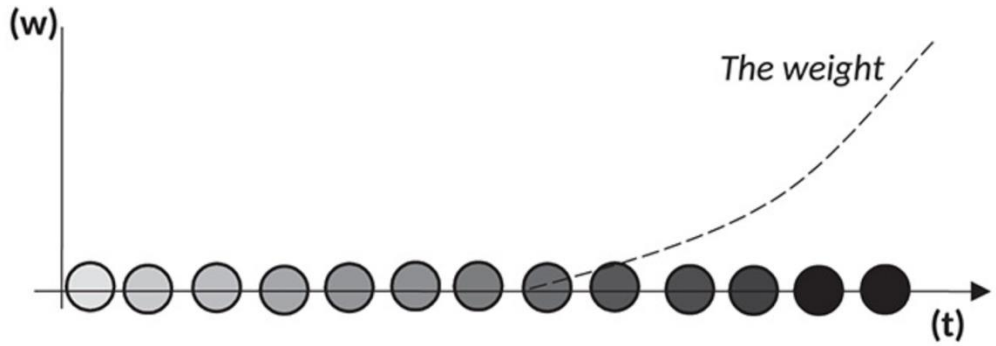
Sliding window adaptácia

- Uchovávajú len najnovšie dáta v definovanom časovom okne, čím implicitne zabúda staré koncepty
- Veľkosť okna - kritický parameter
  - malé okno = rýchla adaptácia na náhle zmeny, ale môže viesť k nestabilite modelu
  - veľké okno = robustnejšie odhady, ale pomalšia reakcia na drift
- Používajú sa rôzne varianty:
  - pevné okno
  - adaptívne okno (automaticky mení veľkosť podľa stability dát)
- Tieto prístupy sú jednoduché, ale efektívne najmä pri **lokálnych a krátkodobých zmenách distribúcie**



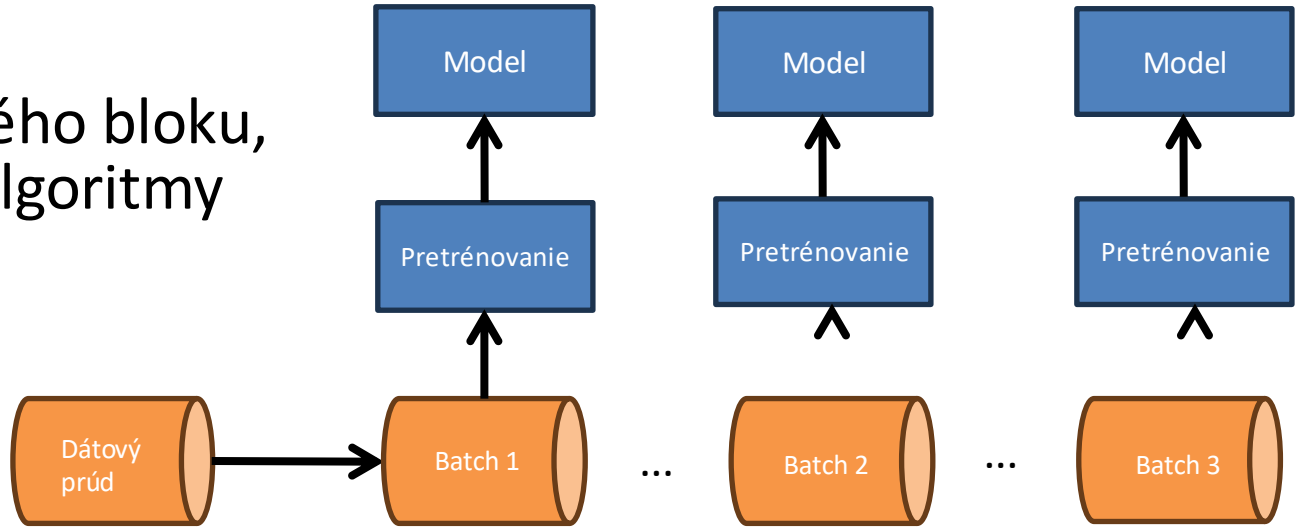
Fading factors a time decay

- Namiesto explicitného zahadzovania „starých“ dát sa pozorovaniam priraďujú váhy, ktoré klesajú s časom
  - staršie dáta sa stávajú menej významnými pre učenie
- Typicky sa používa exponenciálny pokles váhy: $w_t = \lambda^{(T-1)}$, kde $\lambda \in (0, 1)$ určuje rýchlosť zabúdania
- Tento prístup umožňuje hladkú adaptáciu bez náhlych zmien v trénovacej množine
- Je vhodný najmä pre **postupný** alebo **inkrementálny** drift, kde sa distribúcia mení postupne



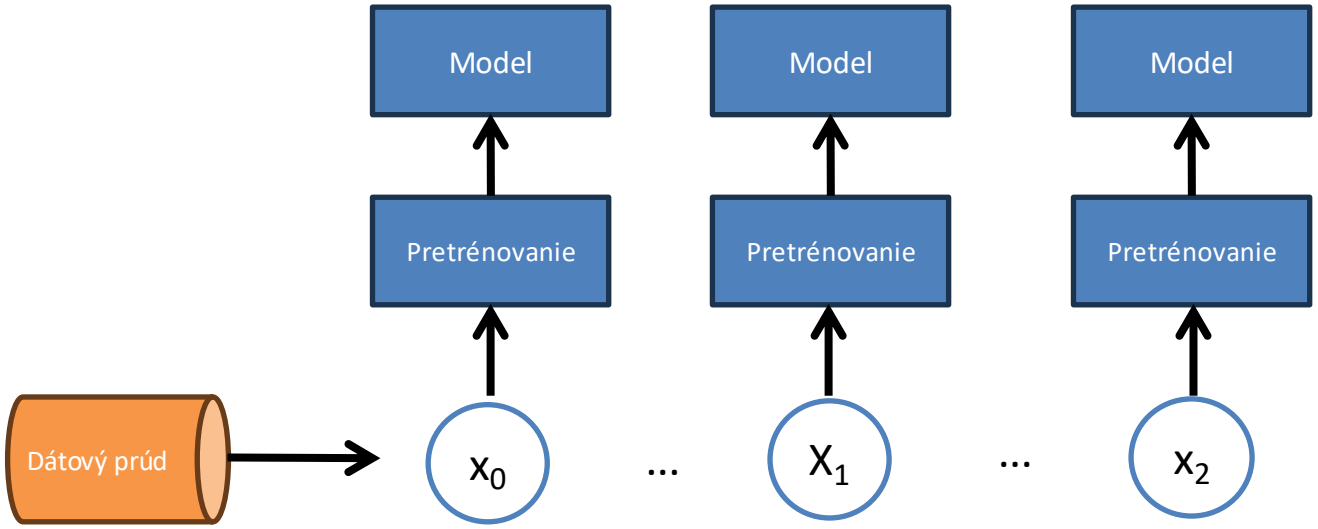
Učenie po krátkych dávkach

- Tzv. **chunk-based learning**, dáta zo streamu sú rozdelené na bloky (*chunks*), ktoré sú spracovávané ako micro-batch dáta
- Model je aktualizovaný po prijatí každého bloku, čo umožňuje použiť aj klasické batch algoritmy
- Výhody:
  - stabilnejšie učenie (viac dát naraz)
  - možnosť použiť komplexnejšie modely
- Nevýhody:
  - oneskorená reakcia na drift (závisí od veľkosti chunku)
  - predstavuje kompromis medzi batch a online učením



Online učenie

- Model je aktualizovaný po každom novom príklade alebo malej dávke dát bez potreby pretrénovania na celom datasete
- Typické algoritmy:
  - Stochastic Gradient Descent
  - Online perceptron
  - Naive Bayes (prirodzene inkrementálny)
- Výhodoa - nízka pamäťová náročnosť a schopnosť spracovávať dáta v reálnom čase
- Nevýhoda - citlivosť na šum a potreba dobre nastavených parametrov učenia
  - Tento prístup je základom väčšiny moderných systémov pre učenie na streamoch



Ostatné metódy

- **Instance selection prístupy**
  - miesto použitia všetkých dát sa selektujú len relevantné inštancie pre aktuálny koncept (napr. reservoir sampling)
  - Cieľ - redukovať pamäťové nároky, odstrániť zastarané alebo irelevantné dáta
- **Adaptácia modelu**
  - priamo upravujeme parametre modelu tak, aby reflektovali nové podmienky (napr. adaptívne learning rate, dynamická regularizácia adaptácia štruktúry modelu (napr. pruning stromov))
  - Efektívne pri modeloch, ktoré majú explicitnú parametrickú reprezentáciu
  - Používa sa najmä v kombinácii s online learningom.
- **Hybridné prístupy**
  - často sa kombinujú viaceré mechanizmy adaptácie, napr.:
    - drift detection + chunk-based learning
    - fading factors + instance selection
  - Cieľom je využiť výhody jednotlivých prístupov a minimalizovať ich nevýhody

Evaluácia streaming modelov

- V streamoch nie je možné použiť klasické rozdelenie na train/test, pretože dáta prichádzajú postupne
- Model sa neustále mení, čo znamená, že jeho výkon je potrebné sledovať **v čase**
- Evaluácia musí byť realizovaná online, bez potreby uchovávania veľkého množstva dát
- Výsledky musia reflektovať aktuálny stav modelu, nie historický priemer
- To vedie k potrebe nových evaluačných metodík

Prequential evaluation

- *Interleaved test-then-train* - každý prichádzajúci príklad zo streamu je najprv použitý na otestovanie aktuálneho modelu, a až následne je využitý na jeho aktualizáciu
- Simuluje reálne nasadenie - model vždy pracuje s historickými dátami a musí predikovať nové vzorky
- Formálne: pre každý časový krok *t* model vypočíta predikciu $\hat{y}_t = f_{t-1}(x_t)$, následne sa porovná s reálnou hodnotou y_t, vypočíta sa chyba $\widehat{L(y_t, y_t)}$, model sa aktualizuje na f_t pomocou (x_t, y_t)
- Výhody:
  - nevyžaduje explicitné rozdelenie dát na tréningovú a testovaciu množinu
  - poskytuje realistický odhad výkonu v dynamickom prostredí
  - umožňuje sledovať vývoj výkonu modelu v čase
- Nevýhody:
  - jednoduché kumulatívne metriky môžu byť skreslené staršími dátami
  - model môže byť penalizovaný za staré chyby aj keď sa už adaptoval (riešiteľné dodatočnými mechanizmov (napr. sliding window alebo fading factors))

Evaluácia pri oneskorených labeloch

- V mnohých aplikáciách nie je cieľová hodnota y_t dostupná okamžite po príchode vstupu x_t, ale až po určitom časovom oneskorení (napr. fraud detection – potvrdenie podvodu môže prísť až po dňoch, medicína– výsledok liečby je známy až po dlhšom čase, financie – úspešnosť rozhodnutia sa prejaví až v budúcnosti)
- Dôsledky pre učenie a evaluáciu:
  - model nemôže byť okamžite aktualizovaný, pretože nepozná správnu odpoveď
  - evaluácia modelu je oneskorená a nemusí reflektovať aktuálny stav modelu
- Dochádza k rozdeleniu procesu vyhodnotenia na *fázu predikcie (bez labelu)* a fázu *oneskorenej aktualizácie*
- Dôsledky:
  - oneskorená spätná väzba = pomalšia adaptácia modelu na koncept drift
  - model môže dlhšie pracovať s neaktuálnymi znalosťami, čo vedie k dočasnému zhoršeniu výkonu
- Evaluácia okrem metrík musí zohľadňovať časové oneskorenie medzi predikciou a dostupnosťou ground truth
- Pri porovnávaní - nielen presnosť, ale aj schopnosť adaptácie pri oneskorených labeloch
- **Delayed prequential evaluation** - model vykoná predikciu v t, ale chyba sa vyhodnotí až v $t + \Delta$, keď je label dostupný (metriky počítané s oneskorením)
- **Bufferovanie dát** - príklady sa ukladajú do bufferu a model sa aktualizuje až po získaní labelov (náročné na pamäť)
- **Semi-supervised učenie** - model sa učí aj z nelabelovaných dát a labely sú využité len keď sú dostupné
- **Aktívne učenie** - systém si selektívne vyberá, ktoré príklady majú byť označené (napr. najneistejšie)