Univerzita Hradec Králové
Fakulta informatiky a managementu

Katedra informačních technologií

Data x Process Mining

Bakalářská práce

Autor: Varvara Chikina

Studijní obor: Informační management

Vedoucí práce: doc. Ing. Hana Tomášková, Ph.D.

Hradec Králové Září 2021

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně a s použitím uvedené literatury.

V Hradci Králové dne Varvara Chikina

Poděkování

Chtěla bych poděkovat své vedoucí bakalářské práce doc. Ing. Haně Tomáškové, Ph.D. za odborné vedení, za pomoc a rady při zpracování této práce.

Anotace

Bakalářská práce se zabývá algoritmy relativně nových disciplín Process Mining, přemosťující mezeru mezi Process Science a Data Science a Data Mining. Hlavním cílem práce je popsat a ukázat rozdíl algoritmů, používaných v procesní a datové analýze. Za tímto cílem se skrývá teoretické přiblížení tématu spočívající v analýze obecných principů, vymezení postavení Process Miningu vůči Data Miningu. Dalším úkolem je jasně určit oblasti praktického použití těchto moderních metod analýzy dat. Práce se skládá ze čtyř hlavních fází, kterými je systém práce s daty jako Data Science a Business Intelligence, popis Data Mining a Process Mining a jejích algoritmů. Hlavním výstupem této práce je pochopení rozdílu mezi Data a Process miningem a jejich algoritmů.

Klíčová slova: Data Mining, Process Mining, Data Science, Algoritmy DM, Algoritmy PM

Annotation

The bachelor thesis deals with algorithms of the relatively new disciplines of Process Mining, bridging the gap between Process Science and Data Science and Data Mining. The main objective of the thesis is to describe and show the difference of algorithms used in process and data analysis. Behind this goal is a theoretical approach to the topic consisting in the analysis of general principles, defining the position of Process Mining in relation to Data Mining. Another objective is to clearly identify the areas of practical application of these modern methods of data analysis. The thesis consists of four main phases, which are the system of working with data as Data Science and Business Intelligence, the description of data through Data Mining and Process Mining and their algorithms. The main output of this thesis is to understand the difference between Data and Process mining and to analyze their algorithm.

Keywords: Data Mining, Process Mining, Data Science, DM Algorithms, PM Algorithms

Obsah

Úvod

Vyhledáváním informací v rámci pátrání po zákonitostech v nich skrytých se lidé zabývají po mnoho staletí. Ale teprve s příchodem počítačů, databází, lokálních a globálních sítí získal pojem "velká data" (big data) současný význam. Jejich zkoumání a skenování které dříve zajímalo jen vyzvědače a kabalisty-mystiky, později sociology kultury a teoretiky médií s jejich vášní k obsahové analýze, se vyvinulo v důležitý průmysl. V moderní době je každému jasné, že tak jako ropa nebo zlato, patří i data k jedním z nejcennějším zdrojům.

Mezinárodní datová společnost (IDC) předpověděla, že v roce 2020 vzroste digitální vesmír na 40 000 exabajtů (nebo 40 bilionů gigabajtů).[1] V roce 2020 bylo vytvořeno nebo replikováno 64.2 ZB dat. „V průběhu pandemie Covid-19 vzrostl systémový tlak na mnoho průmyslových odvětví a její dopad se projeví za několik let“[1] řekl Dave Reinsel, senior viceprezident společnosti IDC Global DataSphere. „Množství digitálních dat vytvořených v příštích pěti letech bude větší než dvojnásobek množství dat vytvořených od příchodu digitálního úložiště“.[2]

V konečném důsledku před byznysem a lidstvem nestojí otázka akumulace, ale zpracování a hledání cenných informací v tomto neuvěřitelném objemu dat.

Cílem této předložené práce je přiblížení obecných principů, vymezení postavení Process Miningu vůči Data Miningu. V rámci vymezení je taky popsaní a analýza algoritmů, které se používají při zpracování dat a proces mining. Navzdory skutečnosti, že v procesu zpracování dat metodami Data a Proces Mining existují kroky k jejich zpracování, tato práce klade důraz na pochopení algoritmů. Proto předpokládáme, že data jsou již maximálně vyčištěna a připravena k analýze.

1 Systém práce s daty

V této kapitole bude nejprve vysvětlen skutečný pohled na systém práce s daty, tj. jak jsou navzájem spojeny terminy, s nimiž se v posledních letech můžeme poměrně často setkat v oboru spravování dat. Dále vysvětleny pojmy, které se nejčastěji používají jako: Data Science, KDD, Data Mining, Process Mining a Business analýza a jaký je mezi nimi rozdíl.

1.1 Úrovně informací

V souvislosti s řešením obchodních problémů diskutujeme o Data science (věda o datech) a Business Intelligence (BI). Jak Data science, tak Business Intelligence zahrnují sběr dat, modelování a shromažďování informací.

Ale nejprve si musíme ujasnit některé důležité pojmy. Russel Ackoff zvýraznil několik úrovní informací:

Data – jsou symboly, které představují vlastnosti objektů, událostí a jejich prostředí. Jsou to produkty pozorování. Technologie snímání, přístrojového vybavení je samozřejmě velmi rozvinutá. Data nepředstavují hodnotu, dokud nebudou zpracována v použitelné (tj. aktuální) podobě. [3 s. 3–6]

Informace – jsou získávány z dat analýzou v mnoha aspektech, které počítače poskytují. Myšlenkou je, že se lidé ptají na otázky jako kdo, co, kde, kdy a kolik, a data nabídnou patřičný výsledek. Informační systémy generují, ukládají, extrahují a zpracovávají data. V mnoha případech jejich zpracování má statistický nebo aritmetický charakter. Jsou informace vyvozeny z dat. [3 s. 3–6]

Rozdíl mezi daty a informacemi je funkční, nikoliv strukturální, ale údaje se obvykle snižují, když jsou převedeny na informace.

Znalosti – jsou know-how, například jak systém funguje. To je to, co umožňuje konverzi informací do pokynů a správu systému. Do všech řídicích systémů jsou zabudovány systémy znalostní. Znalosti lze získat dvěma způsoby: buď přenosem z jiného zdroje, podle návodu, nebo čerpáním ze zkušenosti. Znalosti jsou přenášeny pomocí pokynů, odpovědí na praktickou otázku "proč". Získávání znalostí je učení. [3 s. 3–6]

Moudrost – je schopnost naučené koncepty a znalosti z předchozích situací aplikovat na problémy nové.

2 Business intelligence

Termín Business Intelligence (BI) poprvé definoval Howard Dresner takto:

„*Business Intelligence je množina konceptů a metodik, která zlepší rozhodovací proces za použití metrik, nebo systémů založených na metrikách. Účelem procesu je konvertovat velké objemy dat na poznatky, které jsou potřebné pro koncové uživatele. Tyto poznatky potom můžeme efektivně použít například v procesu rozhodování a mohou tvořit velmi významnou konkurenční výhodu*“.[4]

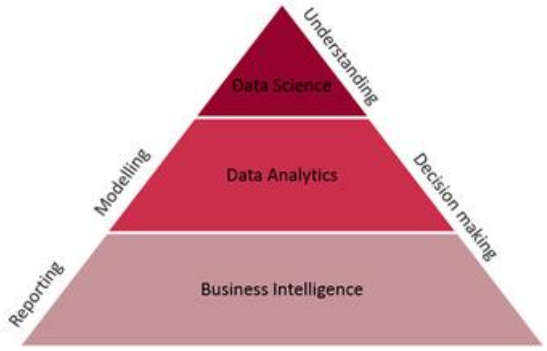
Business Intelligence je analýza dat společnosti se statistickými koncepty pro získání řešení a nápadů. V Business Intelligence se používají pouze strukturovaná digitální data, umožňující velmi omezený obraz okolního světa. Ta řeší specifické problémy, jako jsou náklady, zisk a podobně a jejích výsledným produktem je informace.

“*Úspěch nástrojů BI však závisí na kvalitě dat, která používá. Kvalitní data, jejich transformace na informace a získávání znalostí z nich jsou proto nezbytné pro úspěšnou implementaci BI. V důsledku toho je důležité prozkoumat techniky, které lze implementovat pro výběr a analýzu organizačních dat. Objev znalostí v databázích (KDD) je jeden proces, který lze prozkoumat, aby byla zajištěna nejvyšší kvalita dat pro aplikace BI*.“[5]

3 Data Science

Data Science je poměrně nová disciplína. Podobu samostatné disciplíny dostala někdy v roce 2010, proto je terminologie často matoucí. Rychlé vyhledávání na internetu o data science vede k několika termínům a definicím:

Data science založena na několika disciplínách včetně informatiky, matematiky, statistiky, ukládání a zpracování dat a jejich prediktivní analýzy.[6] Kombinuje data s budováním algoritmů a technologií, aby odpověděla na celou řadu specifických otázek, jako jsou například dopad geografie či sezónních faktorů a preferencí zákazníků na byznys. Úkolem toho, kdo se zabývá data science, je extrakce znalostí s využitím technik, spojených pod společným názvem Data mining: spojení statistiky, strojového učení a umělé inteligence a dalších metod analýzy dat za účelem pochopení toho, co data obsahují. V data science mohou být použity všechny údaje, které jsou dostatečné pro obraz okolního světa, s veškerou požadovanou úplností. Pro data science je výsledným produktem znalost.



Obrázek 1: Pyramida BI, převzato z [7]

Závěrem je třeba dodat, že BI a data science lze představit jako dva póly na společné ose datových technologií. Na jednom pólu se z dat získávají informace, na druhém znalosti. Stejně jako v mnoha případech v životě - hranice mezi nimi je nejasná.[8] Navíc tyto dva koncepty kombinují proces dobývání znalostí z databází.

4 Knowledge Discovery in Databases

Inteligentní analýza dat, známá také jako Dobývání znalostí v databázích (KDD, Knowledge Discovery in Databases). Tento termín byl uveden na prvním semináři KDD v roce 1989 [9] a byl popularizován v oblasti umělé inteligence a strojového učení.[10]

KDD je proces detekce vzorků ve velké sadě dat a datových úložišť. Tento pojem je definován jako:

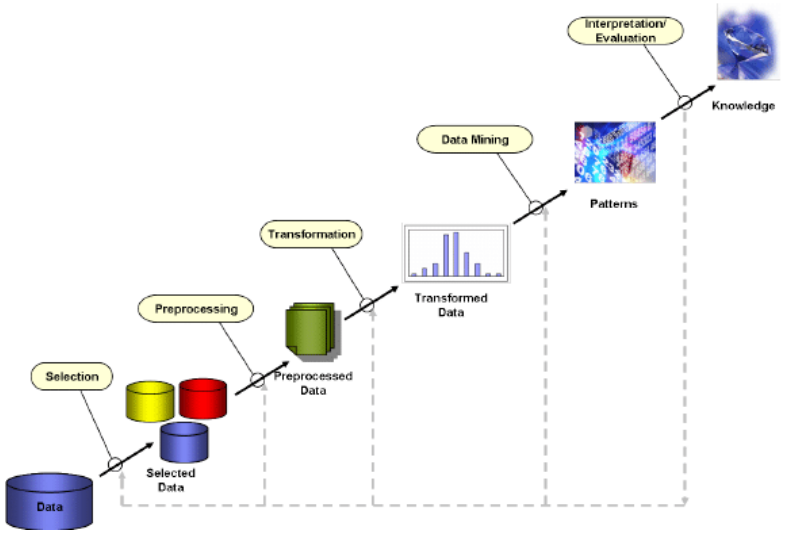
„*Proces netriviálního objevování implicitních, dopředu neznámých a potenciálně použitelných znalostí v datech*“.[10]

Podle této definice se někteří jedinci mohou mylně domnívat, že KDD je synonymum pro „data mining“, ovšem není tomu tak.

Data mining je pouze jedním krokem z celkového procesu dobývání znalostí z databází, a to jako proces vyhledávání požadovaných dat a jejich poskytování uživateli/procesu k dalšímu využití.

4.1 KDD process

Existuje řada variant KDD procesu, jako jsou ty, jenž byly publikovány u Adriaans & Zantinge v roce 1996, Brachman & Anand v roce 1996, a Han A Kamber v roce 2006 kromě jiných. Nicméně, všechny varianty procesu KDD jsou si blízké, jak to popsal Fayyad v roce 1996[10] a podporovány Roiger [6]:



Obrázek 2: Steps in the KDD process, převzato z [11]

Určení cíle. V této fázi se zaměříme na pochopení oblasti, která se zabývá zjištěním znalostí a relevantních předběžných znalostí. Stanovení cíle procesu KDD z pohledu zákazníka, který musí být dosažen. Je možné formulovat hypotézu, jež naznačuje pravděpodobný nebo požadovaný výsledek.

Vytvoření cílové datové sady. Výběr datové sady nebo zaměření se na podmnožinu proměnných nebo datových vzorků, pro které musí být detekce provedena.

Čištění a Integrace dat. Tato fáze se počítá za předzpracování získaných údajů, které jsou uloženy v datovém úložišti. V této fázi mohou data obsahovat vynechání nadbytečných dat, abnormální hodnoty atd.

Transformace dat. V této fázi jsou data zkrácena, převedena a kombinována do forem vhodných pro dolování, vytěžování dat prováděním operací generalizace nebo agregace, aniž by byla ohrožena jejich integrita. Informace jsou organizovány a tříděny pomocí metod snižování velikosti nebo konverze, často kombinovány do jednoho typu, aby se snížil efektivní počet uvažovaných proměnných. Data jsou plně použitelná. Někdy se konverze a konsolidace dat provádí před procesem výběru dat, zejména v případě datového skladu.

Data Mining. Více o této fázi bude napsáno v následující kapitole. Stručně pojem Data Mining v sobě zahrnuje:

- Výběr konkrétní metody dolování dat, odpovídající cíli procesu KDD, vybranému v prvním bodu.
- Výběr algoritmu pro dolování dat, výběr metody, která má být použita pro vyhledávání patternů.
- Vlastní Data Mining, hledání zajímavých modelů v určité srozumitelné podobě nebo sadě takových reprezentací: jako klasifikační pravidla nebo stromy, regrese, shlukování a tak dále.

Správné provedení předchozích kroků může výrazně usnadnit metodu dolování dat.

Komponenta data mining pro proces KDD se zabývá algoritmickými prostředky, kterými jsou vzory extrahovány a vyčísleny z dat.

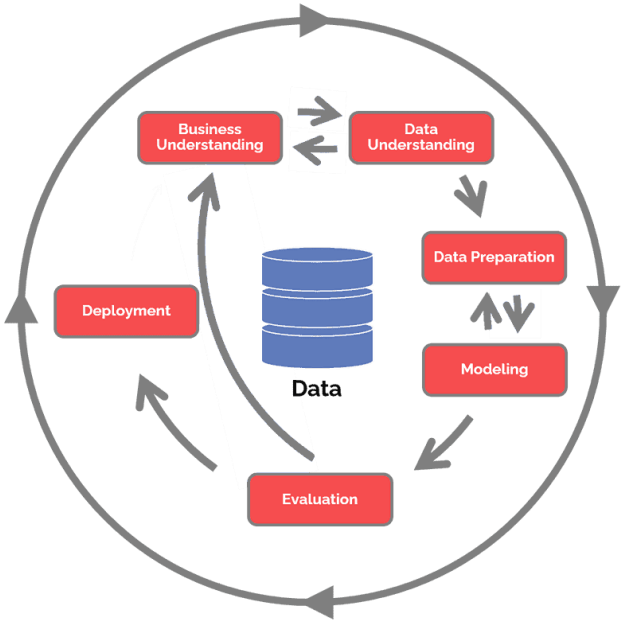
Interpretace / Hodnocení. Často tento krok může zahrnovat vizualizaci extrahovaných dat s ohledem na modely. Probíhá hodnocení vzorků, které představují

znalosti založené na ukazatelích zájmu. Interpretací přijatých vzorů je možné se vrátit k některému z předchozích kroků pro další iteraci.

Consolidating discovered knowledge. Konsolidace zjištěných znalostí: Pokud jsou zjištěné znalosti jsou považovány za užitečné, jsou zahrnuty do jiného systému pro další řešení problémů nebo akce, nebo jen dokumentace a jejich prezentace zájemcům.

Tento proces pouze o šesti kroků používá společnost Statistica, pod názvem CRISP-DM – Cross Industry Standard Process for Data Mining.

Jak je vidět na obrázku 3, proces není „jednosměrný“. Tato situace může například nastat, když výsledky modelování nejsou dostatečně věrohodné, nebo pokud je třeba data jinak transformovat.



Obrázek 3: Diagram CRISP-DM procesu, převzato z [12]

5 Data Mining

Jedním z prvních, kdo začal mluvit o Data Mining jako potenciálním zlatém dolu, byl Rakesh Agrawal:

„Jednou z takových aplikačních domén, která pravděpodobně v blízké budoucnosti získá značné významy, je dolování databází. Rostoucí počet organizací je vytváření ultra velkých datových základen (měřených v gigabajtech a dokonce terabajtech) obchodních dat, jako jsou spotřebitelská data, historie transakcí, záznamy o prodeji atd. Taková data tvoří potenciální zlatý důl cenných obchodních informací.“[13]

Výše v této práci, jako v českém prostředí se v některých případech užívá českého výrazu dolování dat, ale v této práci budeme využívat pojmu Data Mining, jelikož se jedná o ustálený pojem.

5.1 Definice Data Mining

Jiawei Han ve své knize popisuje Data Mining, jako skutečně interdisciplinární předmět, který může být definován mnoha různými způsoby. Srovnává jej s dobýváním zlata z písku nebo hornin. Stejně tak by Data Mining mohl být vhodněji nazýván "inteligentní analýza znalostí založených na datech".[14] Ale to je příliš dlouhý název, a proto se nejčastěji můžeme setkat s definicí “proces hledání zajímavé struktury v datech“.[6] Struktura může mít mnoho forem: sady pravidel, grafů nebo sítě, stromu, jedné nebo více rovnic a mnoho dalších. Data Mining používá jeden nebo více algoritmů s cílem identifikovat zajímavé trendy a vzory v datech.

Je třeba poznamenat, že tato definice znamená, že užitečné informace jsou již obsaženy v datech. Nicméně, složitost dat a jejich vícestupňová povaha dat znamenají, že užitečné informace nemůžeme najít bez výkonných matematických nástrojů.

Jednou z důležitých skutečností definice dolování dat je, že nezahrnuje žádný postup učení. Proto jsou metody nebo algoritmy používané pro dolování dat nekontrolovatelné.[15]

Jak bylo napsáno výše, Data Mining hledá znalosti v již dříve zpracovaných, strukturovaných datech. Na kvalitě a zdroji těchto informací závisí výsledek.

Klíčové vlastnosti dolování dat:

- Automatická předpověď vzorků na základě analýzy trendů a chování,
- Prognóza založená na pravděpodobných výsledcích,
- Vytváření informací zaměřených na rozhodování,
- Zaměření na velké datové sady a databáze pro analýzu,
- Clustering založený na vyhledávání a vizuálním dokumentování skupin faktů, která nebyla dříve známa.

5.2 Klasifikace úloh Data Mining

Úkoly řešené těžbou dat:

- **Klasifikace** – používá klasifikovaná data a na jejich základě se snaží předpovědět, do jaké třídy by měla být nová data připsána.
- **Clustering** – základní koncept clusteringu je velmi jednoduchý: podobné případy jsou seskupeny společně a případy, které se liší, jsou odděleny. Dobrá segmentace vytváří klastry, které jsou co nejkompaktnější (distance mezi klastry v rámci skupiny jsou minimalizovány) a maximalizuje vzdálenosti mezi různými skupinami (existuje dobré oddělení mezi skupinami odlišných případů).[16]
- **Zkrácení popisu** – pro vizualizaci dat, zjednodušení účtu a interpretaci, kompresi objemů shromážděných a uložených informací.
- **Asociace** – hledání opakujících se vzorků. Například hledání "udržitelných vazeb v nákupním koši".
- **Predikce** – zjištění budoucích stavů objektu na základě předchozích stavů (historických dat).
- **Analýza odchylek** – například detekce atypické síťové aktivity umožňuje detekovat malware.
- **Vizualizace dat**.

5.3 Základní pojmy

Pro lepší pochopení strategie Data Mining a používaných algoritmů je nejprve třeba nastavit některé základní pojmy používané při jejich popisu.

5.3.1 Klasifikace proměnných

Zde vysvětlím důležitý rozdíl mezi typy dat, které mohou být kvantitativní a kvalitativní, nominální a pořadové, diskrétní a spojité.

- **Kvantitativní (numerická) data**
  Proměnné jsou spojeny s vnitřně číselnými veličinami, jako jsou věk a příjem. Je možné navázat spojení a numerické vztahy mezi jejich úrovněmi. Lze je rozdělit na diskrétní kvantitativní proměnné a spojité kvantitativní proměnné. [17]
- **Diskrétní proměnné**
  Diskrétní data jsou data, jejichž počet je konečný nebo nekonečný, ale lze je vypočítat pomocí přirozených čísel od jednoho do nekonečna. Diskrétní jsou všechna data typu řetězce nebo logiky. Diskrétní mohou být číselná data, například, "Kód produktu" nebo počet telefonních hovorů přijatých za den. Data přijímající hodnotu typu **integer**, jsou diskrétní, protože aritmetické operace nad tímto ukazatelem nemá smysl.
- **Kontinuální (spojité) proměnné**
  Jsou data, která mohou v určitém intervalu přijímat libovolné hodnoty. Nad takovými hodnotami lze provádět aritmetické operace a mají smysl. Příklady dat jsou: výška, hmotnost, množství zboží, roční příjmy společnosti atd.

Typy dat	Kontinuální	Diskrétní
Reální (0,9)	•	•
Celý (87)	•	•
Řetězce (abc)		•
Logický (1/0)		•
Data/čas	•	•

Tabulka 1. Typy dat vs Proměnné, převzato z [18 s. 176–177]

- **Kvalitativní proměnné**

  Kvalitní data patří k datům, která "nejsou čísla". Nabývá hodnot „druh auta“, například: osobní auto, náklaďák atd. Jinými slovy, není to, co počítáme nebo měříme s pevnou stupnicí nebo složitými statistikami a matematikou. Dále kvalitativní proměnné lze rozdělit na nominální a ordinální.
- **Nominální proměnné**

  Jsou z různých kategorií, není možné jejich hodnoty seřadit. Jediná možnost porovnání je určit, zda se hodnoty rovnají, nebo ne.[17]
- **Ordinální proměnné**

  Jsou z jedné kategorie a data lze uspořádat od nejmenšího po největší. Například velikosti triček: velikost S, M, L, XL atd.
- **Kategorické proměnné**

  Obecně platí, že jakýkoliv atribut dat, který je kategorický, představuje diskrétní hodnoty, které patří do určité konečné sady kategorií nebo tříd. Proměnné dat mohou být klasifikovány nebo seskupeny. Například kurz, pohlaví, adresa URL stránky.

Obecně platí, že jakýkoliv standardní pracovní postup zahrnuje určitou formu přeměny kategorických hodnot na číselné značky a poté použití nějakého schématu kódování na tyto hodnoty.

5.3.2 Metody učení

Supervised learning neboli učení s učitelem. Nejznámější a nejpoužívanější technika učení v Data Miningu. Je to směr strojového učení, který spojuje algoritmy a metody modelování na základě mnoha příkladů, které obsahují dvojice „známý vstup — známý výstup“. Jinými slovy, aby algoritmus patřil k učení s učitelem, musí pracovat s příklady, které obsahují nejen vstupní proměnné (atributy, znaky), ale také cílové hodnoty, které by měl obsahovat model po vyučování. Rozdíl mezi cílovým a skutečným výstupem modelu se nazývá chyba učení, která se minimalizuje v procesu učení a působí jako „učitel“. Hodnota výstupní chyby se pak používá k výpočtu korekce parametrů modelu na každé iteraci učení.

Mezi algoritmy učení s učitelem pro řešení úkolů klasifikace patří:

- rozhodovací stromy,
- stroje nosných vektorů,
- bayesovský klasifikátor,
- lineární diskriminační analýza,
- metoda k-nejbližší sousedé.

Unsupervised learning neboli učení bez učitele. Je technologie strojového učení, ve které se pro korekci parametrů modelu nepoužívá cílová hodnota. Jinými slovy, v učebních příkladech při výuce bez učitele není nutné mít předem dané výstupy modelu.

Hlavním uplatněním výuky bez učitele je vytváření clustering modelů. Vzhledem k tomu, že clusterová struktura dat je předem neznámá a je určena v procesu učení modelu, není možné použít žádné cílové hodnoty.

5.3.3 Metody dolování dat

Metody dolování dat se mohou rozdělit do tří základních skupin:

- **Predikce** – tyto metody využívají dříve získaných znalostí. Technika těchto metod spočívá v porovnávání dat, které máme k dispozici, se vzorovými daty. Na základě podobností se pak stanovuje předpoklad dalšího vývoje.
- **Deskripce** – cílem těchto metod je analyzovat data a zkoumat nové vzájemné vztahy, které dosud nebyly objeveny a jsou potencionálně užitečné.
- **Indikace** – tyto metody odhalují odchylky od normálního stavu, aby se včas zabránilo např. poruše zařízení.

5.4 Algoritmy Data Mining

Algoritmus v Data Miningu (nebo strojovém učení) je sada heuristiky a výpočtů, která vytváří model z dat. Chceme-li vytvořit model, algoritmus nejprve analyzuje poskytnutá data a hledá konkrétní typy vzorů nebo trendů. Algoritmus využívá výsledky této analýzy v mnoha iteracích k nalezení optimálních parametrů pro vytvoření důlního modelu. Tyto parametry jsou pak použity v celé datové sadě pro extrakci žalovatelných vzorů a podrobných statistik.[19]

Jeden z klíčových problémů, kterému výzkumný pracovník čelí při vývoji statistického modelu studovaného jevu, nastává při výběru extrakce vzorců optimálních pro konkrétní případ algoritmu. Během několika posledních desetiletí bylo vyvinuto obrovské množství metod pro řešení úkolů klasifikace a regrese, což jistě značně ztěžuje tuto volbu.

Úkolem klasifikace a regrese je určit hodnotu závislé proměnné objektu podle jeho nezávislých proměnných. Pokud závislá proměnná přijímá číselné hodnoty, pak mluví o úkolu regrese, jinak – o úkolu klasifikace.

5.4.1 Cross-validation

Pro posouzení přesnosti klasifikace se provádí křížová kontrola. Cross-validation je postup pro hodnocení přesnosti klasifikace dat z testovací sady. Přesnost kvalifikace testovacího souboru je srovnávána s přesností výcvikového souboru. Pokud klasifikace testovacího souboru poskytuje přibližně stejné výsledky přesnosti jako klasifikace výcvikového souboru, předpokládá se, že tato možnost prošla křížovou kontrolou. Rozdělení na učební a testovací množiny se provádí dělením vzorku v určitém poměru, například učební množina – dvě třetiny dat a zkušební – jedna třetina dat. Tato metoda by měla být použita pro vzorky s mnoha příklady.

„*Nejdůležitější algoritmickou částí je použití algoritmů strojového učení, tedy budování modelu; pro data majningový systém je to stejně důležité jako motor pro sportovní auto. Hlavní úsilí však obvykle trvá na přípravě dat*“.[20]

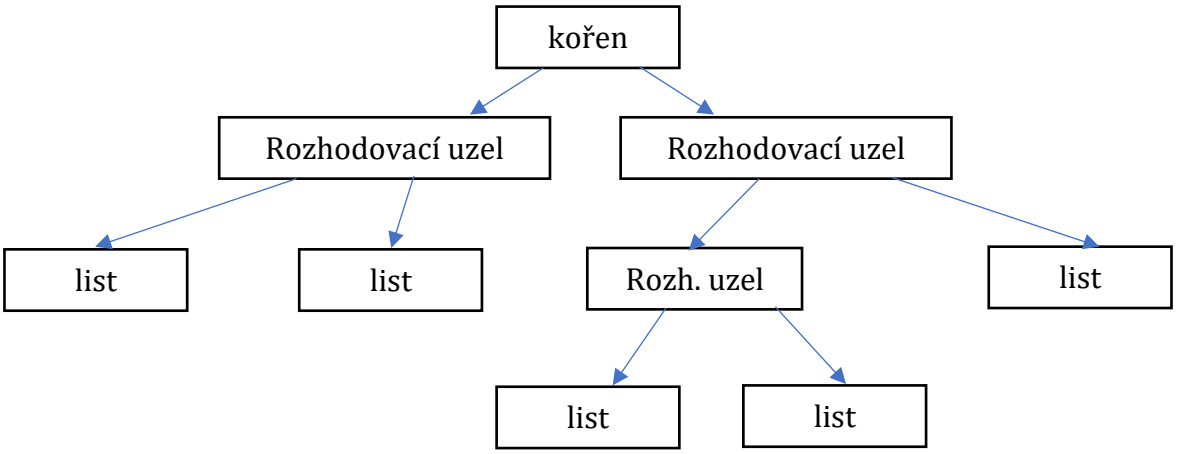
5.5 Rozhodovací stromy

Jeden z nejpopulárnějších řešení pro získaní znalosti. Rozhodovací strom představuje soubor klasifikačních pravidel ve stromové podobě.

Ve stromu řešení je několik proměnných prediktorů a na základě těchto proměnných prediktorů se snažíme předpovědět, co je takzvaná proměnná odezvy. Řešení pomoci rozhodovacích stromů je forma učení s učitelem, protože data jsou označena.

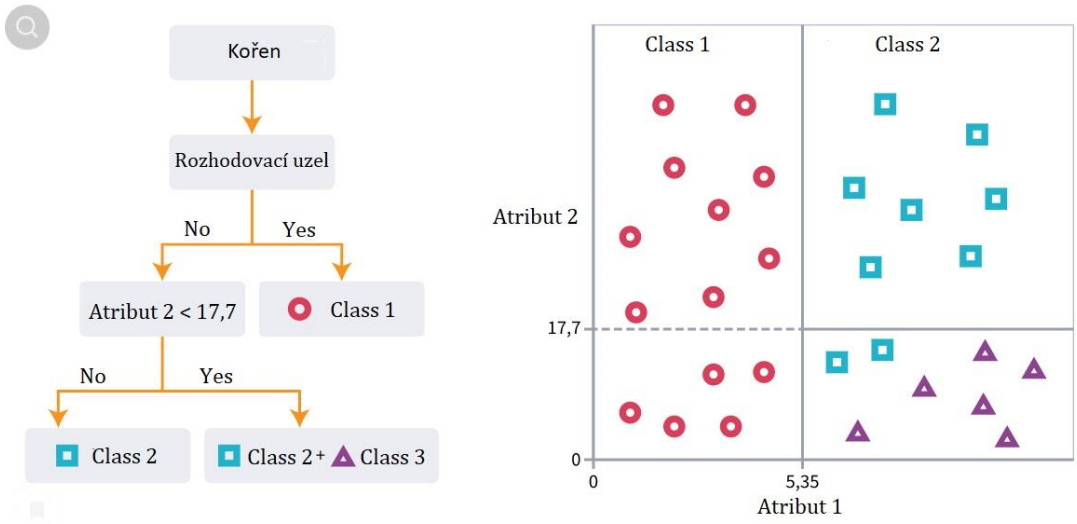
Hlavním úkolem při implementaci stromu řešení je určit, jaké atributy bychom měli považovat za kořenový uzel a každou úroveň. Zpracování by mělo být známé jako výběr atributů.

Ve stromu řešení algoritmus rozdělí datovou sadu na podmnožiny na základě nejdůležitějšího nebo významného atributu. Nejvýznamnější atribut je označen v kořenovém uzlu, a to je místo, kde rozdělení zaujímá místo celé datové sady přítomné v kořenovém uzlu. Takové rozdělení je známé jako rozhodovací uzly. V případě, že další rozdělení není možné, tento uzel se nazývá koncový uzel.[21]



Obrázek 4: Typický tvar stromu, samostatně zpracované.

Rozhodovací strom je lineární klasifikátor, tj. vytváří rozdělení objektů ve vícerozměrném prostoru rovinami (ve dvourozměrném případě čarami).



Obrázek 5: Rozdělení objektů čárami ve dvourozměrném prostoru, vlastně zpracované na základě [21]

Nejpopulárnější algoritmy pro rozhodovací stromy jsou ID3, CART a C4.5. Myšlenkou všech algoritmů učení je rozdělit atributový prostor, dokud není dosaženo určitého kritéria dokončení v každém listu. Obvykle je kritériem, že všechny body na listu patří jedné třídě.

Rozhodovací stromy mají velké výhody: jsou jednoduché a snadně pochopitelné. Mohou pracovat se smíšenými proměnnými (jak číselnými, tak i kategoriálními). Postup je rekurzivní, s jehož pomocí se soubor z **n** statistických jednotek postupně rozdělí na skupiny v souladu s pravidlem dělení, které je zaměřeno na maximalizaci homogenity v každé ze získaných skupin. Dělení se provádí postupně vždy podle jednoho atributu. Před každým dělením musí být zhodnoceno, který atribut má největší schopnost odlišit objekty jednotlivých tříd.[6]

Výběr atributů provedených při stavbě rozhodovacího stromu určuje velikost postaveného stromu. Hlavním cílem je minimalizovat počet úrovní stromu a uzlů stromu, čímž maximalizujete zobecnění dat.

Výběr algoritmu je také založen na typu cílových proměnných.

5.5.1 ID3

Iterative Dichotomiser 3. V analýze dat a strojovém učení je ID3 jedním z nejoblíbenějších algoritmů učení rozhodovacích stromů.

Byl vyvinut v roce 1986 Rossem Quinlanem. Algoritmus vytváří vícestupňový strom tím, že najde pomoci rekurzivního rozdělení pro kořenový a každý další uzel (tj. chamtivým způsobem) kategorický znak, který poskytne největší informační zisk pro kategorické účely. Stromy rostou na svou maximální velikost, a pak se obvykle aplikuje fáze prořezávání, aby se zlepšila schopnost stromu zobecnit neviditelná data.

Chamtivý způsob – to znamená, že pokud byla jednou vybrána proměnná a došlo k rozdělení na podmnožiny, algoritmus se nemůže vrátit zpět a zvolit jinou proměnnou, která by dala lepší rozdělení. Proto ve fázi výstavby nelze říct, zda zvolená proměnná nakonec poskytne optimální členění.

Rozdělení v každém uzlu stromu se provádí podle určitého atributu z učební sady. Vzhledem k tomu, že atributů je několik, při každém rozdělení je nutné vyřešit úkol výběru nejlepšího atributu. Za nejlepší atribut bude považován ten, který bude poskytovat maximální možné zvýšení homogenity tříd v podřízeném uzlu relativně nadřízené, nebo jedno a totéž, maximální snížení entropie.

5.5.1.1 Entropie

Jedná se o míru nepořádku a entropie datové sady je měřítkem nepořádku v cílovém objektu datové sady. V případě binární klasifikace (tam, kde rozhodovací uzel má pouze dva typy tříd) je entropie rovna 0, pokud jsou všechny hodnoty homogenní (podobné), a naopak se bude rovnat 1, pokud cílový sloupec má stejné číselné hodnoty pro obě třídy. Čím vyšší je homogenita podmnožiny, tj. čím více příkladů jedné třídy a méně "příměsi" příkladů jiných tříd, tím menší entropie a tím lepší výsledky klasifikace.

Entropie se vypočítá jako:

$$H = -\sum_{i=1}^{m} P_i \log P_i$$

P_i — pravděpodobnost i – stavu systému (hodnoty přijaté proměnné),

m — počet stavů systému (hodnoty přijatých proměnnou).

Entropie se měří v bitech, natech (natural units) nebo ditech (desetinná čísla). [14]

Tedy, při řešení úkolů kategorizace, během níž dojde k důslednému rozdělení zdrojového datového souboru na podmnožiny podle určitému rysu, bude nejlepší takové rozdělení, které zajistí minimální entropie výsledné podmnožiny. Pro diskrétní a kontinuální atributy je proces odlišný.

5.5.1.2 Diskrétní atribut

Vypočítá se snížení entropie v uzlu T v důsledku rozdělení podle atributu A, pomoci formule **infoGain**:

$$Gain(A) = Info(T) - Info(T,A)$$

Info(T) — informace před rozdělením

Info(T,A) — informace po rozdělení podle atributu A

Zvolený atribut, rozdělení, které zajistí největší zisk informací

5.5.1.3 Kontinuální atribut

Atribut, podle kterého je rozdělení kontinuální, je nejprve převeden na diskrétní druh, například pomocí kvantizační operace.

To je takový proces zpracování dat, který převádí kontinuální data na diskrétní nahrazením hodnot segmenty, z nichž každý představuje určitý rozsah. Lze použít ke snížení počtu různých hodnot. Obvykle se vyskytuje rozdělení na intervaly nebo se stejnou vzdáleností nebo rovným počtem kopií, v každém rozsahu. Například věková kategorie 0-5 bude kategorie 1, 6-11 bude kategorie 2 a tak dále.

Ve většině případů použití entropie k určení významu atributů vykazuje dobré výsledky. Problémy vznikají, když atribut má širokou škálu jedinečných hodnot. V tomto případě je rozhodovací strom náchylný k rekvalifikaci.

5.5.2 C4.5

C4.5 je nástupcem ID3 a odstranil omezení, že funkce musí být kategoriální, dynamicky definuje binární atribut (na základě číselných proměnných), který rozdělí trvalé hodnoty atributů na diskrétní sadu intervalů. C4.5 převádí vyškolené stromy (tj. výstupy algoritmu ID3) na sady pravidel **if-then**. Tato přesnost každého pravidla je

pak hodnocena tak, aby určila pořadí, ve kterém mají být použita. Oříznutí se provádí odstraněním předběžného pravidla, pokud se přesnost pravidla zlepší bez něj.

Základní myšlenka algoritmu C4.5 spočívá v tom, že pro každý bod výběru ve stromu tento algoritmus vybere atribut, který sdílí data takovým způsobem, aby odhalil největší nárůst informací. [6]

Příklad

Nechte nastavit učební sadu ***T*** obsahující *m* atributy a *n* příklady. Pro řadu ***T*** je definováno *k* třídy C_1, C_2, C_3…C_k

V prvním kroku učení se vytvoří "prázdný" strom, který se skládá pouze z kořenového uzlu, jenž obsahuje všechny vzdělávací sady. Je nutné rozdělit kořenový uzel na podmnožiny, ze kterých budou vytvořeny uzly-potomci. K tomu je vybrán jeden z atributů a jsou vytvořena pravidla, která rozdělují učební množinu na podmnožiny, jejichž počet se rovná počtu *p* jedinečných hodnot atributu.[22]

Chceme-li vybrat atribut, podle kterého bude nejlepší spustit strom, algoritmus používá informace o počtu příkladů všech tříd v učební množině a v každé výsledné podmnožině.

Nechme pravidlo rozdělení, které používá atribut *A*, aplikovat na učící se množinu *N,* který přijímá hodnoty a_1, a_2, a_3…a_p. Výsledkem bude vytvoření *p* podmnožin *S*, kde budou distribuovány příklady, ve kterých atribut *A* přebírá příslušnou hodnotu. Toto už dávno známe ze statistiky. K tomu používáme informace o počtu příkladů všech tříd v učební množině a v každé výsledné podmnožině.

$$P = \frac{N(C_j S)}{N(S)}$$

N (C_j S) – počet příkladů třídy C_j v množině *S*

N (S) - celkový počet příkladů v souboru *S*

A tak jako v předchozím algoritmu použijeme vzorec entropie.

$$\text{Info(S)} = -\sum_{i=1}^{m} \frac{N(Si)}{N(S)} \log \frac{N(C_j S)}{N(S)}$$

Stejný odhad získaný po rozdělení sady S podle atributu A lze zapsat jako:

$$\text{Info(S, A)} = -\sum_{i=1}^{k} \frac{N(C_j S)}{N(S)} \text{Info}(Si)$$

Pro výběr nejlepšího atributu větvení zase používáme kritérium **infoGain**:

$$\text{Gain(A)} = \text{Info(S)} - \text{Info(S,A)}$$

Popsaný postup se vztahuje na podmnožiny S_i a dále, dokud se hodnoty kritérií nepřestanou významně zvyšovat při nových poruchách, nebo bude splněna jiná podmínka zastavení.

Popsaný postup platí pro diskrétní atributy. V případě kontinuálních atributů algoritmus funguje poněkud jinak. Zvolí se práh, se kterým se budou porovnávat všechny hodnoty. Uspořádáním příkladů vzestupných hodnot atributu zjistíme, že každá hodnota rozděluje všechny příklady do dvou relativně stejných podmnožin.[22]

Pak jako hranici můžeme zvolit **průměr.** Důsledné uplatňování výše uvedených formulí na všechny potenciální prahy, vybereme ten, který dává maximální hodnotu podle **infoGain.**

Kritérium rozdělení založené na infoGain upřednostňuje atributy, které obsahují velké množství jedinečných hodnot. To je způsobeno tím, že se vytváří velké množství uzlů-potomků, což nakonec dává rovnoměrnější rozdělení tříd a tím i menší entropii rozdělení.[6] V mezním případě, pokud jsou všechny hodnoty atributu jedinečné, vytvoří se pro každý příklad samostatné pravidlo a samostatný list s jedinou třídou. V důsledku toho rozdělení přinese nulovou entropii.

Je to optimální z hlediska algoritmu, ale z praktického hlediska naprosto zbytečné, protože:

- význam pravidel bude extrémně nízký, protože pravidlo platí pouze pro konkrétní objekt,
- strom řešení bude velmi obtížné pochopit,
- strom řešení bude rekvalifikován, tj. nebude mít zobecňující schopnost.

5.5.2.1 Split

K vyřešení tohoto problému se používá **Split** - „pokuta“, která je překryta atributem pro počet potomků vytvořených pomocí uzlů. Algoritmus při výběru rozdělení nebude upřednostňovat atributy s velkým počtem jedinečných hodnot, tato „pokuta“ je prezentována jako koeficient pro hodnotu **Info(S).**

$$\text{Split Info(S)} = -\sum_{i=1}^{p} \frac{N(Si)}{N(S)} \log \frac{N(Si)}{N(S)}$$

Pak bude kritérium pro zisk informací s ohledem na metriku SplitInfo(S) bude vypadat:

$$GainRatio\ (S) = \frac{Gain\ (S)}{Split - Info(S)}$$

Význam této úpravy je poměrně jednoduchý. N(Si)/N(S) poměr počtu příkladů v podmnožině získané v důsledku rozdělení k počtu příkladů v nadřazeném souboru *S*. Pokud se v důsledku rozdělení objeví velký počet podmnožin s malým počtem příkladů, což je typické pro rekvalifikaci, pak se ukazatel **Split-Info** zvyšuje.

Gain-Ratio to je nárůst informací „pokutovaný“ pomocí Split-Info. Díky tomu je méně pravděpodobné, že atribut, pro který se Split-Info zvyšuje, bude vybrán pro rozdělení než při použití běžného InfoGain.

Aby se zabránilo vytváření uzlů s malým počtem výtisků, je třeba využít ještě jednoho pravidla: při rozložení skupiny *S* na alespoň dvě podmnožiny musí mít ne méně než nastavené minimální množství příkladů, obvykle se rovná 2.

5.5.2.2 Information Gain vs GainRatio

Pokud dva atributy s různým počtem možných hodnot (kategorií), mají stejnou entropii, Info Gain je nemůže odlišit (algoritmus rozhodovacího stromu vybere jednu z nich náhodně). Ve stejné situaci zisk poměr, bude upřednostňovat atribut s menšími kategoriemi.

Strategie Gain ratio vede k lepší zobecnění (méně nadhodnocování) DT modelů a je lepší použít Gain ration obecně.

I když bychom chtěli upřednostňovat atributy s více kategoriemi, Info Gain by nebyla dobrá volba, protože nerozlišuje mezi atributy s různým počtem kategorií.

5.5.3 CART

Classification And Regression Tree algorithm (CART) - Algoritmus klasifikačního a regresního stromu. Je to populární komerční algoritmus pro vytváření stromů klasifikace, které mohou pracovat jak s diskrétní, tak s kontinuální výstupní proměnnou, tj. řešit úkoly a klasifikace a regrese. Strom klasifikace je poddruh stromu řešení. Výsledkem práce stromu klasifikace je class.

Algoritmus staví binární rozhodovací stromy, které obsahují pouze dva potomky v každém uzlu. V průběhu práce dochází k rekurzivnímu rozdělení příkladů testových množin na podmnožiny, jejichž záznamy mají stejné hodnoty cílové proměnné.

Algoritmus implementuje učení s učitelem a používá jako kritérium pro výběr rozdělení v uzlech **index čistoty Gini** (Gini impurity index). V procesu růstu stromu algoritmus CART provádí pro každý uzel kompletní vyhledávání všech atributů, na základě kterých může být postavené rozdělení, a vybere ten, který maximalizuje hodnotu indexu Gini.

Základní myšlenkou algoritmu je vybrat takové rozdělení ze všech možných v tomto uzlu tak, aby výsledné podřízené uzly byly co nejhomogennější. V tomto případě každé rozdělení provádí pouze jeden atribut.

5.5.3.1 Gini impurity /Gini nečistota

Gini index je statistický ukazatel vyvinutý italským statistikem Corrado Ginim, pomocí kterého můžeme popsat povahu změny jedné velikosti vzhledem ke změně druhé. Nebo to můžeme popsat jinak: Gini index nám říká, jaká je pravděpodobnost nesprávné klasifikace pozorování. Hlavním uplatněním Gini indexu je posouzení nerovnoměrnosti rozdělení studovaného rysu (například ročního příjmu) pro různé sociální skupiny.

$$Gini = 1 - \sum_{k} (p_k)^2$$

p_k - podíl prvků označených třídou i v sadě

Při binárním rozdělení vypočítáme váženou částku nečistot každého výsledného rozdělení. Například, soubor *D* rozdělíme podle atributu *A* na oddíly *D1* a *D2*, index Gini *D* bude:

$$Gini(D, A) = \frac{|D1|}{|D|} Gini(D1) + \frac{|D2|}{|D|} Gini(D2)$$

Strategie je podobná strategii popsané dříve pro získání informací, kde střed mezi každou dvojicí (tříděných) sousedních hodnot je považován za možný dělený bod. Bod udávající minimální index Gini pro daný atribut (kontinuální hodnota) se bere jako bod rozdělení tohoto atributu. [14]

Snížení nečistot, které by vznikly binárním rozdělením na diskrétní nebo spojitý atribut A:

$$\Delta Gini(A) = Gini(D) - Gini(D, A)$$

Čím nižší je index Gini, tím lepší je rozdělení. Jinými slovy, **tím menší je pravděpodobnost nesprávné klasifikace**.

Algoritmus má následující výhody:

- Není statistický, takže nevyžaduje výpočet pravděpodobnostních distribučních parametrů,
- Atributy rozdělení jsou vybírány přímo v procesu budování stromu, takže není nutné provádět postup výběru proměnných pro model,
- Odolný vůči emisím a abnormálním hodnotám,
- Vysoká rychlost provozu.

Nevýhody algoritmu zahrnují nestabilitu týkající se dat: dokonce i malé změny ve výukovém souboru způsobují významné změny ve struktuře stromu řešení.

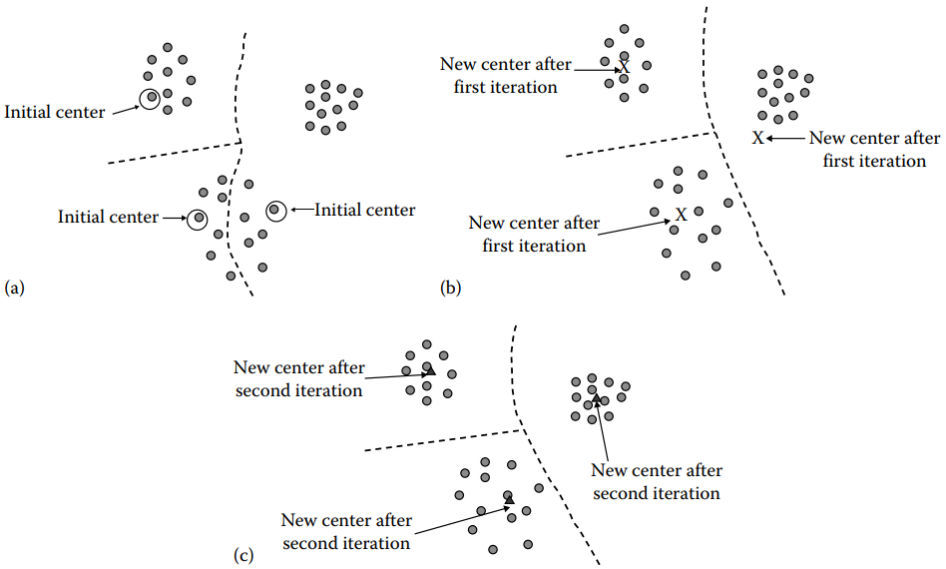
5.6 Analýza cluster

Na rozdíl od klasifikace a regrese, které analyzují datové sady s popisky tříd (výukové), clustering analyzuje datové objekty, aniž by musely patřit do označených tříd. V mnoha případech nemusí data označená třídou jednoduše existovat na začátku. Clustering lze použít k vytvoření štítků tříd pro datovou skupinu.

Skupiny vytváříme sdružováním objektů nebo pozorování do nepřeložitelných skupin, nazývaných clustery, na základě blízkosti hodnot jejich příznaků. Výsledkem je, že v každém klastru budou objekty podobné svými vlastnostmi a maximálně odlišné od objektů obsažených v jiných klastrech. Přitom čím větší je podobnost objektů uvnitř clusteru a čím silnější je jejich rozdíl od objektů v jiných klastrech, tím lepší clustering.[23]

Clustering umožňuje dosáhnout následujících cílů:

- Zlepšuje pochopení dat identifikací strukturálních skupin,
- Rozdělení datové sady do skupin podobných objektů umožňuje snadnější další zpracování a rozhodování tím, že do každého clusteru můžeme použít svou metodu analýzy,
- Umožňuje kompaktně reprezentovat a ukládat data. Chceme-li to provést, místo ukládání všech dat můžete z každého clusteru ponechat jeden typický dohled,
- Novinkou je detekce atypických objektů, které se nedostaly do žádného clusteru.



Obrázek 6: K-means výsledky pro nalezení tří klastrů v jednoduché datové sadě, převzato z [24 obr. 6.1]

Data Mining clustering se používá pro segmentaci zákazníků a trhů, lékařskou diagnostiku, sociální a demografický výzkum, stanovení bonity dlužníků a v mnoha dalších oblastech.

5.6.1 K-means

Metoda k-means vytváří skupiny *K* ze souboru objektů tak, aby součásti skupiny byly co nejvíce homogenní. Jedná se o populární techniku clusterové analýzy pro výzkum datové sady. Největší výhoda je v tom, že řekneme metodě k-means, kolik klastrů potřebujeme, a ona udělá zbytek práce.

Krátce jsou potřebné kroky rozepsané v knize „Data Mining Theories, Algorithms and Examples“ Nong Ye[25]:

1 Vybrat hodnotu pro *k* – celkový počet klastrů, které mají být určeny.
2 Vybrat instance k (datové body) v datovém souboru náhodně. Toto jsou počáteční clusterová centra – centroid.
3 Dál použijme euklidovskou vzdálenost k přiřazení zbývajících instancí k jejich nejbližšímu centrum clusteru.
4 V každém klastru použit instance pro výpočet nového průměru pro každý klastr.
5 Pokud jsou nové střední hodnoty identické se středními hodnotami předchozí iterace, proces končí. V opačném případě jako centra clusteru použijeme nové prostředky a opakují kroky 3-5.

5.6.1.1 Euklidovská vzdálenost

Představuje geometrickou vzdálenost v multidimenzionálním prostoru; v kroku 3 ji spočítáme pomocí:

$$d_{pq} = \sqrt{\sum_{i=0}^{n} (p_i - q_i)^2}$$

Někdy budeme chtít, aby se více blížila vzdálenějším objektům. V takovém případě použijeme mocninu Euklidovské vzdálenosti. Existují takové typy opatření vzdálenosti jako: vzdálenost městských čtvrtí (Manhattan vzdálenost), Chebyshev vzdálenost, Supremum vzdálenost.[14] Výběr vzdálenosti (kritérium podobnosti) leží zcela

na výzkumném pracovníkovi. Při výběru různých opatření se výsledky clusteru mohou výrazně lišit.

Algoritmus k-means pracuje pouze s reálnými daty. Pokud je v naší datové sadě kategorický atribut, měli bychom ji buď odmítnout, nebo převést hodnoty atributů na číselné ekvivalenty. Společným přístupem je vytvořit jeden číselný atribut pro každou hodnotu kategorického atributu.[6]

Problémy algoritmu k-means:

- Musíme předem znát počet klastrů,
- Algoritmus je velmi citlivý na výběr počátečních Center clusterů. Klasická varianta znamená náhodný výběr klastrů, což bylo velmi často zdrojem chyby. Možnost rozhodnutí, v počáteční fázi přijímat jako centra nejvzdálenějších bodů clusterů;
- nezvládá úkol, když objekt patří do různých clusterů stejně nebo není vlastněn žádným.

5.6.2 C-means

Poslední problém řeší C-means algoritmus. Namísto jednoznačné odpovědi na otázku, do kterého clusteru objekt patří, určuje pravděpodobnost, že objekt patří do určitého clusteru. [26 s. 209] Takže tvrzení „objekt a patří do clusteru 1 s pravděpodobností 90 %, do clusteru 2-10 %“ je pravdivé a pohodlnější.

5.6.3 EM

Algoritmus očekávaná (střední) hodnota–maximalizace (expectation-maximization). Obvykle se používá jako algoritmus clusteru (jako algoritmus k-means) pro detekci znalostí. Používá se k vyhodnocení maximální věrohodnosti při výpočtu modelu se skrytými proměnnými. Proměnné mohou být skryté (chybějící data). Existuje mnoho důvodů, proč mohou chybět (nejsou opraveny, ignorovány a tak dále).

V případě použití EM algoritmu na úkol clusteru, interpretace chybějících dat je velmi důležitá, protože nevíme, co je ta třída. Je pro nás důležité podívat se na třídu datového bodu jako na chybějící data. EM algoritmus přiřazuje značky třídy k datovým bodům stejně jako v clusteru.

Zpracování algoritmu prochází 2 kroky, popsané v knize „Data Mining Concepts and Techniques“[14 s. 412]:

1. Krok očekávání(E-step): Jako v k-means, vzhledem k aktuálním centrům clusteru, každý objekt je přiřazen klastru se středem, který je nejblíže objektu. Zde se očekává, že objekt bude patřit k nejbližšímu klastru.
2. Krok maximalizace (M-step): vzhledem k přiřazení clusteru pro každý klastr algoritmus upraví střed tak, aby součet vzdáleností od objektů přiřazených tomuto klastru a nového centra byl minimalizován. To znamená, že podobnost objektů přiřazených klastru je maximalizována.

Předchozí dva kroky se opakují, dokud se parametry modelu a rozdělení clusteru nevyrovnají. Algoritmus se zastaví, když se centra clusteru sbíhají, nebo je změna dostatečně malá.

5.7 Algoritmy klasifikace

Klasifikační modely předpovídají kategorické známky tříd. Úloha klasifikace má svým výsledkem diskrétní hodnotu 0 nebo 1.

5.7.1 K-Nearest Neighbors / KNN

Dva nejčastěji používané algoritmy jsou již výše zmíněný k-means a algoritmus k - nejbližších sousedů (kNN). Často jsou tyto dva navzájem zaměňovány kvůli přítomnosti *k* písmena, ale ve skutečnosti jsou tyto algoritmy od sebe poněkud odlišné.

Clustering k-means je tedy nekontrolovatelný algoritmus, který se primárně používá pro clustering, zatímco kNN je řízený klasifikační algoritmus, který bude dávat nové datové body podle k-počtu nebo nejbližších datových bodů a potřebuje k učení označená data.

KNN může klasifikovat nová, neoznačená data analýzou k-počtu nejbližších datových bodů. Proměnná kNN se tak počítá s parametrem, který bude stanoven strojním učitelem.

Každý algoritmus je určen k řešení různých úkolů a poskytuje jiný pohled na to, co změna znamená *K*.

Navzdory od k-means a c-means kNN nestaví žádný klasifikační model. Místo toho jednoduše ukládá označené tréninkové údaje. Když se objeví nové neoznačené údaje, kNN prochází 2 základními kroky:

1. Nejprve hledá k-počet nejbližším označeným datovým bodům – jinými slovy k nejbližším sousedům. Pro kontinuální data algoritmus používá dálkovou metriku, například Euklidovskou vzdálenost (metriku). Výběr metriky závisí na typu dat, jichž je poměrně hodně. Při práci s diskrétními daty, jsou nejprve převedena na kontinuální.[14]
2. Poté pomocí tříd sousedů kNN rozhodne, jak lépe klasifikovat nová data. Ale i tady jsou dvě možnosti:
    - Přijmout za správné rozhodnutí jednoduchou většinou. Do jaké třídy patří nejvíce sousedů, tam určuje datový bod,
    - Nebo udělat totéž, ale dát nejbližším sousedům větší váhu. Nejjednodušší způsob, jak to udělat, je použít kvantil vzdálenosti. Pokud soused vydrží 4 jednotky, pak jeho váha bude 1/4. S rostoucí vzdáleností se hmotnost stává menší a menší.

Jedním z nejdůležitějších kroků při práci s kNN algoritmem bude výběr hodnoty pro *K*.

5.7.1.1 Výběr hodnoty pro *k*

Řekněme, že celková velikost vzorku = n. Technicky můžeme nastavit *k* na libovolnou hodnotu od 1 do n. Pokud *k* = 1, objekt je jednoduše přiřazen třídě tohoto jediného nejbližšího souseda.[25] Jak se hodnota k zvyšuje, naše předpovědi se stávají stabilnějšími díky většinovému/zprůměrování, a proto jsou pravděpodobnější přesnější předpovědi (až do určitého bodu). Po překonání tohoto bodu nakonec začneme pozorovat rostoucí počet chyb. Právě v tomto okamžiku víme, že hodnota *k* je příliš daleko.

Aby bylo možné vybrat správné *k*, které je vhodné pro data, měli bychom spustit algoritmus kNN několikrát s různými hodnotami *k* a vybrat si *k*, které snižuje počet chyb, s nimiž se setkáváme, při zachování schopnosti algoritmu přesně dělat předpovědi, když dostává údaje, které předtím neviděl.

Algoritmus kNN je univerzální. Může být použit pro klasifikaci, regresi a vyhledávání. Ale zároveň má i své problémy: kNN může být velmi náročné na zdroje, pokud

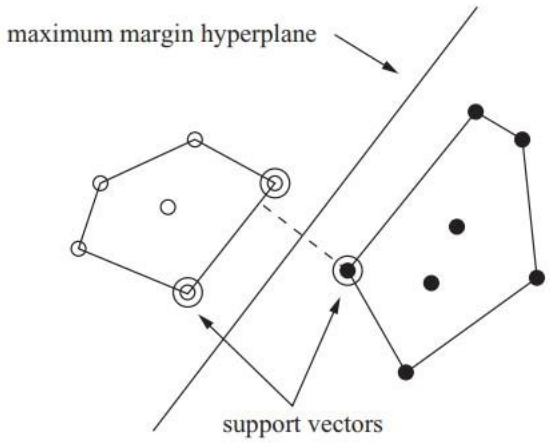
se pokouší identifikovat nejbližší sousedy na velké datové sadě. A jak bylo zmíněno, výběr správné hodnoty k je velmi důležitý pro přesnost kNN.

Je snadné si představit, jak se tento algoritmus používá pro nabídky filmů, které se mohou líbit v systému Netflix nebo Youtube, na základě těch, které již uživatel viděl. S největší pravděpodobností tyto společnosti používají efektivnější algoritmy, schopné zpracovávat obrovské údaje v podstatně kratším čase.

5.7.2 SVM

Support vector machines neboli metoda podpůrných vektorů, to je metoda sloužící pro klasifikační a regresní analýzu. SVM vznikl ze statistické teorie učení, poskytuje spojení mezi očekávaným zobecněním chyb pro danou trénovací množinu a vlastností třídění známou jako jeho kapacita.[27 s. 929] SVM algoritmus na začátku má stejné kroky jako algoritmus C4.5, ale nepoužívá stromy řešení.

Hlavním úkolem tohoto algoritmu je provést binární klasifikace dat na základě učení s učitelem. Ve dvourozměrném prostoru není možné rozdělit vektory dat jednou čarou tak, aby bylo minimální množství chyb. Proto je realizováno nelineární mapování do prostoru s vyšším rozměrem. Základní myšlenka klasifikátoru na referenčních vektorech je vytvořit oddělovací povrch pomocí pouze malé podmnožiny bodů ležících v zóně kritické pro rozdělení, zatímco ostatní jsou správně klasifikované pozorování; vzdělávací vzorky mimo tuto zónu jsou ignorovány.



Obrázek 7: Hyperplane s maximálním rozpětím, převzato z [28 obr. 6.9]

Na obrázku 7 je znázorněn princip rozdělení. Mezi všemi hyperplochami, které oddělují třídy, je hyperplocha s maximálním odsazením, která je, co nejdále od nejbližších instancí. Instance, které jsou nejblíže maximu-hyperplocha s minimální

vzdáleností k němu – nazývají se podpůrné vektory. Pro každou třídu existuje vždy alespoň jeden takový vektor a často je jich více. Vzhledem k referenčním vektorům pro dvě třídy můžeme snadno vytvořit hyperplochu s maximální rezervou. Intuitivně očekáváme, že hyperplocha s větším odsazením bude přesnější při klasifikaci budoucích datových tříd než hyperplocha s menší rezervou.[14 s. 337]

Termín „odsazení“ (margin) je často spojován se SVM. Rozpětí hyperploch je vzdálenost mezi hyperplochami a 2 nejbližšími datovými body každé třídy. Pointa je v tom, že SVM snaží maximalizovat své odsazení tak, aby se hyperplocha nacházela přibližně ve stejné vzdálenosti od instancí každé třídy – to snižuje možnost chyb klasifikace.

5.7.3 Naive bayes

Neboli naivní Bayes. Analýza založená na Bayesovské klasifikaci, byla aktivně studována a používána od padesátých let v oblasti klasifikace dokumentů, kde se jako znak používala frekvence slov. Algoritmus je škálovatelný podle počtu znaků a přesnost je srovnatelná s jinými populárními metodami, jako SVM.

Při vypočtu implementuje Bayesovu větu a používá úrovně tříd prezentované jako hodnoty znaků nebo vektorů prediktorů pro klasifikaci.

Zjednodušená rovnice pro klasifikaci vypadá takto:

$$P(A|B,C) = \frac{P(B|A) \times P(C|A) \times P(A)}{P(B) \times P(C)}$$

Rovnice najde pravděpodobnost třídy A na základě parametrů B a C. Jinými slovy, pokud vidíme parametry B a C, pak jsou to pravděpodobně data třídy A.

Naivní bayesovský algoritmus je naivní, protože každý parametr klasifikovaných dat je považován bez ohledu na ostatní parametry třídy. I když většinou tomu tak není. Například v datech o pacientovi bude růst ovlivňovat váhu. Nicméně, Naivní bayesovský klasifikátor „se domnívá“, že žádný z těchto příznaků nemá vliv na pravděpodobnost příslušnosti pozorovaného objektu ke třídě, bez ohledu na případné korelace mezi vlastnostmi. Nebo naopak, každý z těchto příznaků je nezávisle ovlivněn.[28]

Nejzřetelnější příklad použití Naivního Bayes byl představen na stackoverflow[29]

- Máme tréninkový soubor dat o 1000 ovoci.
- Ovoce může být banán, pomeranč nebo nějaké jiné (to jsou třídy).
- Ovoce může být dlouhé, sladké nebo žluté (to jsou parametry).

Class	Long	Sweet	Yellow	Total
Banana	400	350	450	500
Orange	0	150	300	300
Other	100	150	50	200
Total	500	650	800	1000

Tabulka 2: Tréninková sada dat Naivní Bayes, převzato z [29]

Z tyto sady vidíme:

Takzvané „apriorní“ pravděpodobnosti. (Kdybychom neznali žádný z atributů ovoce, byl by to náš předpoklad.)

- P (Banana) = 0.5 (500/1000)
  Z 500 banánů 400 jsou dlouhé, 350 jsou sladké a 450 jsou žluté;
- P (Orange) = 0.3 (300/1000)
  Mezi 300 pomeranči není žádný dlouhý, ale ukázalo se, že 150 sladkých a 300 žlutých.
- P (Other) = 0.2 (200/1000)
  Ze zbývajících 200 ovoce bylo 100 dlouhé, 150 sladké a 50 žluté.

Předpokládejme, že neznámé ovoce je dlouhé, sladké a žluté.

Pro výpočet pravděpodobnosti, že neznámé ovoce je banán, se nejprve rozhodneme, zda je toto ovoce podobné banánu. Zde je výpočet pravděpodobnosti třídy „banán“ na základě parametrů „dlouhý“, „sladký“, „žlutý“:

$$\mathrm{P}(\mathrm{Banana}|\mathrm{Long},\mathrm{Sweet},\mathrm{Yellow}) = \frac{P(Long|Banana) \times P(Sweet|Banana) \times P(Yellow|Banana) \times P(Banana)}{P(Long) \times P(Sweet) \times P(Yellow)}$$

Jmenovatel (důkazy) pro všechny následné výpočty bude stejný:

P (Long) = 0.5; P(Sweet) = 0.65; P(Yellow) = 0.8

P (Banana|Long,Sweet,Yellow) = 0.8 * 0.7 * 0.9 * 0.5 / P(důkazy) = 0.252/ P(důkazy)

P (Orange|Long, Sweet, Yellow) = 0/ P(důkazy)

P (Other|Long, Sweet, Yellow) = 0.01875/ P(důkazy)

Vzhledem k tomu, že 0,252 je větší než 0,01875, naivní bayesovský algoritmus klasifikuje toto dlouhé, sladké a žluté ovoce jako banán.

V naivním Bayese se věci pokazí, pokud se určitá hodnota atributu nenachází v učební sadě v kombinaci s každou hodnotou třídy, předpokládejme, že v učebních datech byla hodnota atributu sweet vždy spojena s výsledkem banan. Pravděpodobnost, že sweet dostane „other“: P (sweet | other) bude nulová, a protože i ostatní pravděpodobnosti se násobí, konečná pravděpodobnost „other“ v předchozím příkladu bude rovna nule, bez ohledu na to, jak byly skvělé. Pravděpodobnosti, které jsou nulové, vetují ostatní. [28]

5.7.3.1 Laplaceovo Vyhlazení

Laplaceova vyhlazovací metoda, která zpracovává problém s nulovou pravděpodobností v naivních Bayesech. Pomocí Laplaceova vyhlazování můžeme představit P (sweet|other) jako:

$$P(sweet|other) = \frac{počet\ sweet\ v\ other + \alpha}{other + \alpha * k}$$

Alpha – představuje parametr vyhlazování. Pokud bude vybrána hodnota α! = 0 (ne 0), pravděpodobnost již nebude nulová, i když instance W není přítomna ve výukové sadě dat.

k – představuje počet rozměrů (vlastností) v datech

Řekněme, že výskyt „sweet“ je 3 spojena s výsledkem „other“ v tréninkových datech. Za předpokladu, že máme 2 funkce v našem datovém souboru, tj., k = 2 a celkový počet „other“ ovoce.

Alpha = 1;

$$P(sweet|other) = \frac{3 + 1}{100 + 1 * 2} = \frac{3}{102} = 0{,}029;$$

Jak se zvyšuje pravděpodobnost alpha, pravděpodobnost věrohodnosti se pohybuje směrem k rovnoměrnému rozdělení (0,5). Většinu času se alpha = 1 používá k vyřešení problému nulové pravděpodobnosti.

Nyní je to zcela bayesovská formulace, ve které byly předchozí pravděpodobnosti připisovány všemu, co je v dohledu.

Tento algoritmus je vhodný pro predikci v reálném čase, vícestupňovou prognózu, systém doporučení, klasifikaci textu a analýzu sentimentu. Naivní Bayes algoritmy jsou používány především v analýze sentimentu, filtrování spamu, doporučujících systémech a tak dále. Jsou rychle a snadno implementovatelné, ale jejich největší nevýhoda spočívá v tom, že prediktory musí být nezávislé. Ve většině skutečných případů jsou prediktory závislé, což ztěžuje práci klasifikátoru.

5.8 Apriori

Apriori je originální algoritmus navržený R. Agravalem a R. Srikantem v roce 1994, který vyhledává časté sady prvků pro logická pravidla asociace.[14]

Co jsou metody analýzy pravidel asociace? Používají se k detekci zajímavých vazeb mezi atributy obsaženými v databázi. Nejčastěji pravidla sdružení jsou populární metoda analýzy nákupního košíku, protože je možné prozkoumat všechny možné kombinace potenciálně zajímavých skupin produktů v košíku kupujícího. Pravidla asociace mohou mít jeden nebo více výstupních atributů.[6] Kromě toho, výstupní atribut pro jedno pravidlo může být vstupním atributem pro jiné pravidlo. Z tohoto důvodu je omezený počet atributů schopen generovat stovky asociativních pravidel.

Dalo by se říci, že tento algoritmus je nejpochopitelnější a nejzajímavější z pohledu obyčejného člověka, stejně jako algoritmy rozhodovacích stromů.

Jak vypadá příklad použití algoritmu Apriori? Řekněme, že máme databázi transakcí supermarketu. Můžete si představit databázi jako obrovskou tabulku, kde každý řádek je číslo transakce a každý sloupec představuje jednotlivé nákupy. Algoritmus Apriori nezpracovává číselná data. Proto je nutné před použitím algoritmu převést data do diskrétních kategorií.

Než přejdeme k samotnému algoritmu, musíme určit některé parametry:

1. Zaprvé, musíme nastavit velikost sady. Chceme definovat dvoučlennou, tříčlennou sadu/ kombinace?

2. Za druhé, určení podpory (support) je počet transakcí, které jsou součástí sady, rozdělených do celkového počtu transakcí. Sada, která se rovná podpoře, je nejčastěji viděnou sadou.

$$support(I) = \frac{počet\ transakcí\ obsahujících\ I}{celkový\ počet\ transakcí}$$

3. Za třetí, určit věrohodnost, tedy podmíněnou pravděpodobnost, že určité zboží skončí v koši s jiným zbožím. Příklad: Nejznámější příběh o asociativním pravidle těžby je „pivo a plenka“.

$$confidence(X \rightarrow Y) = \frac{počet\ transakcí\ obsahujících\ X\ a\ Y}{počet\ transakcí\ obsahujících\ X}$$

Jednou z nevýhod opatření je, že může nesprávně reprezentovat význam sdružení. To proto, že vysvětluje jen to, jak je populární zboží X, ale ne zboží Y. Pokud je zboží Y také obecně velmi populární, pak bude větší šance, že dohoda obsahující zboží X bude obsahovat i zboží Y, což povede k nadhodnocení důvěry. Pro vysvětlení základní popularity obou složek, používá se třetí opatření nazývané zvedání (Lift).

4. To říká to, jak je pravděpodobné pořízení prvku Y při nákupu prvku X, ovládající, jak populární prvek Y. Hodnota Lift větší než 1 znamená, že je pravděpodobné, že prvek Y bude zakoupen, je-li zakoupen prvek X, zatímco hodnota nižší než 1 znamená, že je nepravděpodobné, že prvek Y bude koupen, pokud bude zakoupen prvek X. [30]

$$Lift\ (X \rightarrow Y) = \frac{Support(X, Y)}{Support(X),\ \ Support(Y)}$$

Pravidla asociace však obsahují dva typy vztahů, které jsou zajímavé pro marketing:

- Pravidla asociace, která ukazují na růst prodeje konkrétního výrobku, kdy zvýšení prodeje je přímým důsledkem propojení produktu s jedním nebo více jinými produkty v sadě. V tomto případě tato informace může být použita k podpoře propagace produktu.
- Pravidla asociace, která vykazují nižší než očekávaný stupeň důvěry v konkrétní sdružení. V tomto případě je možným závěrem, že produkty uvedené v asociačním pravidle soutěží o stejný trh.

5.9 Oblasti použití Data Mining

Nejčastější oblast použití Data Mining tam, kde je přítomno velké množství dat a zdrojů pro jejich analýzu. Proto je Data Mining nejčastěji zvažován v kontextu:

Finanční analýzy dat

Většina bank a finančních institucí nabízí širokou škálu bankovních, investičních a úvěrových služeb. V tomto sektoru jsou data většinou spolehlivá, mají vysokou kvalitu a jsou relativně kompletní. Prognóza plateb za úvěry a analýza bonity klientů jsou pro podnikání banky zásadní. Banka může učinit svou marketingovou strategii více zaměřenou a efektivní, pokud rozdělí své klienty na podkategorie.

V pojišťovnictví, stejně jako v bankovnictví a marketingu, vzniká úkol zpracovávat množství informací, které mají určit typické skupiny (profily) klientů. Tyto informace se používají k tomu, aby nabízely určité služby pojištění s nejmenším rizikem pro společnost a případně s přínosem pro klienta. Také pomocí technologie Data Mining je řešen takový často se vyskytující problém v pojištění, jako je identifikace případů podvodu (fraud detection).

Marketing a obchod

V oblasti marketingu najde Data Mining velmi široké uplatnění. Základní marketingové otázky „Co se prodává?“, „Jak se prodává?“, „Kdo je spotřebitel?“. Další běžnou sadou metod pro řešení marketingových úkolů jsou metody a algoritmy pro hledání asociativních pravidel. Úspěšně se zde také používá hledání časových zákonitostí.

Analýza spotřebního koše. Určena k identifikaci zboží, o jehož koupi usilují kupující společně. Znalost nákupního košíku je důležitá, když je třeba zlepšit reklamu, vytvořit strategii vytváření zásob zboží, stanovit způsoby jeho rozložení v obchodních halách.

Bioinformatika, lékařství, farmaceutika

V lékařském a biologickém výzkumu, stejně jako v praktické medicíně, je spektrum řešených úkolů tak široké, že je možné použít jakékoliv metodiky těžby dat. Příkladem může být budování diagnostického systému nebo zkoumání účinnosti chirurgického zákroku. Bioinformatika je směr, jehož cílem je vyvinout algoritmy pro analýzu a systematizaci genetických informací. [31] Získané algoritmy se používají k určení struktur makromolekul, stejně jako jejich funkcí, s cílem vysvětlit různé biologické jevy.

Průmyslová výroba

Funkce průmyslové výroby a technologických procesů vytvářejí dobré předpoklady pro možnosti využití technologií Data Mining v rámci řešení různých výrobních úkolů. Technologický proces by měl být ve své podstatě kontrolovaný, a všechny jeho odchylky jsou v předem známých limitech; tj. zde můžeme mluvit o určité stabilitě, která je obvykle neoddělitelnou součástí většiny úkolů, kterým čelí technologie Data Mining.

CRM

Jednou z nejslibnějších oblastí aplikace Data Mining je použití této technologie v analytickém CRM. CRM (Customer Relationship Management) - Správa vztahů se zákazníky. Při sdílení těchto technologií je těžba znalostí kombinována s „těžbou peněz“ z údajů o zákaznících. Důležitým aspektem v práci marketingových a prodejních oddělení je vytvoření holistického pohledu na zákazníky, informace o jejich vlastnostech, struktuře zákaznické základny.

Dvě poslední oblasti použití Data Mining jsou těsně spojeny s další částí této práce. Ale než k tomu přistoupím, krátké shrnutí omezení Data Mining, které řeší Process Mining.

- Používání Data Mining dat k analýze dat a detekci nebo předvídání vzorků má přímou vazbu na obchodní procesy.
- Data Mining analyzuje statické informace. Jinými slovy: data dostupná v době analýzy.
- Data Mining bude hledat skryté vzory v datových sbírkách, ale neposkytne odpověď na konkrétní otázky.
- Data Mining odhaluje určité zákonitosti, ale neposkytuje nadále odpověď na otázku, jak byly tyto zákonitosti nastaveny.
- Data Mining je omezena výhradně na analýzu výsledků. Při inteligentní analýze dat je důležité zaměřit se na základní vzorce v datové sadě. Údaje, které přesahují tyto základní modely, často nejsou zahrnuty do analýz.

6 Process Mining

Termín Process Mining na začátku minulého desetiletí zavedl Wil van der Aalst, profesor na Eindhovenské Technické univerzitě a Queenslandské Technické univerzitě, který je považován za ideologa této technologie. V manifestu, založeném na knize „Process Mining: Discovery, Conformance and Enhancement of Business Processes“ definuje:

„Myšlenkou Process Mining je objevovat, monitorovat a zlepšovat reálné procesy (tj. nepředpokládané procesy) získáváním znalostí z eventu protokolů snadno dostupné v dnešních (informačních) systémech. Zahrnuje automatizovaný proces zjišťování (tj. extrahování proces modelů z protokolu událostí), kontrola shody (tj. sledování odchylek porovnáním modelu a log), sociální sítě/organizační důlní, automatické konstrukce simulačních modelů, rozšíření modelu, opravu modelu, pouzdro predikce a doporučení založená na historii.“[32]

Process Mining, na rozdíl od Data Mining, nemá zájem o nízkoúrovňové zákonitosti v původních datech a nesnaží se přijímat rozhodnutí na jejich základě, ale klade si za úkol **optimalizaci podnikových procesů**, vycházející z referenčních údajů.

Faktem je, že Process Mining sbližuje disciplínu Data Science, zaměřenou na získávání hodnoty z dat, se metodami a technologiemi výzkumu, modelování a optimalizace obchodních procesů.[33]

Obrázek 8: Process Mining jako most mezi data science a process science, převzato z [33 obr. 1.7]

Princip Process Mining je velmi jednoduchý: v případě, že obchodní proces se provádí v jedné cestě nebo jiném informačním systému, pak na základě „stopy“ jeho výkonu je možné obnovit skutečný algoritmus ve formě vizualizace pro pozdější analýzu obchodních procesů. Jako „stopa“ obchodních procesů slouží Event logy (protokol událostí) ze všech systémů používaných ve firmě.

6.1 Protokol událostí

Pod událostmi se rozumí nejširší soubor akcí fixovaných v elektronické podobě, jako je provedení operace v bankomatu, nastavení rentgenového přístroje odborníkem, podání dokladů o placení daní nebo pohovor z HR.

Výchozím bodem pro Process Mining je protokol událostí (Event log). I když to neznamená, že události musí být uloženy ve specializovaných souborech protokolu. Události mohou být uloženy v databázových tabulkách, logech zpráv, e-mailových archivech, logech transakce a dalších zdrojích dat[32], kde každý řádek v takovém protokolu odpovídá samostatné události. Na druhé straně, každá událost nese v sobě informace o případu, který byl proveden v rámci jeho činnosti a doby jeho registrace. [33] Podobné protokoly událostí mohou být považovány za souhrn případů a jednotlivé případy jako sekvence událostí, které se na ně odkazují.

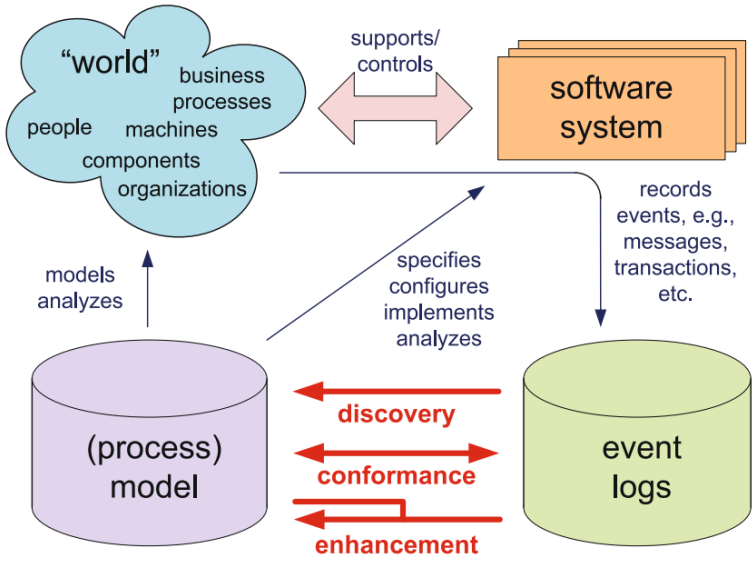
Jako minimální požadavek – Protokol událostí nejčastěji vypadá jako tabulka, a měl by se skládat z identifikátoru zacházení jako unikátní číselný identifikátor (**Case_ID**), Akce (**Activity name**) jako specifikace toho, která akce se konala, a Čas (**Timestamp**) pro přesný čas každé proběhlé akce. Další atributy mohou být přidány, aby poskytly další informace o konkrétních činnostech. V podnikovém prostředí jsou protokoly událostí digitální stopy, které jsou uloženy pro jakoukoli obchodní činnost v databázích, jako jsou ERP/EAM (Enterprise Resource Planning), CRM (Customer Relationship Management), SRM (Supplier Relationship Management), MES (Manufacturing Execution System) nebo nějaký specifický systém. Uchovávají historické údaje o změně stavu objektu, ať už je to aktivum, pracovní úloha nebo odvolání uživatele, každá zákaznická nabídka, objednávka, faktura atd.[34] Pro ukládání protokolů událostí existuje otevřený formát OpenXES.

OpenXES je schéma xml popisující formát textových souborů obsahujících chování informačního systému ve formě toku událostí.

Problém je, že každý ze systémů řeší pouze svůj specializovaný úkol. Jestli budeme chtít vidět úplný obrázek, musíme provést integraci: shromažďovat a mapovat data ze všech systémů. Jen tak se podaří odhalit slabá místa v celkové organizaci práce.

6.2 Klasifikace úloh Process Mining

Protokoly událostí mohou být použity k provedení tří forem Process Mining.



Obrázek 9: Umístění tří hlavních typů Process Mining: objev (discovery), shoda (conformance) a vylepšení (enhancement), převzato z [35 obr. 1.4]

Discovery – Extrakce procesů je nejvýznamnější složkou. Stávající algoritmy umožňují extrahovat modely skutečně fungujících procesů jednoduše zpracováním záznamů událostí. Původní model procesu lze vytvořit ručně.

Conformance – kontrola shody. Zde se porovnává existující model procesu s protokolem událostí stejného procesu. Kontrola shody může být použita k posouzení toho, jak skutečné údaje protokolu odpovídají modelu a naopak.

Enhancement – zlepšení procesu. Stávající model procesu se zlepšuje na základě informací o skutečném procesu zaznamenaném v jakémkoliv protokolu událostí.

Jedna z vážnějších funkcí v Process Mining – grafická vizualizace procesů v databázi, však již nemá schopnost být metodicky přiřazena Process Miningu. Vizualizace databázových procesů dává firmám přehled mimo jiné o velmi náročných obchodních procesech. Firmy nejen vytvářejí důkladné povědomí o procesech, ale také vytvářejí

základnu pro nekonečné zlepšování procesů. Zjištěné procesy mají všechny šance být použity jako nadcházející referenční modely, například pro testování štěstí derivátů optimalizačních potenciálů.

Model procesů je zpravidla prezentován ve formě diagramu BPMN, UML, Petriho síť. Každý z uvedených úkolů pro úspěšné řešení vyžaduje určité pochopení oboru, procesně orientovaný přístup a strukturovaný dost kvalitní protokol. I když se v 1 odstavci Discovery navrhuje nejprve najít vazby a pak z nich vytvořit model BPM, v praxi se to stává velmi zřídka V minulých letech se to dělalo ručně, dlouho se sledovala práce každého zaměstnance a kreslilo složitě pro vnímání vývojových diagramů výrobních procesů. Nyní to není nutné, protože každý informační systém vede podrobný protokol a automaticky nastaví časové značky.

6.3 Petriho síť

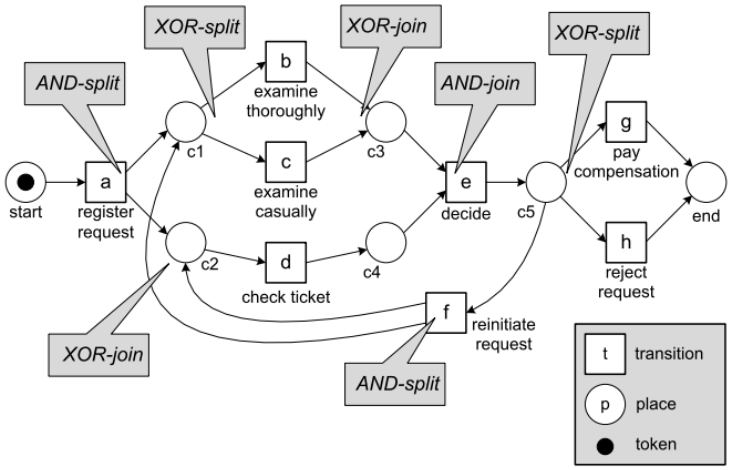
Petriho sítě jsou považovány za skutečný a více naučný jazyk modelování procesů, který umožňuje simulovat paralelismus. Obsahují velmi základní koncepcí modelování toků míst/přechodů. Ale grafická notace je instinktivně jasná a běžná. Petriho sítě jsou považovány za spustitelné, a pro jejich analýzu je možné použít velké množství způsobů analýzy. V současné době jsou Petriho sítě na nízké úrovni, pravděpodobně hlavně proto, že se používají v oblasti počítačových věd a/nebo vývoji softwaru, podle Wikipedie, či k popisu distribuovaných systémů.

Se svou jednoduchou sadou prvků jsou Petriho sítě navrženy tak, aby matematicky formalizovaly algoritmy. Petriho sítě obsahují místa, přechody a hrany.

- Hrany jsou pouze mezi místy a přechody, nikoliv mezi dvěma místy nebo dvěma přechody.
- Místa, ze kterých vedou hrany do přechodu, jsou nazývána vstupní místa tohoto přechodu; místa, do kterých vedou hrany z přechodu, jsou nazývána výstupní místa tohoto přechodu.[36]

Vizuálně Petriho sítě bude vypadat jako na obrázku 10, ale v současné době je důležité poznamenat, že dobře modelovaný proces BPMN je srozumitelný lidem, kteří neznají

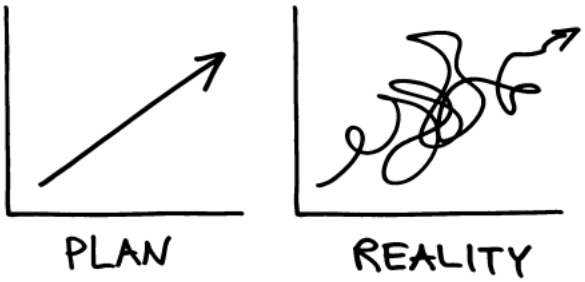
BPMN, protože BPMN 2.0 je notace na vysoké úrovni a je poměrně blízko intuitivnímu pochopení procesu.



Obrázek 10: Značená Petriho sítě, převzato z [35]

6.4 BPM

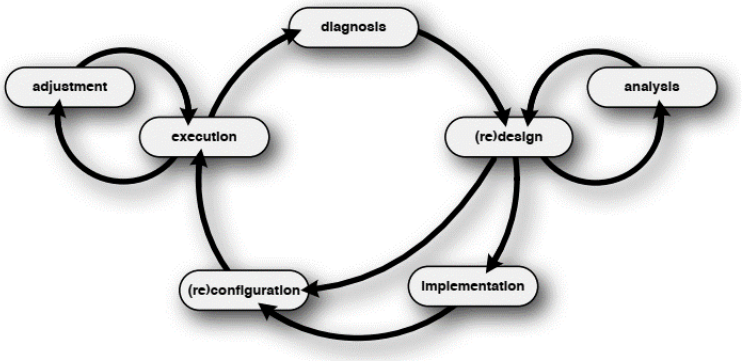
Business process management, životní cyklus řízení procesů. BPM a procesní těžba se zásadně liší: zatímco BPM umožňuje model **To-be** procesy s cílem popsat, jak by měly procesy fungovat, představuje ideální, perfektní procesní tok bez tření, jak je obvykle navrženo teoreticky. Proces Mining vizualizuje **As-Is** procesy a realitu, představuje skutečný procesní tok se všemi odchylkami a složitostmi, ke kterým dochází v provozních procesech v reálném životě.



Obrázek 11: Krátce BPM plán a realita, převzato z [34]

6.4.1 Životní cyklus BPM

Životní cyklus BPM obsahuje sedm fází obchodních procesů a souvisejících informačních systémů.



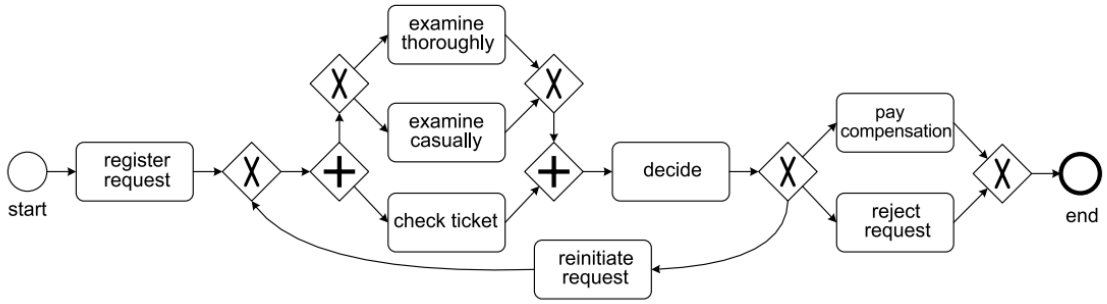
Obrázek 12: Životní cyklus BPM ukazující různá použití procesních modelů, převzato z [32 obr. 4]

Ve fázi **(re)návrhu** (**(re)design**) buď vznikne nový procesní model, nebo se změní stávající. Zde je obchodní proces popsán a zdokumentován. Následuje fáze analýzy (**analysis**) navrhovaného nového modelu a jeho alternativ. Po fázi (re)design následuje fáze implementace (**implementation**) modelu. Pokud je model určen pro systém řízení procesu nebo systému řízení životního cyklu procesu, pak informační systém odpovídající tomuto modelu již může být **použit**. V opačném případě stávající systém (re)nakonfigurován (fáze **(re)configuration**). Ve fázi **provedení** (**execution**) se provádí spuštění vytvořeného modelu. Ve době provádění této fáze je model **monitorován**. Kromě toho mohou být provedeny malé úpravy, které nevyžadují změnu procesu (**adjustment**). Ve fázi **diagnostiky** je běžící proces analyzován a výsledky analýzy mohou iniciovat opětovné spuštění fáze (re)design.[32]

Je zřejmé, že především diagnostická fáze Process Mining je užitečným nástrojem pro realizaci. Další způsoby Process Mining je možné použít ve fázi provedení pro operační doprovod. Monitorování a rady založené na extrahovaných modelech mohou být použity k ovlivnění zpracovaných případů. Podobné formy pomoci při přijímání závěrů lze použít i pro úpravu procesů a pro řízení (re)konfigurace procesů.

6.4.2 BPMN 2.0

Notace modelování obchodních procesů (BPMN) se v poslední době stala jedním z nejpoužívanějších jazyků pro modelování obchodních procesů. BPMN je podporován mnoha dodavateli nástrojů a byl standardizován OMG (Object Management Group).[35]



Obrázek 13: Procesní model pomocí notace BPMN, převzato z [35 obr. 2.7]

Obrázek 13 ukazuje malou podmnožinu všech prvků notace, dohromady je jich zhruba 50. Čtvercem označeným akce se nazývají úkoly. **Akce** mohou být vnořené. Logika směrování se netýká úkolů, ale jednotlivých **eventů**. Existují separační brány a připojení různých typů: AND, XOR, NEBO.

Je třeba poznamenat, že Process Mining nabízí spoustu algoritmů pro kontrolu toku discovery a každý z nich má své vlastní charakteristiky. Cílem není vymýšlet nové algoritmy, ale těžit z těch stávajících a učinit je kompatibilními s BPMN. Objev řídicího toku se tedy opírá o konverzní algoritmy a existující techniky Process Miningu.[37]

6.5 Discovery algoritmy Process Mining

Proces objevu algoritmu je funkce, která mapuje protokol logů na procesní model tak, aby model byl „reprezentativní“ pro chování. Tato definice neurčuje, jaký model procesu by měl být generován, například model BPMN nebo Petriho sítě. Kromě toho mohou být jako vstup použity protokoly událostí s potenciálně mnoha atributy.

Obecně platí, že existuje kompromis mezi následujícími čtyřmi kritérii kvality:

- Fitness – objevený model by měl umožnit chování viděné v protokolu událostí,
- Přesnost – objevený model by neměl umožňovat chování zcela nesouvisející s tím, co bylo vidět v protokolu událostí,
- Zobecnění – objevený model by měl zobecnit příklad chování vidět v protokolu událostí,
- Jednoduchost – objevený model by měl být co nejjednodušší.

6.5.1 Alpha algoritmus

Takže, v protokolech událostí chceme najít model, který může mít cykly, které mohou mít paralelní části, které mohou mít na výběr, a algoritmus Alpha byl prvním algoritmem, který dokázal adekvátně zvládnout paralelismus.

Pomocí protokolu událostí jako vstupních dat α-algoritmus zobrazuje různé „vztahy“ mezi aktivitami, ke kterým dochází v protokolu událostí. Tyto vztahy se používají k vytvoření Petriho sítě, která představuje model procesních toků. Ačkoli α-algoritmus by neměl být považován za metodu těžby, která může být použita v praxi, poskytuje dobrý příklad a sloužil jako základ pro mnoho dalších metod detekce procesů.[38]

Alpha algoritmus používá z logu jako vstupní parametry pouze Aktivity a jejich pořádek. Jiné datové elementy ignoruje: tamestamp, user, id. Tímto způsobem můžeme takový protokol událostí převést na seznam přechodů (transition). A každé trasování je posloupnost názvů akcí.

Prvním krokem najdeme vstupní a výstupní přechody pro Petriho síť. Jakmile to uděláme, budeme mít klíčový krok k objevování nových míst. A najdeme místa tím, že identifikujeme sady přechodů A i B, kde jsou A-vstupní bod pro místo a B-výstupní bod pro místo. Tyto aktivity by měly mít následující vlastnosti: v jedné sadě A nebo B, nikdy by akce neměly následovat samy sebe. Pokud vezmeme nějakou aktivitu v sadě

A i v sadě B, vždy by měla existovat přímá posloupnost mezi těmito dvěma aktivitami. Protokol by tedy měl obsahovat alespoň jednu pozici, ve které za položkou A následuje prvek B, a to by mělo být provedeno pro všechny kombinace. [33] Sada obsahuje maximální možný počet prvků, které lze připojit přes jedno místo. Pro každý takový pár (A, B) spojujeme všechny prvky z A se všemi prvky z B s jedním místem p (A, B). Pak také připojíme příslušné přechody se vstupními a výstupními místy, nakonec připojíme počáteční místo i ke všem přechodům a všem přechodům do koncového stavu.[35]

přímá posloupnost x>y

x>y ⟺ vidíme v Log sub-traces ...xy...

kauzalita x→y

x→y ⟺ x>y ∧ y≯x

pokud existují stopy ...xy...; a žádné stopy ...yx...tento vztah může znamenat, že budeme muset umístit místo mezi **x** a **y**

paralelní x||y

x || y ⟺ x>y ∧ y>x

tedy vidět obojí ...xy... a ...yx...; nelze umístit místo pro takové x a y-pokud bychom umístili, uložili bychom jim nějaký příkaz. jedná se o symetrický vztah (a || B→B || a)

nesouvisející x # y

x # y ⟺x≯y ∧ y≯x; to znamená, že nejsou žádné stopy ...xy... ani ...yx...

to je také symetrický vztah x # y → y # x

Příklad:

L=[abcd,acbd,aed]

	a	b	c	d	e
a	#	→	→	#	→
b	←	#	||	→	→
c	←	||	#	#	→
d	#	←	←	#	←
e	←	#	#	→	#

Tabulka 3: Aktivity a vztahy.

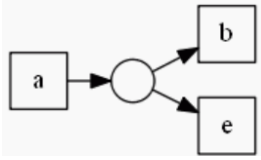
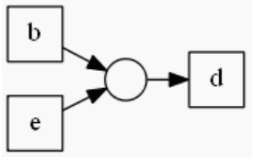
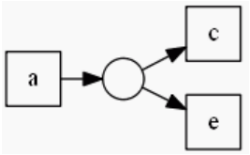
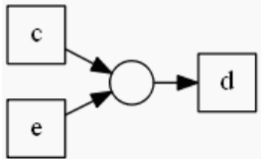
>L = (a,b),(a,c),(a,e),(b,c),(c,b),(b,d),(c,d),(e,d) obsahuje všechny akční dvojice týkající se „přímo sleduje“;

→L= (a,b),(a,c),(a,e),(b,d),(c,d),(e,d) obsahuje všechny páry aktivit ve vztahu „kauzalita“;

#L=a,a),(a,d),(b,b),(b,e),(c,c),(c,e),(d,a),(d,d),(e,b),(e,c),(e,e) někdy C následuje b a někdy naopak

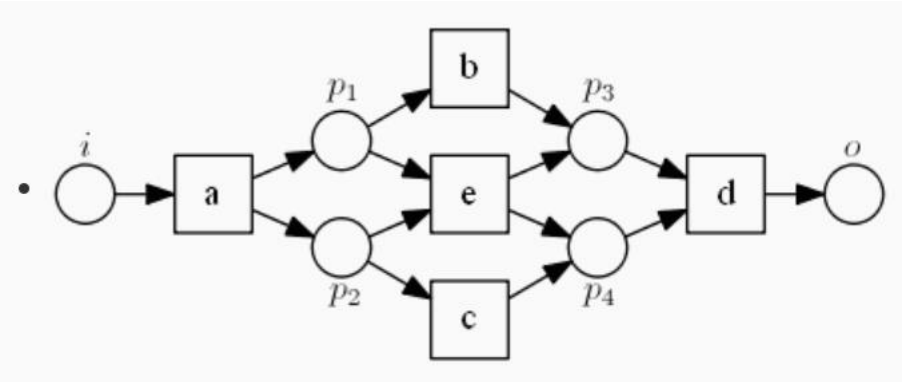
||L = (b,c),(c,b) protože E nikdy přímo nenásleduje B, B nikdy přímo nenásleduje E

Pokud předáme tento event log, Alpha algoritmus automaticky přezkoumá všechna tato data na procesní model. V podstatě tento model umožňuje nekonečné množství stop, ale je založen na konečném počtu příkladů.

	A	B			A	B	
p1	{a}	{b,e}		p3	{b,e}	{d}	
p2	{a}	{c,e}		p4	{c,e}	{d}	

Tabulka 4: Aktivity a vztahy v modelu, převzato z [39]

Model si umí představit chování v podobě Petriho sítě nebo jej můžeme představit jako model BPMN. Chceme jen zachytit chování.



Obrázek 14: Petriho sítě k příkladu L., převzato z [39]

6.5.2 Heuristic miner

Další rodinou algoritmů jsou heuristické algoritmy. Výše uvedené Alpha-algoritmy mohou zpracovávat velké množství speciálních konstrukcí implementaci, ale šumy v datech zpracovávají obvykle nesprávně. K „šumům“ lze přičíst i všechny ojedinělé události v logu (v literatuře je uveden běžný stav, kdy člověk zaplatil za parkování, ale zároveň přijel vlakem). Heuristický algoritmus může pracovat s hlukem v logách, a může být použit k odrazu základní podstaty událostí v logu bez detailů a výjimečných situací. [40] Algoritmus podporuje všechny základní konstrukce, souběžnost, cykly, „neviditelné události“, cykly „krátké cykly“, některé druhy „non-free volby“. Výjimkou jsou „duplicitní“ události.

Heuristický algoritmus se skládá ze tří základních kroků.

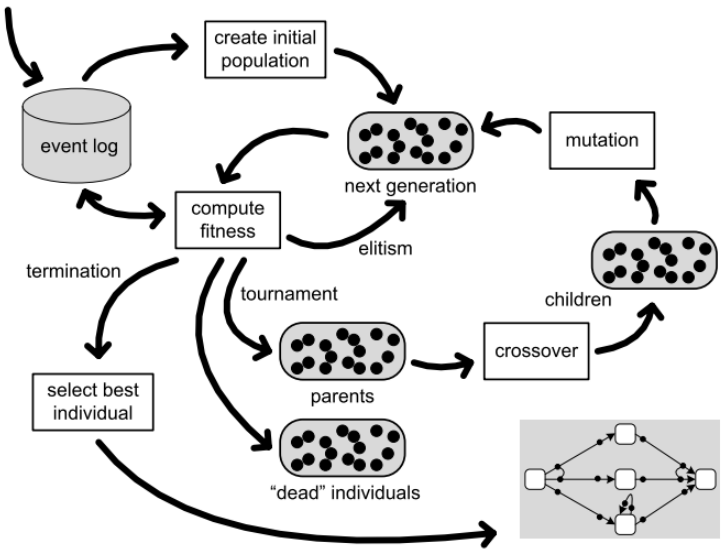
V prvním kroku sestrojíme graf závislosti mezi událostmi v logách (dependency graph), v druhém kroku pro každý vrchol jsou postaveny vstupní a výstupní výrazy, a jsou hledány závislosti na velké vzdálenosti mezi událostmi (vzdálenost podle počtu událostí). Ve třetím kroku se na základě zjištěných informací v předchozích dvou krocích buduje Petriho síť. Algoritmus umožňuje vystavit prahovou hodnotu citlivosti, která vylučuje zpracování vzácných událostí. [40]

6.5.3 Genetic miner

Občas se v oblasti Process Mining objevují potíže s detekcí procesních modelů s určitými konstrukčními systémy a/nebo s přítomností hluku v logách. Mezi hlavní problémové systémy patří: nesvobodná volba (Non-free-choice), neviditelné úkoly (Invisible tasks) a opakující se úkoly (Duplicate tasks).[41] Non-free-choice kombinuje synchronizaci a výběr. Invisible tasks se používají pouze pro účely směrování a nejsou zobrazeny v protokolu. Moderní metody pak z tohoto důvodu mají potíže s detekcí těchto směrovacích úkolů.

K překonání omezení současných mining procesu, slouží o algoritmus z rodiny genetických algoritmů pro business Process Mining. Evoluční přístupy, na rozdíl od algoritmu Alpha, používají iterativní postup k simulaci procesu přirozené evoluce a do značné míry závisí na randomizaci k nalezení nových alternativ. Stejně jako u jakéhokoli genetického algoritmu existují čtyři hlavní kroky: Inicializace, Výběr, Reprodukce a Dokončení [41]

- Je výrazně odolnější vůči hluku,
- Povoluje pomalu usilovat o dokonalost,
- Má schopnost kombinovat s jinými rozloženími.



Obrázek 15: Přehled přístupu používaného k analýze genetických procesů, převzato z [35]

6.5.3.1 Kroky Genetic process miner

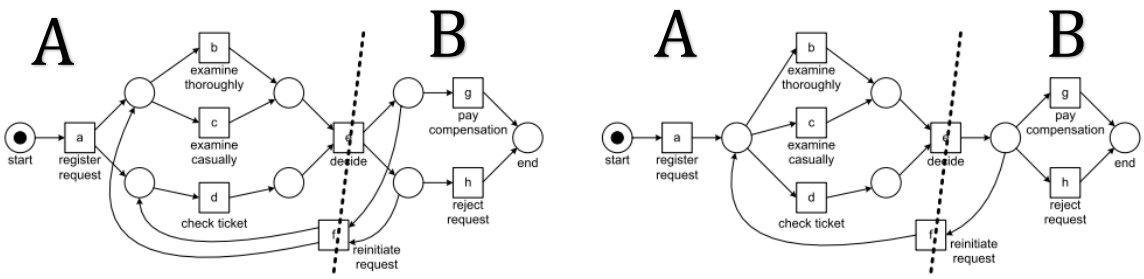
Genetic process miner prochází čtyřmi kroky:

- Nejprve vytvoříme počáteční populaci
    - můžete jednoduše náhodně vytvořit některé Petriho sítě s náhodným přidáním míst,
    - nebo použití Alpha algoritmus.
- V každém kroku vypočítáme vhodnost pro všechny instance populace,
- Elitářství – proces zachování nejlepších,
- Pak jsou všechny „přeživší“ exempláře považovány za „rodiče“
    - **cross-over** proces sdružování různých sítí Petriho,
    - **mutace** přidání některých náhodných změn k diverzifikaci.

6.5.3.2 Cross-Over

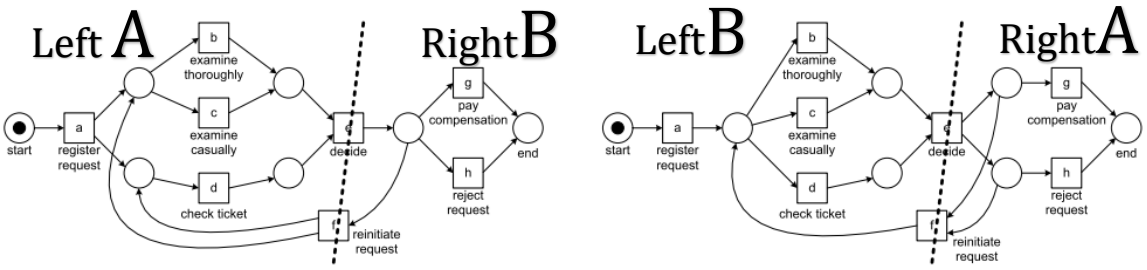
To je proces výroby „dítěte“ možný dvěma rodičovskými instancemi (výběr rodičů), nebo zcela náhodně, či pomoci použití fitness pro rychlejší sblížení. Není vždy třeba vybírat jen „to nejlepší“, potřebujeme rozmanitost.

Jak vytvořit Petriho síť z dalších dvou sítí. Předpokládejme, že máme dvě náhodně vytvořené rodičovské Petriho sítě:



Obrázek 16: Dva mateřské modely, převzato z [35 obr. 6.9]

Minimální řez, který sníží rodiče a dělá z nich děti na obrázku 16 je (e, f), protože v obou případech z **e** jde do **g, h** a vlevo zůstanou **a, b, c, d.**



Obrázek 17: Dva dětské modely, převzato z [35 obr. 6.9]

Vezmeme (e,f) ven a dostaneme dvě odpojené komponenty kombinací levé a pravé části Petriho sítí se vytvoří děti. Tak byly získaný 2 nové modely (Obrázek 17).

Další způsob, jak získat silnější model, je **Mutace.** Způsoby mutace: odstranit nebo přidat místo, přidat oblouk.

6.5.3.3 Funkce Hodnoty: Fitness

V našem případě míra fitness hodnotí kvalitu bodu (samostatný nebo model procesu) ve vyhledávacím prostoru podle protokolu událostí. Genetický algoritmus hledá jedince, jejichž vhodnost je maximální. Takže míra fitness zajišťuje, že lidé s maximální fitness budou moci analyzovat všechny instance procesu (stopy) v logách, v ideálním případě ne více než tyto přechody. [41]

Definice trace-level fitness jako:

$$fitness(\sigma, N) = \frac{1}{2}(1 - \frac{m}{c}) + \frac{1}{2}(1 - \frac{r}{p})$$

m - chybějící tokeny;

c - spotřebované tokeny;

r - tokeny zbývající po dosažení místa výstupu;

p - vyrobené tokeny;

Na příkladu potomka z obrázku 17 počítám fitness:

logtrace: L1=⟨abeg⟩;

M=0, C=0, R=0, P=1 (p = 1, protože na vstupním místě je token);

A - generuje 2 tokena, spotřebuje 1 = p←p+2=3; c←c+1=1

B - generuje + token, spotřebuje 2 = p←p+1=4; c←c+1=2;

E - musí spálit, ale nemůže: jeden token chybí, takže jej přidáme a vypálíme

E = p←p+1=5; c←c+2=4; m←m+1=1

G - p←p+1=6; c←c+1=5

ale je tu jeden zbývající token, který zůstal po vypálení = r←r+1=1

a nakonec odstraníme jeden token z výstupního místa = c←c+1=6

Nahradím data do výše uvedeného vzorce: $fitness(\mathrm{L1}, N) = \frac{1}{2}(1 - \frac{1}{6}) + \frac{1}{2}(1 - \frac{1}{6}) = \frac{5}{6}$

Genetický proces dolování je flexibilní a robustní. Stejně jako heuristické těžební techniky se dokáže vypořádat s hlukem a neúplností. Přístup lze také snadno přizpůsobit a rozšířit. Změnou funkce fitness je možné upřednostnit konkrétní konstrukce. Bohužel, stejně jako většina evolučních přístupů, není těžba genetického procesu příliš účinná pro větší modely a protokoly.

6.5.4 Inductive miner

Na základě moderní literatury, algoritmus detekce je proces známý jako Inductive Miner (IM) v něm není splněna vzdělávací sady dat do přítomného času. IM znamená zlepšení ve srovnání s Alpha a Heuristic miner, zjednodušit studium protokolu událostí, je schopen se vypořádat s nečastým chováním a velkými event logy, zároveň je zajištěna spolehlivost [42]. Zatímco vnitřní reprezentace procesu objeveného Heuristic miner je heuristická síť, vnitřní reprezentace procesu objeveného Inductive miner je strom.

Stejně jako heuristický miner, induktivní miner bere v úvahu frekvence událostí a ignoruje nízkofrekvenční události a izolované cykly událostí. Jako zlepšení jak pro Alpha miner, tak pro Heuristic miner, zaručuje také spolehlivost.

6.6 Použití Process Mining

Otázky, na které Process Mining odpovídá, lze rozdělit do dvou skupin:

- Problémy s výkonem (efektivitou) procesů.
- Otázky konzistence procesů.

Ve skutečnosti to dělá „rentgen“ procesů probíhajících ve společnosti, které poskytují komplexní, a hlavně pravdivý obraz celého řetězce událostí, a nejen jednotlivé kroky, čímž se eliminují bariéry mezi jednotlivými odděleními a odbory. Získané skóre současné provozní výkonnosti společnosti, je objektivní, a proto může být použito pro zjištění možností optimalizace, sledování změn a rychlé reakce. Process Mining vymazáním organizačních hranic umožňuje získat Insights pro jakoukoliv úroveň detailů. Hodnota těchto insitů je, že dávají společnosti základnu a směr pro další kroky ke zlepšení a digitalizaci procesů ve všech divizích.

- Process Mining se zaměřuje na identifikaci, kontrolu a zdokonalování skutečných obchodních procesů. Analýza dat získaných z IT systémů, které podporují naše procesy, dává skutečný, end-to-end pohled na to, jak obchodní procesy fungují.
- Process Mining se dívá na to, jak byly údaje skutečně vytvořeny. Techniky pro dolování procesů také umožňují uživatelům dynamicky vytvářet procesy založené na nejnovějších datech. Procesní těžba může dokonce poskytnout prezentaci obchodních procesů v reálném čase prostřednictvím živého vysílání.
- Techniky pro Process Mining umožňují konkrétně hledat odpovědi na jasné a předem definované otázky.
- Process Mining může poskytnout představu o tom, jak byly výsledky získány. Tato technika nehledá zákonitosti v datech, ale kauzální procesy.
- V Process Miningu mohou být výjimky někdy neméně důležité. Výjimky mohou být časným ukazatelem neefektivity nebo příležitostí ke zlepšení.

7 Příklady algoritmů v praxi

Než se podíváme na aplikace algoritmů v praxi, stojí za to přemýšlet o tom, jak bychom mohli volit mezi algoritmy. To je něco, na co neexistuje jednoznačná odpověď — neexistuje žádný "lepší" algoritmus. Spíše jeden algoritmus může být vhodnější pro daný konkrétní úkol než jiný, nakonec můžeme vyměnit jednu vlastnost algoritmu za jinou. Například, pokud je důležitá přesnost, pak můžeme být připraveni ztratit některé výpočetní účinnosti, nebo investovat více výpočetního výkonu, aby se použít její a naopak. Nebo bychom mohli upřednostňovat jednoduchost před přesností.

Kromě obecných vlastností algoritmů existují některé důležité vlastnosti týkající se modelů odvozených z těchto algoritmů.

- **Bezpečnost** – všechny události přítomné v modelu jsou povoleny a že se tyto události vzájemně vylučují. To znamená, že každé místo může obsahovat pouze jeden token,
- **Správné dokončení** – říká, že když je proces dokončen, nemůže být provedena žádná z jeho událostí,
- **Možnost dokončení** – znamená, že vždy musí existovat možnost dokončit proces.
- **Nedostatek mrtvých částí** – znamená, že pro jakýkoli přechod do modelu existuje vždy jedna nebo více sekvencí událostí, které vedou k jeho vzniku.

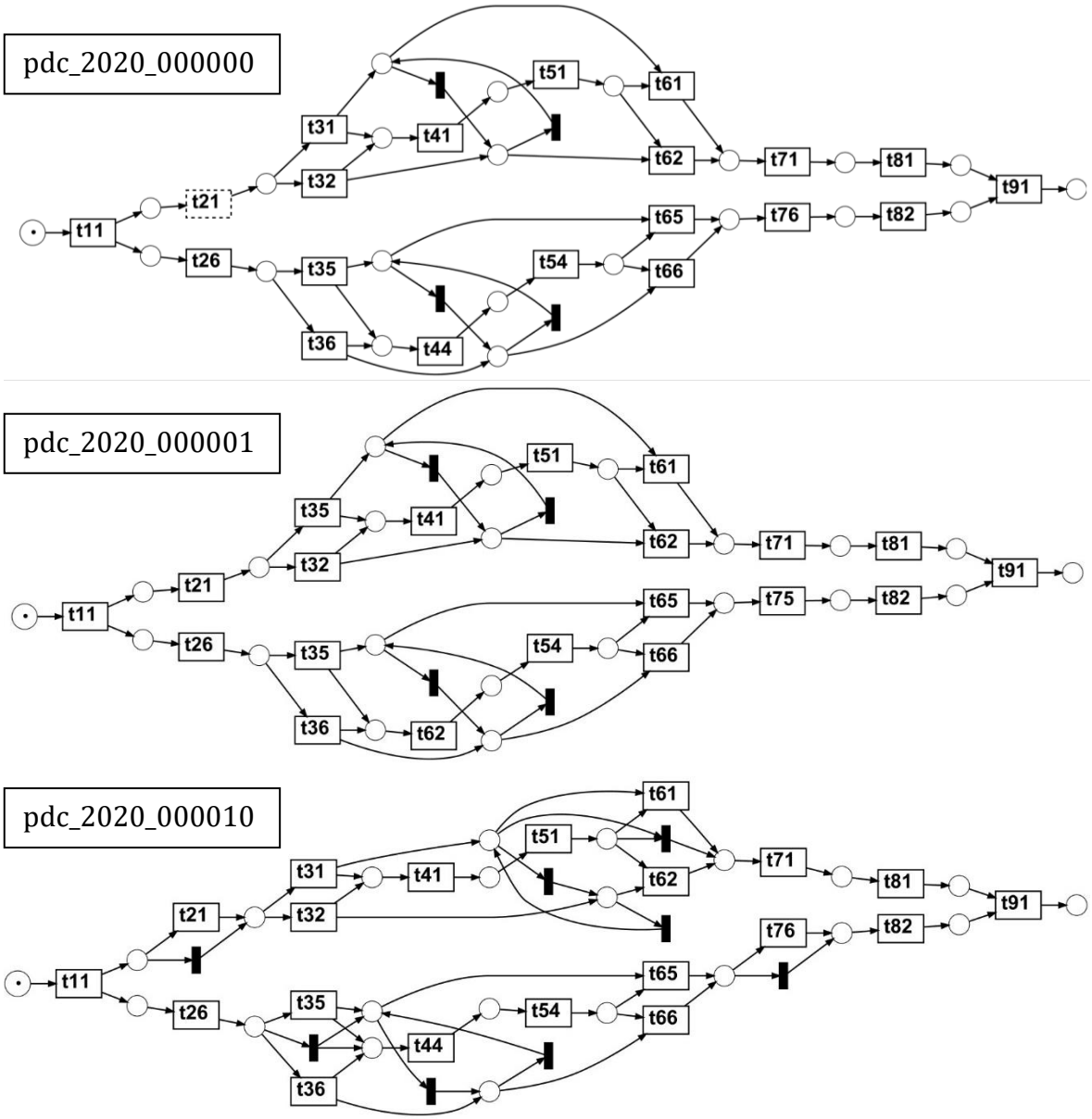
Výše uvedené vlastnosti spojuje **Zdravý rozum**. To znamená, že procesy a modely budou vždy dokončené (bezpečnost, správné ukončení a možnost dokončení) a že alespoň jedna z událostí v procesu bude provedena (chybějící neživých částí). Proto, aby byla síť pracovního postupu spolehlivá, musí mít všechny výše uvedené vlastnosti. Zdravý rozum je jeden z klíčových rozdílů, které odlišují Inductive miner od Alpha a Heuristic miner.

7.1 Process discovery pomoci proM

ProM-to je rozšiřitelný framework, který podporuje širokou škálu metod Process Mining ve formě plug-inů a distribuovány pomocí licence GNU Public License (GPL) s open source.[43] Data pro analýzu jsou převzata z automatizované soutěže Process Mining Conference 2020.[44] Celkem byla v rámci soutěže vygenerována datová sada obsahující 192 traning logů, ale jako příklad fungování algoritmů použity pouze údaje

ze tří prvních protokolů: pdc_2020_0000000, pdc_2020_0000001, pdc_2020_0000010.xes Všechny protokoly jsou prezentovány ve formátu XES.

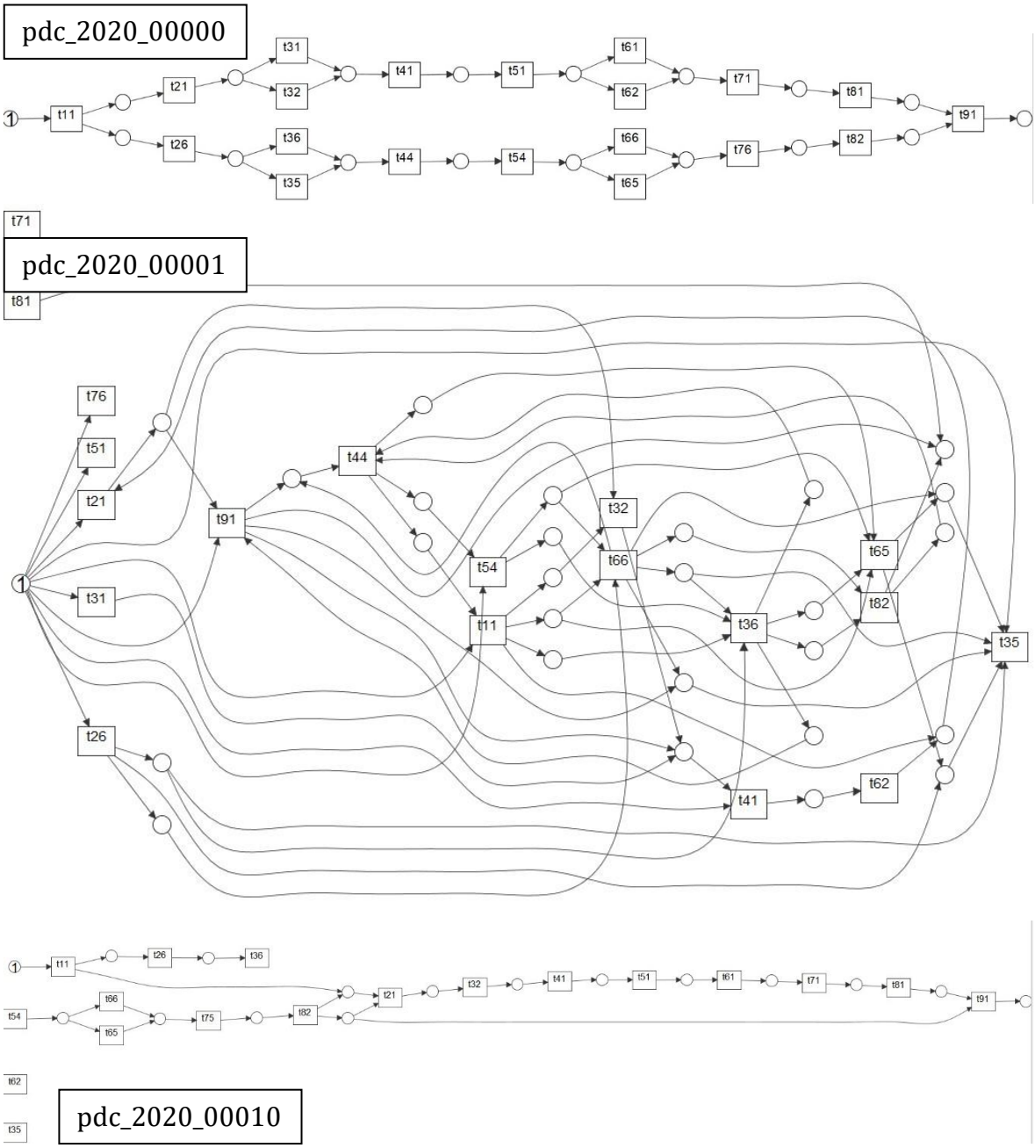
Původní workflow sítě používaný pro generování protokolů jsou:



Obrázek 18: Původní Petrího sít, pomoci [44]

Na obrázku 18 je jasně vidět, že všechny procesy začínají přechodem T11 a končí na T91, ale uvnitř processů logy jsou trochu odlišné. Je vidět poměrně velké množství smyček a paralelních procesů, stejně jako různé sekvence akcií. Sítě jsou dané v ramci souboru pro analýzu.

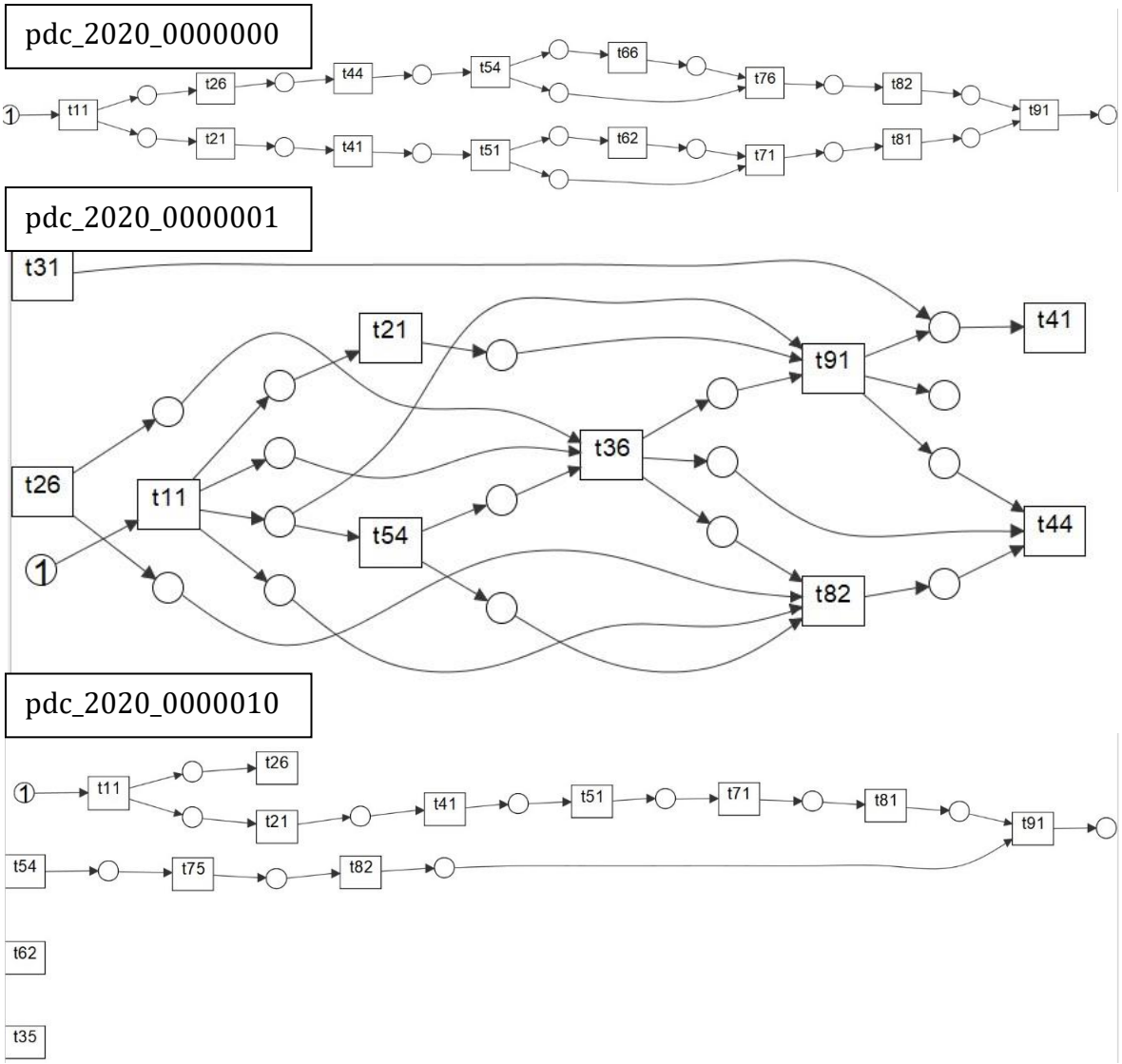
Nyní z testovacích logů pomocí Alpha miner získám model procesu, který skutečně probíhá:



Obrázek 19: Alpha miner, pomoci [44]

Při analýze testovaných logů procesů pomocí Alpha miner je získán zcela jiný model. Logicky se ukázalo, že procesní schéma je prakticky nerozlišitelné pdc_2020_00000, jen se usnadnila a v logech pdc_2020_00001 a pdc_2020_00010 se objevily akce, které nevedou k dokončení. Model pdc_2020_00001 se prakticky nezlepšil, naopak se objevily nové akce a spojení.

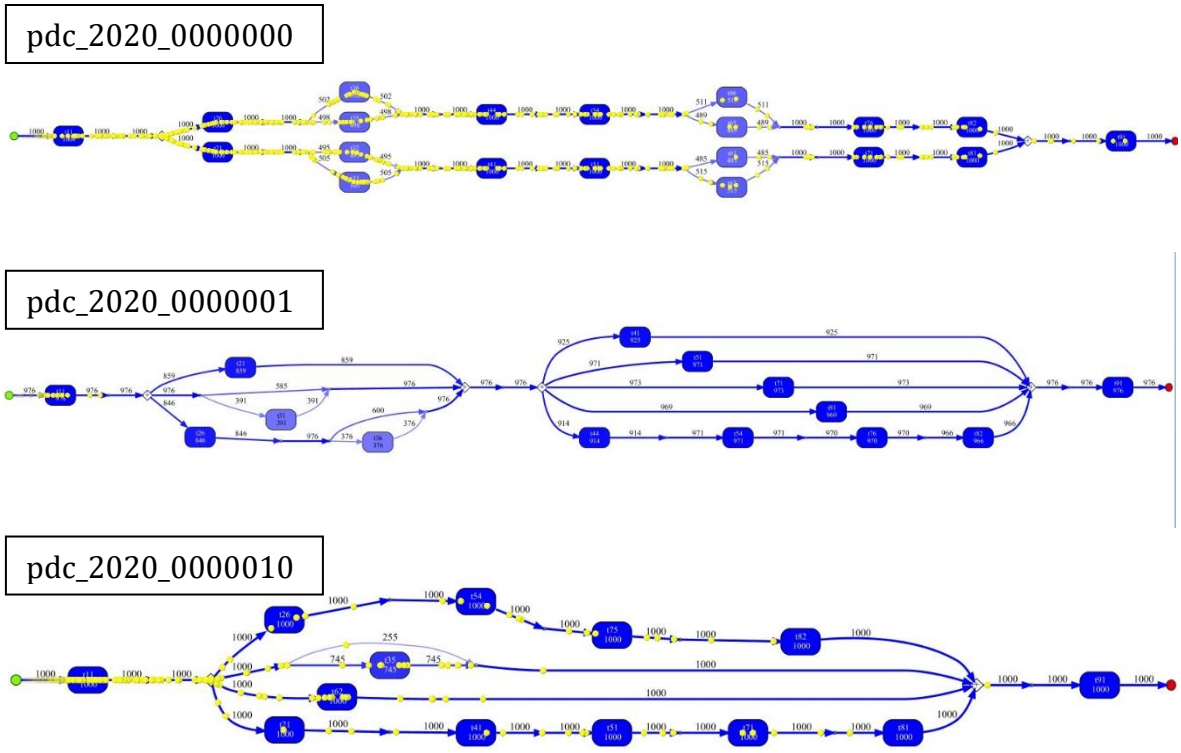
Vnitřní reprezentace procesu objeveného Heuristic miner je heuristická síť. Tuto heuristickou síť můžeme vizualizovat pomocí vizualizatora objektu. Před vizualizací provádím filtr logů: startovní událost bude T11, končí na události T91, procento nejčastějších událostí které přivedou k dokončení procesu je nastaveno na 80%. Petriho sítí zase dostaneme po filtrovaní Heuristic minerem a pomocí Alpha algoritmu:



Obrázek 20: Heuristic miner a Alpha miner, pomoci [44]

Vzhledem k tomu, že se pozornost soustředí na proces jako celek, analýzu by mělo smysl založit pouze na dokončených instancích procesu. Nemá smysl mluvit o nejčastější cestě, pokud není dokončena, nebo spekulovat o době šířky pásma případů, kdy některé z nich stále fungují. Proto předem protokoly vyčisti nebo filtruje Heuristic miner. Ve výsledku vidíme menší počet akcí v průběhu procesu.

ProM umožňuje vizualizaci probíhajících procesů. Standardní algoritmus, který je součástí knihovny proM 6.1 je Inductive vizual miner:



Obrázek 21: Inductive vizual miner, pomoci [44]

Induktivní miner zobrazuje, jak probíhají vzorky akcií (žluté tečky). Čím silnější je proces, tím jasnější jsou modré mosty. Čísla ukazují, jaký počet exemplářů vede mostem.

Jednoduchý příklad použití algoritmů Proces Mining v programu proM 6.1 na příkladu tří procesů, nám ukázal, že ne všechny akce, které byly koncipované na základním modelu, ve skutečnosti běží. Některé z nich nevedou k požadovanému výsledku a dokončení procesu T91. Dokonce i po filtrování existují slepé akce.

Závěr

Tato práce se zabývala popsáním nových oblastí zpracování dat z databáze, jak jsou Data Mining a Process Mining. V procesu průzkumu prací odborníků a odborných článků bylo zjištěno, že navzdory skutečnosti, že mnozí vědci vkládají do termínů nových moderních disciplín Data a Proces Mining poměrně bohaté množství významů, i když Process Mining a Data Mining začínají z dat, techniky Data Mining nejsou zaměřeny na proces a nezaměřují se na data o událostech (instance a akcí obchodních procesů). Pro metody Data Mining mohou řádky (instance) a sloupce (proměnné) znamenat cokoli. Zatímco v procesu klasifikace nebo clusteringu dat se můžou objevit mimořádné výsledky, obecně tyto údaje mohou odkazovat na jakékoliv oblasti a nemusí vždy přinést praktickou hodnotu.

Navzdory Data Mining technologie Process Mining ve svém jádru využívá právě logy pro vizualizaci obchodních procesů ve stavu „as-is“. Díky tomu budou všechny získané výsledky mimořádně spolehlivé a transparentní. Jako výsledek Process Mining bude obchodní proces – posloupnost operací, během nichž se získá smysluplný výsledek pro organizaci.

Velká část práce se věnuje popsání algoritmů s příklady a jejich vnitřnímu zpracování. Na základě informací získaných při studiu různých algoritmů bylo zjištěno, že celkem mining jako metoda zpracování dat může kombinovat modely procesů a metody Data Mining různými způsoby. Nicméně úkoly, které stojí před analýzou procesů vyžadují speciální algoritmy. Kromě toho formát souborů pro Data a Process Mining se úplně liší.

Většina univerzálních algoritmů Data a Process Mining, které prošly časovým testem, však byly popsány v této práci. Pokud algoritmy Data Mining ve většině případů se objevily na základě statistiky, algoritmy Process Mining na základě otevřeného kódu a globálních soutěží jako PDC se objevují až dosud a jsou implementované ve volně distribuované knihovně ProM. Pomocí ní se povedlo na základě training logů z automatizované soutěže Process Mining Conference 2020 ukázat, jak pracují základní algoritmy, kromě Genetic miner. Ten je v beta verzi a nefunguje, jak by měl.

Během práce se objevilo obrovské množství algoritmů a jejich kombinace, ale někdy je jejich výběr pro analýzu dat, která zcela spočívá na zkušenostech a intuici analytického pracovníka. Neexistuje žádné univerzální nebo ready-made řešení problémů.

Výsledkem práce je, že pro kompletní získávaní informací ze stávajících skutečných procesů podniku a pro možnost předvídat nebo najít skryté příležitosti, bude lepší použit obě metody analýzy dat.

Seznam použité literatury

[1] GANTZ, John a David REINSEL. *The digital universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East*. B.m.: IDC IVIEW. prosinec 2012

[2] NEEDHAM, Mass. *Data Creation and Replication Will Grow at a Faster Rate than Installed Storage Capacity, According to the IDC Global DataSphere and StorageSphere Forecasts* [online]. 24. březen 2021. Dostupné z: https://www.idc.com/getdoc.jsp?containerId=prUS47560321&utm_medium=rss_feed&utm_source=alert&utm_campaign=rss_syndication

[2] ACKOFF, Russell L. From Data to Wisdom. *Journal of applied systems analysis*. **1989**(16), 3–9. ISSN 0308-9541

[4] LACKO, Ľuboslav. *Business Intelligence v SQL Serveru 2008: reportovací, analytické a další datové služby*. Brno: Computer Press, 2009. ISBN 978-80-251-2887-9.

[5] KHAN, Rafi Ahmad. KDD for Business Intelligence. *Journal of Computing* [online]. 2012, **2012**(13) [vid. 2021-07-20]. ISSN 2151-9617. Dostupné z: http://www.tlainc.com/articl304.htm

[6] ROIGER, Richard J. *Data mining: a tutorial-based primer*. Second edition. Boca Raton London New York: CRC Press, Taylor & Francis Group, a Chapman & Hall book, 2017. Chapman & Hall/CRC data mining knowledge discovery series. ISBN 978-1-4987-6397-4.

[7] EULER, Christoph. Hypothesis driven thinking in data science | Analytics blog. *Capgemini UK* [online]. 13. leden 2017 [vid. 2021-07-20]. Dostupné z: https://www.capgemini.com/gb-en/2017/01/hypothesis-driven-thinking-in-data-science/

[7] ČERNIAK, Leonid. Proč je Data Scientist sexy než BI analytik. *TAdviser.ru* [online]. 2017 [vid. 2021-07-20]. Dostupné z: https://www.tadviser.ru/index.php/Статья:Почему_Data_Scientist_сексуальнее,_чем_BI-аналитик

[9] FRAWLEY, William J., Gregory PIATETSKY-SHAPIRO a Christopher J. MATHEUS. Knowledge Discovery in Databases: An Overview. *AI Magazine* [online]. 1992, **13**(3), 57. Dostupné z: doi:10.1609/aimag.v13i3.1011

[10] FAYYAD, Usama, Gregory PIATETSKY-SHAPIRO a Padhraic SMYTH. Knowledge Discovery and Data Mining: Towards a Unifying Framework. *AAAI Press*. 1996, **1996**, KDD'96, 7.

[11] GUERRA-HERNANDEZ, Alejandro a Rosibelda MONDRAGON-BECERRA. *Explorations of the BDI Multi-Agent support for the Knowledge Discovery in Databases Process*. B.m.: Departamento de Inteligencia Artificial Universidad Veracruzana. srpen 2008

[12] CRISP-DM. *Data Science Process Alliance* [online]. [vid. 2021-07-28]. Dostupné z: https://www.datascience-pm.com/crisp-dm-2/

[13] AGRAWAL, R., T. IMIELINSKI a A. SWAMI. Database mining: a performance perspective. *IEEE Transactions on Knowledge and Data Engineering* [online]. 1993, **5**(6), 914–925. ISSN 1041-4347. Dostupné z: doi:10.1109/69.250074

[14] HAN, Jiawei, Jian PEI, Micheline KAMBER a an O'Reilly Media Company SAFARI. *Data Mining: Concepts and Techniques, 3rd Edition* [online]. 2011 [vid. 2021-07-23]. Dostupné
z: https://www.safaribooksonline.com/library/view//9780123814791/?ar

[15] AMIGO, José Manuel. Data Mining, Machine Learning, Deep Learning, Chemometrics. Definitions, common points and Trends (Spoiler Alert: VALIDATE your models!). *Brazilian Journal of Analytical Chemistry* [online]. 2021, **8**(32), 22–38. ISSN 21793425, 21793433. Dostupné z: doi:10.30744/brjac.2179-3425.AR-38-2021

[16] NETTLETON, David. *Commercial data mining processing, analysis and modeling for predictive analytics projects* [online]. Amsterdam: Elsevier, 2014 [vid. 2021-07-25]. ISBN 978-1-306-44792-8. Dostupné z: http://sfx.urv.cat/urv?url_ver=Z39.88-2004&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx&ctx_enc=info:ofi/enc:UTF-8&ctx_ver=Z39.88-2004&rfr_id=info:sid/sfxit.com:azbook&sfx.ignore_date_threshold=1&rft.object_id=2550000001242657

[17] GIUDICI, Paolo. *Applied data mining: statistical methods for business and industry*. Reprinted. Chichester: Wiley, 2005. ISBN 978-0-470-84679-7.

[18] NOVÁK, Tomáš a Miloslav KOPEČEK. Typy proměnných. *Psychiatrie pro praxi*. nedatováno, **2010**(11(4)), 176–177.

[17] DUNCAN, Owen a John PARENTE. *Data Mining Algorithms (Analysis Services - Data Mining* [online]. 5. leden 2018 [vid. 2021-07-17]. Dostupné z: https://docs.microsoft.com/en-us/analysis-services/data-mining/data-mining-algorithms-analysis-services-data-mining

[18] LEVKOVIČ-MASLJUK, Leonid. Velké výkopy a velké výzvy. *11 (679)*. 2007, 48–51. Dostupno z: http://old.computerra.ru/2007/679/311831/

[19] GHOSHAL, Alokananda. Decision Tree in Data Mining | Application | Importance | Advantages. *EDUCBA* [online]. 18. prosinec 2019 [vid. 2021-07-28]. Dostupné z: https://www.educba.com/decision-tree-in-data-mining/

[22] SALZBERG, Steven L. C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning* [online]. 1994, **16**(3), 235–240. ISSN 1573-0565. Dostupné z: doi:10.1007/BF00993309

[23] WANG, John, ed. *Encyclopedia of data warehousing and mining*. 2. ed. Hershey, Pa.: Information Science Reference, 2009. ISBN 978-1-60566-011-0.

[24] ROMERO, Cristóbal, Sebastian VENTURA, Mykola PECHENIZKIY a Ryan Shaun Joazeiro de BAKER, ed. *Handbook of educational data mining*. Boca Raton, Fla.: CRC Press, 2011. Chapman & Hall/CRC data mining and knowledge discovery series. ISBN 978-1-4398-0457-5.

[25] YE, Nong. *Data mining: theories, algorithms, and examples*. Boca Raton: CRC Press, 2014. ISBN 978-1-4398-0839-9.

[26] COX, Earl. *Fuzzy modeling and genetic algorithms for data mining and exploration* [online]. San Francisco, CA: Elsevier/Morgan Kaufmann, 2005 [vid. 2021-08-03]. ISBN 9780080470597. Dostupné z: http://site.ebrary.com/id/10167072

[27] KAO, Ming-Yang. *Encyclopedia of algorithms: with 183 figures and 38 tables*. New York, NY: Springer, 2008. Springer reference. ISBN 978-0-387-30162-4.

[28] WITTEN, Ian H., Eibe FRANK a Mark A. HALL. *Data mining: practical machine learning tools and techniques*. 3. ed. Amsterdam Heidelberg: Morgan Kaufmann, Elsevier, 2011. ISBN 978-0-12-374856-0.

[29] Yavar. A simple explanation of Naive Bayes Classification [closed]. In: *Stack Overflow* [online]. 2019 [vid. 2021-08-04]. Dostupné z: https://stackoverflow.com/questions/10059594/a-simple-explanation-of-naive-bayes-classification

[30] NG, Annalyn. Association Rules and the Apriori Algorithm: A Tutorial. *KDnuggets* [online]. 2016 [vid. 2021-08-04]. Dostupné z: https://www.kdnuggets.com/association-rules-and-the-apriori-algorithm-a-tutorial.html/

[31] RAZA, Khalid. APPLICATION OF DATA MINING IN BIOINFORMATICS. 2021, **1**(2), 5.

[32] VAN DER AALST, Wil a Wei TAN. Process Mining Manifesto. In: Florian DANIEL, ed. *Business Process Management Workshops* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012 [vid. 2021-08-05], Lecture Notes in Business Information Processing, s. 169–194. ISBN 978-3-642-28107-5. Dostupné z: doi:10.1007/978-3-642-28108-2_19

[33] AALST, Wil van der. *Process mining: data science in action* [online]. Second edition. Berlin Heidelberg New York Dordrecht London: Springer, 2016. ISBN 978-3-662-49850-7. Dostupné z: doi:10.1007/978-3-662-49851-4

[34] REINKEMEYER, Lars, ed. *Process mining in action: principles, use cases and outlook*. Cham: Springer, 2020. ISBN 978-3-030-40171-9.

[35] AALST, Wil van der. *Process mining: discovery, conformance and enhancement of business processes* [online]. Berlin Heidelberg: Springer, 2011. ISBN 978-3-642-19344-6. Dostupné z: doi:10.1007/978-3-642-19345-3

[36] *Petriho síť* [online]. 2020 [vid. 2021-08-07]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=Petriho_s%C3%AD%C5%A5&oldid=18810452

[37] KALENKOVA, Anna A., Wil M. P. VAN DER AALST, Irina A. LOMAZOVA a Vladimir A. RUBIN. Process mining using BPMN: relating event logs and process models. *Software & Systems Modeling* [online]. 2017, **16**(4), 1019–1048. ISSN 1619-1366, 1619-1374. Dostupné z: doi:10.1007/s10270-015-0502-0

[38] AALST, Wil, A. WEIJTERS a Laura MĂRUŞTER. Workflow Mining: Discovering Process Models from Event Logs. *Knowledge and Data Engineering, IEEE Transactions on* [online]. 2004, **16**, 1128–1142. Dostupné z: doi:10.1109/TKDE.2004.47

[39] CHANG, Hung. *The Alpha Plus Algorithm for Process Mining* [online]. B.m., 2014 [vid. 2021-08-14]. Université Libre de Bruxelles. Dostupné z: https://docs.google.com/document/d/1JtuECbGZ3DusNpmBZhXeq8R_UPCRU5V7NG8GL17h1aA/pub

[40] WEIJTERS, A., Wil AALST a Alves MEDEIROS. *Process Mining with the Heuristics Miner-algorithm*. 2006.

[41] DE MEDEIROS, A. K. A., A. J. M. M. WEIJTERS a W. M. P. VAN DER AALST. Genetic process mining: an experimental evaluation. *Data Mining and Knowledge Discovery* [online]. 2007, **14**(2), 245–304. ISSN 1384-5810, 1573-756X. Dostupné z: doi:10.1007/s10618-006-0061-7

[42] LEEMANS, Sander, Dirk FAHLAND a Wil AALST. *Process and Deviation Exploration with Inductive Visual Miner*. 2014.

[43] *Tools start | ProM* [online]. [vid. 2021-08-15]. Dostupné z: https://www.promtools.org/doku.php

[44] VERBEEK, Eric. *Process Discovery Contest 2020* [online]. IEEE XES ISO PNML. B.m.: 4TU.ResearchData. 21. květen 2021 [vid. 2021-08-12]. Dostupné z: doi:10.4121/14626020.V1

Seznam obrázků

Seznam tabulek

Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání bakalářské práce

Autor:	**Varvara Chikina**
Studium:	I1800415
Studijní program:	B6209 Systémové inženýrství a informatika
Studijní obor:	Informační management
Název bakalářské práce:	**Data x Process Mining**
Název bakalářské práce AJ:	Data x Process Mining

Cíl, metody, literatura, předpoklady:

Porovnání Data a process Minigových algoritmů.

1. Der Aalst, V., {\&
   Mining, W. P. (2011). Discovery, Conformance and Enhancement of Business Processes.
2. Han, J., Pei, J., {\&
   Kamber, M. (2011). *Data mining: concepts and techniques. Elsevier.*

Garantující pracoviště:	Katedra informačních technologií, Fakulta informatiky a managementu
Vedoucí práce:	doc. Ing. Hana Tomášková, Ph.D.
Oponent:	Ing. Karel Mls, Ph.D.
Datum zadání závěrečné práce:	21.10.2019