



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská



Těžba dat z experimentů na tokamaku COMPASS

Data mining on the COMPASS tokamak experiments

Bakalářská práce

Autor:	Matěj Zorek
Vedoucí práce:	Ing. Vít Škvára
Konzultant:	Ing. Jakub Urban, PhD
Akademický rok:	2017/2018

- Zadání práce -

- Zadání práce (zadní strana) -

Poděkování:

Chtěl bych zde poděkovat především svému školiteli Ing. Vítovi Škvárovi za pečlivost, ochotu, vstřícnost a odborné i lidské zázemí při vedení mé bakalářské práce. Dále děkuji svému konzultantovi Ing. Jakubu Urbanovi, PhD., za pomoc nejen v začátcích řešení problému, jímž se tato práce zabývá. V neposlední řadě bych chtěl poděkovat Ing. Ondřeji Groverovi za pomoc s fyzikální složkou věci.

Čestné prohlášení:

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 2. července 2018

Matěj Zorek

Těžba dat z experimentů na tokamaku COMPASS

Obor: Matematické inženýrství

Druh práce: Bakalářská práce

Konzultant: Ing. Jakub Urban, PhD., Ústav fyziky plazmatu, AV ČR, Za Slovankou 1782/3, 182 00 Praha 8

Klíčová slova: klíčová slova (nebo výrazy) seřazená podle abecedy a oddělená čárkou

Data mining on the COMPASS tokamak experiments

[illegible]

Key words: keywords in alphabetical order separated by commas

Obsah

Úvod	11
1 Úvodní terminologie	13
1.1 Plazma	13
1.2 Tokamak COMPASS	13
1.3 Záření H_α (H-alpha)	14
1.4 Podrobnější popis problematiky	14
1.5 Strojové učení	15
2 Teoretická část	17
2.1 Diskrétní Markovův proces	17
2.2 Skrytý Markovův model	18
2.2.1 Viterbiho algoritmus	20
2.3 K-means Clustering	22
3 Praktická část - experimenty	25
3.1 Příznaky (Features)	25
3.1.1 První a druhá derivace	26
3.1.2 Savitzky-Golay filtr	26
3.1.3 Klouzavý průměr	29
3.1.4 Exponenciální klouzavý průměr	30
3.1.5 Klouzavý rozptyl	30
3.2 Skrytý Markovův model vs K-means na syntetických datech	31
3.3 Skrytý Markovův model vs K-means na reálných datech detekce H-modu	31
3.4 Skrytý Markovův model vs K-means na reálných datech obecné datasety	31
4 Výsledky	33
4.1 Způsoby vyhodnocení výsledků	33
4.1.1 Confusion Matrix (matice záměn)	33
4.1.2 Přesnost	34
4.1.3 Precision	34
4.1.4 Recall	34
4.1.5 F míra	35
4.2 Skrytý Markovův model vs K-means na syntetických datech výsledky	35
4.3 Skrytý Markovův model vs K-means na reálných datech detekce H-modu - výsledky	35
4.4 Skrytý Markovův model vs K-means na reálných datech obecné datasety - výsledky	35
Závěr	37

Úvod

V dnešní moderní společnosti funguje téměř vše na elektřinu a nároky stále rostou. Řešením by mohlo být ovládnutí termojaderné reakce. Abychom tento potenciál mohly naplno využít, potřebujeme tuto reakci pořádně pochopit. Jedním z nepokročilejších zařízení sloužících k jejímu výzkumu jsou tokamaky. Znalost okamžitého stavu plazmatu uvnitř tokamaku je tedy poměrně důležitá. Právě tímto problémem se bude tato práce zabývat.

Cílem práce je nalézt způsob, jak určit tyto stavy v reálném čase za použití strojového učení. Nejdříve je potřeba seznámit se se základní fyzikální podstatou jevů, vyvstávajících při výbojích na tokamaku. Následně je třeba prostudovat metody strojového učení schopné řešit tento problém. Poté budou tyto algoritmy natrénovány na skutečných datech z tokamaku COMPASS a aplikovány na testovací vzorek dat. Posledním úkolem je vyhodnotit výsledky a porovnat metody mezi sebou.

V první kapitole je stručně zodpovězeno, co je to plazma. Následuje popis základní problematiky tokamaků. Dále je zde podrobněji představen problém, který tato práce řeší. Na konci kapitoly je vysvětleno strojové učení.

V druhé kapitole jsou uvedeny dvě metody strojového učení. Nejprve je podrobně popsán Skrytý Markovův model. Poté je vysvětlen Viterbiho algoritmus sloužící právě při výpočtu dříve jmenovaného modelu. Na konci této kapitoly představena shlukovací metoda K-means.

Ve třetí a poslední kapitole jsou postupně popsány příznaky používané při aplikaci obou metod. Následuje seznam metrik, sloužících k vyhodnocení úspěšnosti výsledků. Dále je zde uveden postup práce a řešení dílčích problémů. Tato kapitola je zakončena porovnáním výsledků použitých metod.

Kapitola 1

Úvodní terminologie

V této kapitole si nejdříve zodpovíme, co je to plazma. Dále se seznámíme se základní problematikou tokamaků, zejména tokamaku COMPASS. Poté si podrobněji představíme problém, kterým se bude tato práce zabývat. Nakonec bude uvedeno strojové učení.

1.1 Plazma

Plazma je jedním ze čtyř základních skupenství. Jedná se o ionizovaný plyn, jehož atomy jsou rozděleny na pozitivní ionty a na negativní elektrony. Na rozdíl od zbylých tří skupenství se volně, za normálních podmínek, na zemském povrchu nevyskytuje. Paradoxně však 99% veškeré hmoty ve vesmíru tvoří právě plazma.

V laboratoři se získává typicky zahříváním a ionizováním malého množství plynu pomocí elektrického proudu nebo radiových vln. Obvykle tyto prostředky dodávají energii přímo volným elektronům uvnitř. Poté se během srážek těchto nabytých elektronů s atomy uvolňují další elektrony a tento kaskádový proces postupuje, dokud plyn nedosáhne požadovaného stupně ionizace.

Rozdíl mezi velmi slabě ionizovaným plynem a plazmatem je do značné míry záležitostí terminologie a způsobu interpretace. V závislosti na okolní teplotě prostředí a hustotě dělíme plazmata na částečně ionizované a plně ionizované. Příkladem částečně ionizovaného plazmatu je třeba blesk nebo neonové světlo. Naproti tomu plně ionizované plazma lze nalézt uvnitř slunce nebo právě v tokamaku.

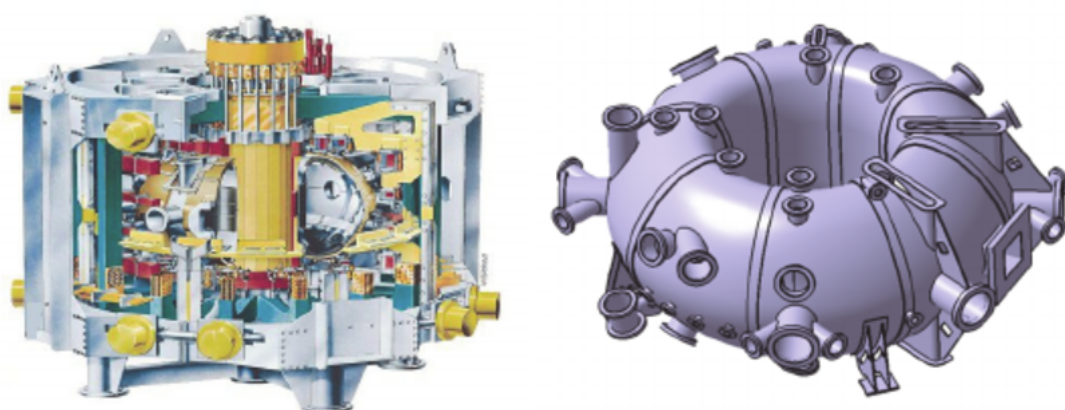
1.2 Tokamak COMPASS

Název tokamak je zkratkou původního ruského názvu toroidálnaja kamera s magnitnymi katuškami (toroidní komora s magnetickými cívkami). V podstatě se jedná o experimentální zařízení využívané k tvorbě vysokoteplotního plazmatu a jeho následné kontrole pomocí magnetického pole. V současnosti jsou tokamaky považovány za jednu z nejnadějnějších cest k dosažení kontrolované jaderné fúze. Pokud by se podařilo fúzi efektivně a trvale udržet, mohlo by se pak přebytečné teplo převést, po vzoru tepelných elektráren, na elektřinu. Tímto způsobem bychom získali téměř nevyčerpatelný zdroj energie, který by byl současně šetrný k životnímu prostředí.

Na rozdíl od jaderných reaktorů, kde probíhá štěpení těžkých jader uranu ^{235}U na lehčí jádra, v tokamacích dochází ke slučování lehkých jader za účelem vzniku těžších jader. Při této termojaderné reakci se nejčastěji slučují jádra deuteria a tricia. Výsledkem reakce je hélium a nositel energie neutron. Problém však tkví v tom, že pro udržení termojaderné fúze je zapotřebí velmi vysokých teplot a dostatečné doby trvání. Abychom toho docílili, musíme držet částice uprostřed toroidní komory, protože při vychý-

lení nebo kontaktu se stěnou dochází k velice rychlému ochlazování. Proto se využívá silné magnetické pole, produkované cívkami, k manipulaci s nabitými částicemi plazmatu.

Tokamak COMPASS je umístěn v Praze Ládví na Ústavu fyziky plazmatu Akademie věd České republiky již od roku 2004 [10]. Původně byl umístěn a provozován ve Velké Británii pod UKAEA (UK Atomic Energy Authority) do roku 2002, kdy byl nahrazen tokamakem MAST. Díky svým rozměrům je řazen do kategorie menších tokamaků. I přes svou malou velikost umožňuje dosáhnout vysokoudržitelného stavu plazmatu neboli H-módu (High-confinement mode) a zároveň odpovídá desetině velikosti tokamaku ITER, v současnosti budovanému ve Francii. Právě díky těmto dvěma vlastnostem je nyní využíván ke studiu specifických jevů, které jsou třeba k pochopení plazmatu a jeho následného udržení v rovnováze.



Obr. 1.1: Na levém obrázku je vyobrazen průřez tokamakem COMPASS a na pravém obrázku je jeho toroidní vakuová komora. [16]

1.3 Záření H_α (H-alpha)

Záření H_α je specifická červená spektrální čára vyzařovaná vodíkem s vlnovou délkou 656,28 nm. Na COMPASSu se toto záření měří pomocí spektrometru HR 2000+ [12]. Spektrální čáry H_α udávají počet částic, které se neutralizovaly během interakcí plazmatu se stěnou (převzato z [13]).

1.4 Podrobnější popis problematiky

V předchozí podkapitole bylo zmíněno, že tokamak COMPASS slouží v současnosti ke studiu chování plazmatu a jeho udržení. Plazma se může uvnitř tokamaku nacházet ve čtyřech základních stavech. Prvním z nich je nízkoudržitelný L-mód (Low-confinement mode). V tomto stavu se plazma nachází bezprostředně po zažehnutí nebo případně po skončení H-módu. Pokud se plazma nachází v L-módu, je velice náročné udržet termojadernou reakci na delší dobu a téměř nemožné udržet jí dlouhodoběji.

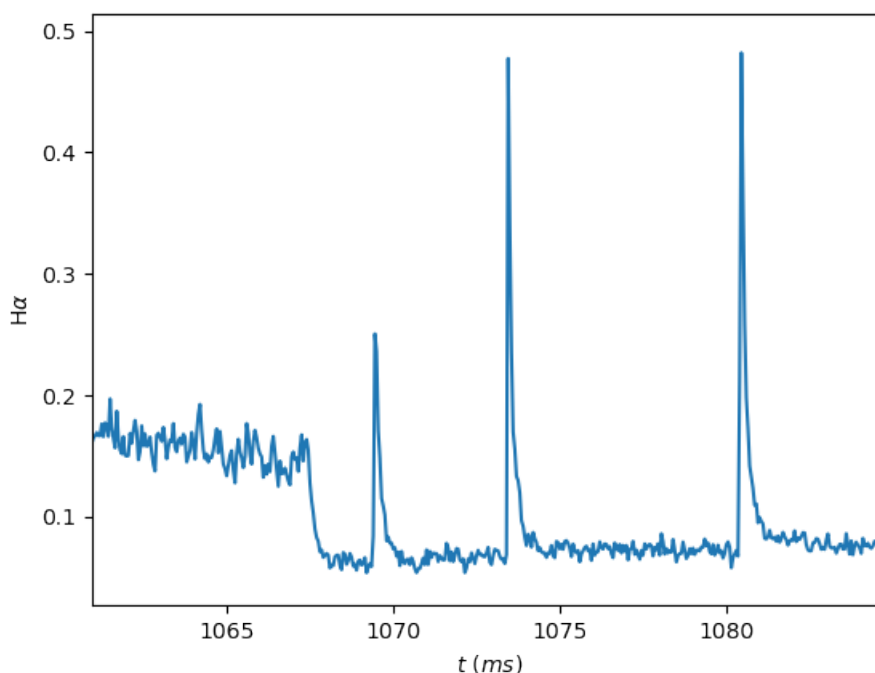
Druhým stavem je již dříve zmíněný H-mód, který byl objeven německým vědcem Fritzem Wagne-rem v roce 1982 [11]. V tomto stavu je možné lépe kontrolovat chování plazmatu a především jí držet v rovnováze po delší dobu. Tento stav je také standartním referenčním režimem budoucího tokamaku ITER.

Třetím v řadě je ELM (edge-localized mode). ELMy jsou v podstatě narušující nestability, k nimž dochází na okraji plazmy. ELMy se navíc mohou vyskytovat pouze během H-módu.

Posledním stavem je disrupce. Disrupce představuje fyzikální jev, během něhož dochází k přetržení nebo náhlé zráte udzení plazmatu.

Aby bylo možné správně porozumět těmto jevům je zapotřebí velkého množství informací. Příkladem může být teplota během jednotlivých stavů atd. Bohužel většina detektorů a měřicích přístrojů je limitovaná svou snímkovací frekvencí. Proto by bylo třeba vědět, v jakém stavu se zrovna plazma nachází, aby bylo možno měřit ve správnou chvíli.

V této práci se budeme snažit najít způsob/y, jak klasifikovat první tři výše zmíněné stavy plazmatu v reálném čase. Využívat k tomu budeme data naměřených hodnot záření H_α a metody strojového učení. Náhled ideálně vypadajícího signálu je možné vidět na Obr. 1.2. Na tomto obrázku je možné přesně sledovat všechny tři stavy. Signál začíná v L-módu. Asi po 10 milisekundách přechází do H-módu. Jelikož H-mód je lépe udržitelný, nevyzažuje plazma tolik vodíku jako v L-módu. Jednotlivé peaky jsou pak ELMy. Ve vakuu, jako je uvnitř tokamaku, se všechny částice rozprostřou po stěnách komory. Když se pak vlivem nějakých nestabilit utrhne kus plazmatu a dotkne se stěny, částice, jež se na stěně nacházejí, se rozzáří a my vidíme ELM. V praxi však signály vypadají spíše jako na Obr. 1.3, kde už některé stavy nejsou jednoznačně viditelné.



Obr. 1.2: Ukázka ideálně vypadajícího záření H_α zaznamenaného detektorem. (číslo výstřelu: 15364)

1.5 Strojové učení

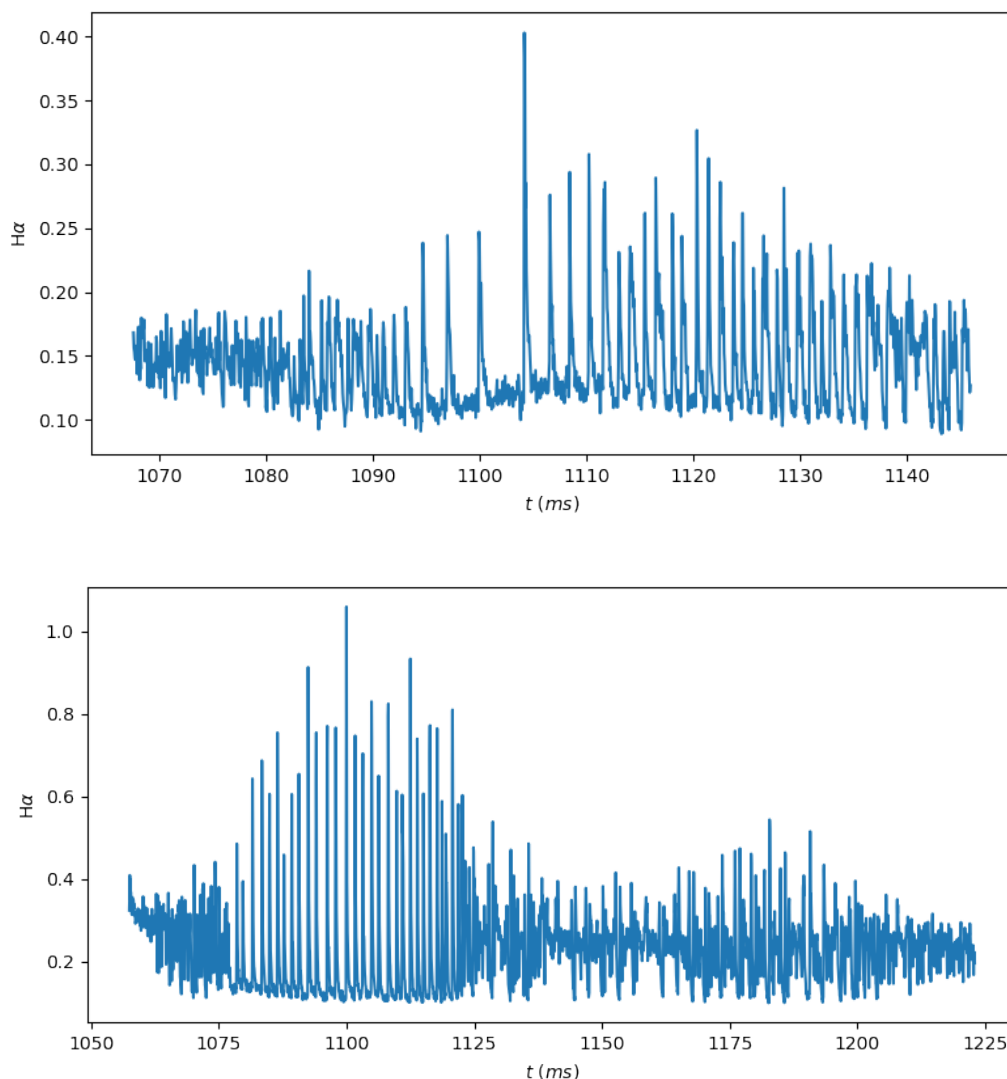
Strojové učení se zabývá počítačovými technikami a algoritmy, často za pomoci statistických metod, dávajícími počítačům schopnost se učit. Schopnost učit se nelze v tomto kontextu brát úplně doslovně, spíše je to schopnost postupně zlepšovat svou výkonnost či přesnost při řešení specifického problému. Strojové učení je velmi úzce spjato s oblastmi výpočetní statistiky a matematickou optimalizací. Za-

tímco první z nich se zaměřuje na tvorbu předpovědí nebo rozhodování s pomocí počítačů, druhá oblast poskytuje teorii, metody a v neposlední řadě aplikaci.

V této práci se budeme zabývat technikami spadajícími do dvou základních oblastí strojového učení. První z nich je známá jako učení bez učitele (unsupervised learning), která úzce souvisí s těžbou dat. Vyznačuje se tím, že algoritmus nebo metoda pracuje zásadně s neoznačenými daty tzn. neznáme výsledky.

Druhá oblast je učení s učitelem (supervised learning) lišící se předběžnou znalostí rozdělení dat. Tuto dodatečnou znalost lze pak využít k přesnějším a kvalitnějším výsledkům. Obě tyto podoblasti mají svá jedinečná uplatnění, jako jsou například: klasifikace a shlukování (tzv. clustering).

Zatímco klasifikace je jednoznačnou ukázkou učení s učitelem, při kterém algoritmus analyzuje jednotlivé jedince každé skupiny, aby odhalil, proč jsou právě v dané skupině. Shlukování je naopak příkladem učení bez učitele, při kterém jsou zkoumána všechna data za účelem nalezení vztahů mezi některými z nich. Pokud jsou nějaké takové vztahy nalezeny, jsou pak tyto data rozdělena do příslušných clusterů (shluků).



Obr. 1.3: Záření $H\alpha$ častější případy. (horní obrázek č.v.: 7880, spodní obrázek č.v.: 13484)

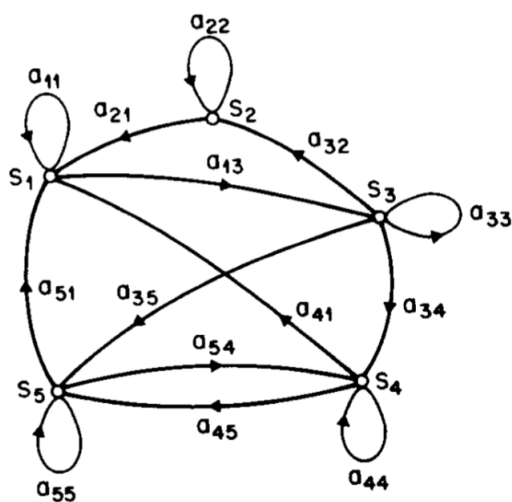
Kapitola 2

Teoretická část

V této kapitole se budeme zabývat teorií ohledně dvou námi použitých metod strojového učení. Nejprve si představíme diskrétní Markovův model. Dále se seznámíme se základními principy Skrytého Markovova modelu. Poté si přiblížíme Viterbiho algoritmus, který slouží k výpočtům již zmíněného modelu, a nakonec si představíme ryze shlukovací metodu K-means.

2.1 Diskrétní Markovův proces

Uvažujme systém, který může být popsán v každém čase pomocí jedné hodnoty z množiny N diskretních stavů. Tuto množinu budeme značit jako $\mathbb{S} = \{S_1, S_2, S_3, \dots, S_N\}$. Takový to systém může vypadat například jako Obr. 2.1 (pro přehlednost $N = 5$). Pokud je čas diskrétní a rovnoměrně rozdělen, systém mění stavy přesně podle pravděpodobností přechodů, které přísluší každému stavu. Na obrázku jsou tyto pravděpodobnosti značeny jako $a_{i,j}$ a udávají pravděpodobnost přechodu ze stavu i do stavu j .



Obr. 2.1: Markovův řetězec s 5 stavy a přechody mezi nimi. [3]

Nyní označme časy t odpovídající změnám stavů jako $t = 1, 2, 3, \dots$ a stav v čase t označme jako q_t . Pro pravděpodobnostní popis takového systému je třeba znát současný stav stejně tak jako předchozí stavy. Pokud budeme uvažovat speciální případ jímž je Markovův řetězec prvního řádu, budeme

potřebovat znát pouze současný stav a jeden předchozí stav, a to díky Markově vlastnosti (tzv. Markov Property).

Máme-li stochastický proces $(q_t)_{t \in \mathbb{N}}$ s diskrétním časem a stavovým prostorem \mathbb{S} , pak tento proces má Markovu vlastnost (je Markovův) právě tehdy, když pro všechny $t \geq 1$ je pravděpodobnostní rozdělení q_{t+1} závislé pouze na stavu q_t . Jinými slovy pro všechny $t \geq 1$ a $S_1, S_2, S_3, \dots, S_i, S_j \in \mathbb{S}$ platí

$$\mathbb{P}(q_{t+1} = S_j | q_t = S_i, q_{t-1} = S_{i_1}, \dots, q_1 = S_1) = \mathbb{P}(q_{t+1} = S_j | q_t = S_i) \quad (2.1)$$

(převzato z [4]).

Pokud je navíc tento systém nezávislý na čase t platí

$$a_{i,j} = \mathbb{P}(q_{t+1} = S_j | q_t = S_i), \quad 1 \leq i, j \leq N, \quad (2.2)$$

kde $a_{i,j} \geq 0$ a

$$\sum_j^N a_{i,j} = 1. \quad (2.3)$$

Tento stochastický proces bychom mohli nazývat pozorovatelný Markovův model, jelikož výstup procesu je množina stavů v každém časovém okamžiku a každý tento stav odpovídá fyzické (pozorovatelné) události.

2.2 Skrytý Markovův model

Skrytý Markovův model známý spíše pod svým anglickým názvem "Hidden Markov model" je statistický model, který slouží k modelování Markovských procesů se skrytými stavy. Skrytý Markovův model je široce používán v rozpoznávání řeči (speech recognition), modelování přirozeného jazyka, rozpoznávání ručně psaného písma a analýza biologických sekvencí, jako například DNA a proteinů [1].

Jedná se o rozšíření diskretního Markova modelu. Tyto modely se bohužel liší v jedné podstatné věci. Zatímco u standartního diskretního modelu, kde je možné pozorovat jednotlivé stavy, a i samotný stochastický proces, u skrytého Markova modelu je tento proces skrytý. Nicméně i tento skrytý proces může být pozorován skrze jiné stochastické procesy, jež poskytují posloupnost pozorování.

Pro lepší představu si tento model ukážeme na příkladě urn a míčků. Předpokládejme, že v místnosti je N velkých skleněných urn. V každé urně je velký počet barevných míčků. Předpokládejme, že máme M odlišných barev míčků. Fyzikální proces pro získání pozorování je následující. Džin je v místnosti a on (nebo ona) podle nějakého náhodného procesu vybírá počáteční urnu. Z této urny vybere náhodně míček a jeho barva je nahrána jako pozorování. Míček je pak nahrazen v urně, z níž byl vybrán. Další urna je vybrána procesem náhodného výběru spojeného se současnou urnou a výběr míčku je opakován. Celý proces generuje konečný počet pozorování posloupnosti barev, který bychom rádi modelovali jako pozorovaný výstup skrytého markovského modelu. (příklad převzat z [3])

Dále se na tento model můžeme dívat jako na specifický případ stavového prostorového modelu, ve kterém jsou skryté proměnné diskretní. Nicméně, když prozkoumáme jednorázový řez modelu, vidíme že odpovídá směšové distribution (viz kapitola 9 v [1]) s hustotou pravděpodobnosti danou $\mathbb{P}(\mathbf{x}|\mathbf{z})$. Proto ho můžeme interpretovat jako rozšíření směšového modelu, kde výběr složky směsi, pro každé pozorování, není nezávislý, ale závisí na volbě složek z předchozího pozorování.

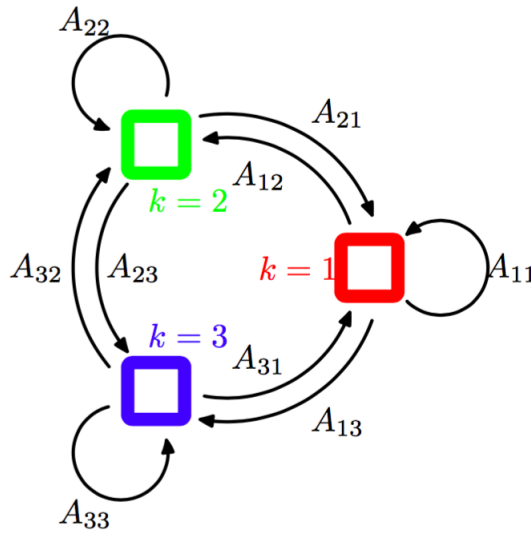
Jako v případě standartního Mixture modelu, skryté proměnné jsou diskretní multinomické proměnné z_n popisující složku směsi, jež je zodpovědná za generování příslušného pozorování \mathbf{x}_n . Od této chvíle budeme předpokládat, že skrytý proces je Markovův, tzn. splňuje Markovu vlastnost. Z tohoto předpokladu nyní plyne, že budoucí stav skryté proměnné z_n závisí pouze na předchozím stavu z_{n-1} skrze

podmíněnou pravděpodobnost $\mathbb{P}(z_n|z_{n-1})$. Pak zavedeme matici přechodů \mathbb{A} danou předpisem

$$A_{i,j} \equiv \mathbb{P}(z_{n,j}|z_{n-1,i} = 1), \quad (2.4)$$

navíc matice splňuje, že $A_{i,j} \in (0, 1)$ a skoupce jsou normovány na 1, tzn $\sum_j A_{i,j} = 1$. Dále můžeme tedy explicitně napsat podmíněné rozdělení pro K skrytých stavů ve tvaru

$$\mathbb{P}(z_n|z_{n-1}, \mathbb{A}) = \prod_{i=1}^K \prod_{j=1}^K A_{i,j}^{z_{n-1,i} z_{n,j}}. \quad (2.5)$$



Obr. 2.2: Diagram přechodů, kde skryté proměnné mají tři možné stavy odpovídající třem boxům. Šipky označují prvky matice přechodů $A_{i,j}$. [1]

Tímto vzorcem můžeme vyjádřit všechny skryté stavy až na počáteční z_1 . Tento stav má pouze marginální rozdělení $\mathbb{P}(z_1)$ reprezentované vektorem pravděpodobností π , který má tvar

$$\pi \equiv \mathbb{P}(z_{1,j} = 1) \quad (2.6)$$

a tedy

$$\mathbb{P}(z_1|\pi) = \prod_{j=1}^K \pi_j^{z_{1,j}}, \quad (2.7)$$

kde $\sum_j \pi_j = 1$. Kdybychom chtěli matici \mathbb{A} vysvětlit i jinak, mohli bychom toho docílit graficky pomocí diagramu na Obr. 2.2. Když tento diagram dále rozvineme s průběhem času, získáme takzvaný mřížový diagram, který nám poskytuje alternativní reprezentaci přechodů mezi jednotlivými skrytými stavy. Pro případ skrytého Markovova modelu tento diagram nabývá tvaru Obr. 2.3.

Abychom měli pravděpodobnostní model kompletní, je třeba ještě zavést emisní pravděpodobnosti (emission probabilities) $\mathbb{P}(x_n|z_n, \Theta)$, kde Θ představuje soubor parametrů řídicího rozdělení. Pro pozorované proměnné x_n se rozdělení $\mathbb{P}(x_n|z_n, \Theta)$ skládá z K -dimenzionálního vektoru odpovídajícího K potenciálním stavům z_n . Emisní pravděpodobnosti můžeme pak zapsat jako

$$\mathbb{P}(\mathbf{x}_n | \mathbf{z}_n, \Theta) = \prod_{j=1}^K \mathbb{P}(\mathbf{x}_n | \Theta_j)^{z_{n,j}}, \quad (2.8)$$

přičemž mohou mít například tvar Gaussova (R -rozměrného) rozdělení

$$\mathbb{P}(\mathbf{x}_n | \mathbf{z}_n) = \prod_{j=1}^K \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)^{z_{n,j}} = \prod_{j=1}^K \left(\frac{1}{(2\pi)^{R/2}} \frac{1}{|\boldsymbol{\Sigma}_j|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_j) \right\} \right)^{z_{n,j}}, \quad (2.9)$$

kde $\boldsymbol{\mu}$ je vektor středních hodnot

$$\boldsymbol{\mu} = \mathbb{E}\mathbf{X} = (\mathbb{E}X_1, \mathbb{E}X_2, \dots) \quad (2.10)$$

a $\boldsymbol{\Sigma}$ je kovarianční matice

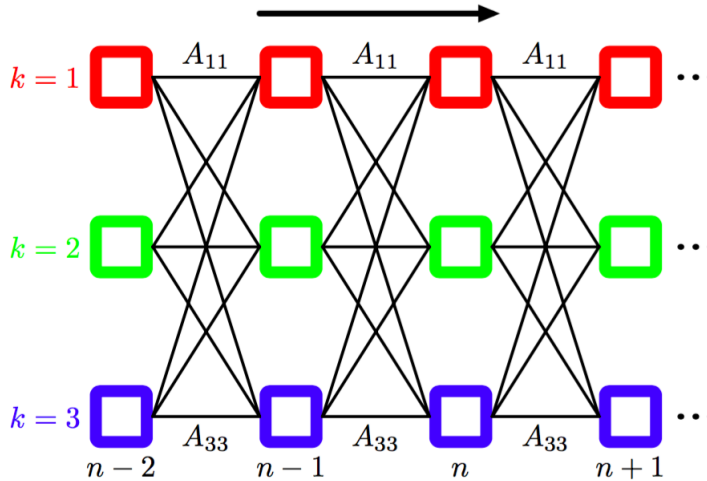
$$\boldsymbol{\Sigma} = \left(\text{Cov}(X_i, X_j) \right)_{i,j} = \left(\mathbb{E}[(X_i - \mathbb{E}X_i) \cdot (X_j - \mathbb{E}X_j)] \right)_{i,j}. \quad (2.11)$$

Nyní se zaměříme na homogenní modely, pro které všechna podmíněná rozdělení řídicí skryté proměnné sdílí stejné parametry \mathbb{A} a podobně všechna emisní rozdělení sdílí stejné parametry Θ .

Sdružené pravděpodobnostní rozdělení přes skryté i pozorované proměnné jsou pak dány vzorcem

$$\mathbb{P}(\mathbf{X}, \mathbf{Z} | \tilde{\Theta}) = \mathbb{P}(\mathbf{z}_1 | \pi) \prod_{n=2}^N \mathbb{P}(\mathbf{z}_n | \mathbf{z}_{n-1}, \mathbb{A}) \prod_{m=1}^N \mathbb{P}(\mathbf{x}_m | \mathbf{z}_m, \Theta), \quad (2.12)$$

kde $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$, $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_N)$ a $\tilde{\Theta} = (\pi, \mathbb{A}, \Theta)$ jsou parametry řídicího modelu.



Obr. 2.3: Stavový diagram Obr. 2.2 rozvinutý s časem. Každý sloupec diagramu odpovídá jedné skryté proměnné \mathbf{z}_n . [1]

2.2.1 Viterbiho algoritmus

Tento algoritmus navrhl Andrew Viterbi, již v roce 1967 [6], za účelem dekódování konvolučních kódů, jež se používají nejen v mobilních sítích, ale také ke komunikaci se satelity a sondami ve vesmíru. V současnosti se používá k rozpoznávání a syntéze řeči, vyhledávání klíčových slov, v bioinformatice nebo, což je pro nás nejdůležitější, k hledání nejpravděpodobnějších posloupností stavů.

V nejobecnější podobě se na Viterbiho algoritmus můžeme dívat jako na řešení problému maximálního aposteriorního pravděpodobnostního odhadu posloupnosti skrytých stavů konečného diskrétního Markova procesu. Tento problém je formálně identický s problémem hledání nejkratší cesty grafem. Posloupnost pozorování X každé cesty může být určena jako délka úměrná $-\ln \mathbb{P}(\mathbf{Z}, \mathbf{X})$, kde \mathbf{Z} je sequence stavů spojená s příslušnou cestou. Tento poznatek nám dovoluje řešit problém hledání posloupnosti stavů, pro které je

$$\mathbb{P}(\mathbf{Z}, \mathbf{X}) = \mathbb{P}(\mathbf{Z}|\mathbf{X})\mathbb{P}(\mathbf{X}) \quad (2.13)$$

maximální, jako problém hledání cesty jejíž délka

$$-\ln \mathbb{P}(\mathbf{Z}, \mathbf{X}) = -\ln \mathbb{P}(\mathbf{Z}|\mathbf{X}) - \ln \mathbb{P}(\mathbf{X}) \quad (2.14)$$

je minimální. Poněvadž $\ln \mathbb{P}(\mathbf{Z}, \mathbf{X})$ je monotonní funkcí $\mathbb{P}(\mathbf{Z}, \mathbf{X})$ a každá cesta odpovídá právě jedné posloupnosti stavů, pak díky platnosti Markovy vlastnosti můžeme přepsat $\mathbb{P}(\mathbf{Z}, \mathbf{X})$ jako

$$\mathbb{P}(\mathbf{Z}, \mathbf{X}) = \mathbb{P}(\mathbf{X}|\mathbf{Z})\mathbb{P}(\mathbf{Z}) = \mathbb{P}(z_1) \left[\prod_{n=2}^N \mathbb{P}(z_n|z_{n-1}) \right] \prod_{n=1}^N \mathbb{P}(x_n|z_n). \quad (2.15)$$

Po zlogaritmování (2.15) můžeme vidět, že celková délka cesty odpovídající libovolnému \mathbf{Z} je

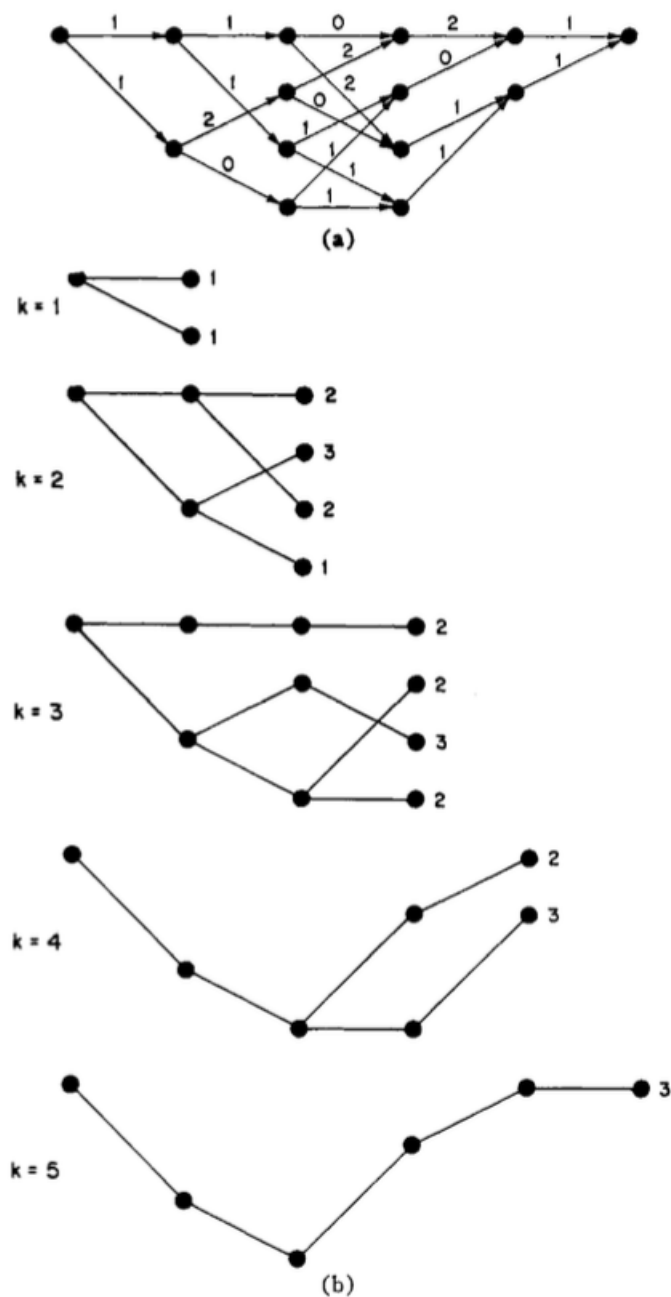
$$-\ln \mathbb{P}(\mathbf{Z}, \mathbf{X}) = \left[\sum_{n=2}^N -\ln \mathbb{P}(z_n|z_{n-1}) - \ln \mathbb{P}(x_n|z_n) \right] - \ln \mathbb{P}(z_1) - \ln \mathbb{P}(x_1|z_1), \quad (2.16)$$

kde $-\ln \mathbb{P}(z_n|z_{n-1}) - \ln \mathbb{P}(x_n|z_n)$ je délka přechodu ze z_{n-1} do z_n .

Ve chvíli, kdy jsme našli nejpravděpodobnější cestu, a tím pádem i združené rozdělení $\mathbb{P}(\mathbf{Z}, \mathbf{X})$, potřebujeme už jen nalézt posloupnost stavů odpovídající této cestě pomocí rekurze.

Jednou z největších předností Viterbiho algoritmu je jeho efektivita. Jelikož počet možných cest roste exponenciálně s délkou procesu, je tak pro většinu algoritmů výpočetně velice náročný a v některých případech i nemožný. To ovšem neplatí pro tento algoritmus, neboť výpočetní náročnost Viterbiho algoritmu roste pouze lineárně s délkou procesu (viz str. 629 v [1]).

Nyní si předvedeme na příkladu, jak přesně algoritmus funguje. Uvažujme graf viz Obr. 2.4. Každý bod představuje jeden stav a každý sloupec reprezentuje jeden časový okamžik (přechod od jednoho sloupce do druhého je právě jeden krok procesu). Šipky mezi stavy jsou možné přechody a čísla nad nimi značí délku cesty mezi stavy ($= -\ln \mathbb{P}(z_i|z_{i-1}) - \ln \mathbb{P}(x_i|z_i)$). Potřebujeme najít nejkratší (nejpravděpodobnější) cestu grafem. Začneme v bodě úplně nalevo (1. sloupec), potřebujeme dojít do bodu úplně vpravo. Podíváme se do tedy druhého sloupce, v něm se nachází dva možné stavy. Hledáme nejkratší cestu do každého z nich, ale jelikož do obou z nich vede pouze jedna cesta, je také tou nejkratší. Přejdeme tedy k dalšímu kroku a podíváme se na třetí sloupec. V němž jsou čtyři stavy a opět do nich vede jen jedna cesta. Zajímavější případ nastává tedy až při třetím kroku ve čtvrtém sloupci. V němž jsou opět čtyři stavy, ale vede do nich i více cest. Podíváme se tedy na horní stav v tomto sloupci a hledáme nejkratší cestu, jež do něj vede. Je patrné, že je to cesta přes horní bod druhého sloupce s délkou 2 ($= 1 + 1 + 0$). Stejně to provedeme i u všech stavů v tomto sloupci. Finální cesty do těchto stavů jsou vyobrazeny na čtvrtém obrázku odshora. Tento postup poté opakujeme u dalších sloupců, až nakonec získáme nejkratší cestu mezi prvním a posledním sloupcem, viz poslední obrázek. Spolu s touto cestou jsme obdrželi také nejpravděpodobnější posloupnost stavů.



Obr. 2.4: (a) Mřížový diagram s délkami přechodů. k (počet kroků) = 5. (b) Rekurzivní hledání nejkratší cesty pomocí Viterbiho algoritmu.

2.3 K-means Clustering

K-means clustering je algoritmus příslušící k učení bez učitele [1], který se používá v případě, že chceme zpracovat neoznačená data, tzn. nemáme předem definované skupiny či kategorie. Právě hlavním cílem algoritmu je najít tyto skupiny.

Předpokládejme, že máme N pozorování (x_1, \dots, x_N) náhodné R rozměrné veličiny X a chceme je rozdělit do K shluků. Pod pojmem shluk budeme rozumět skupinu bodů, jejichž vzájemné vzdálenosti

jsou mnohem menší v porovnání se vzdálenostmi k bodům vně skupiny. Dále je pak potřeba zavést si centroidy ϕ_j , kde $j \in 1, \dots, K$. Tyto centroidy jsou vektory představující středy našich shluků. Pokud všechny tyto vektory známe, můžeme na základě vzdáleností bodů od jednotlivých středů dopočítat hledané shluky. Naším cílem se tedy nyní stává nalezení množiny $\{\phi_j\}$, tak aby bylo splněno, že součet čtverců vzdáleností každého bodu shluku k nejbližšímu vektoru ϕ_j je minimální. Jinými slovy, potřebujeme minimalizovat účelovou funkci ρ definovanou jako

$$\rho = \sum_{n=1}^N \sum_{j=1}^K b_{n,j} \|\mathbf{x}_n - \phi_j\|^2, \quad (2.17)$$

kde proměnné $b_{n,j} \in \{0, 1\}$ příslušící každému bodu \mathbf{x}_n indikují, zda tento bod patří j -tého shluku nebo nikoli. Pokud ano, $b_{n,j}$ je rovno 1, v opačném případě nabývá $b_{n,j}$ hodnoty 0. Minimálního ρ lze dosáhnout pomocí dvoufázového iteračního procesu, ve kterém dochází k postupné minimalizaci účelové funkce, dokud nenastane konvergence. Nejprve vybereme, nejlépe náhodně, počáteční vektory ϕ_j . V první fázi bereme ϕ_j jako fixní a minimalizovat budeme s ohledem na $b_{n,j}$. Jelikož ρ je vůči $b_{n,j}$ lineární a pro rozdílná n jsou $b_{n,j}$ nezávislé, můžeme je minimalizovat pro každé n zvlášť. Jednoduše bereme $b_{n,j}$ následně

$$b_{n,j} = \begin{cases} 1 & \text{pokud } \operatorname{argmin}_m \|\mathbf{x}_n - \phi_m\|^2 = j \\ 0 & \text{jinde.} \end{cases} \quad (2.18)$$

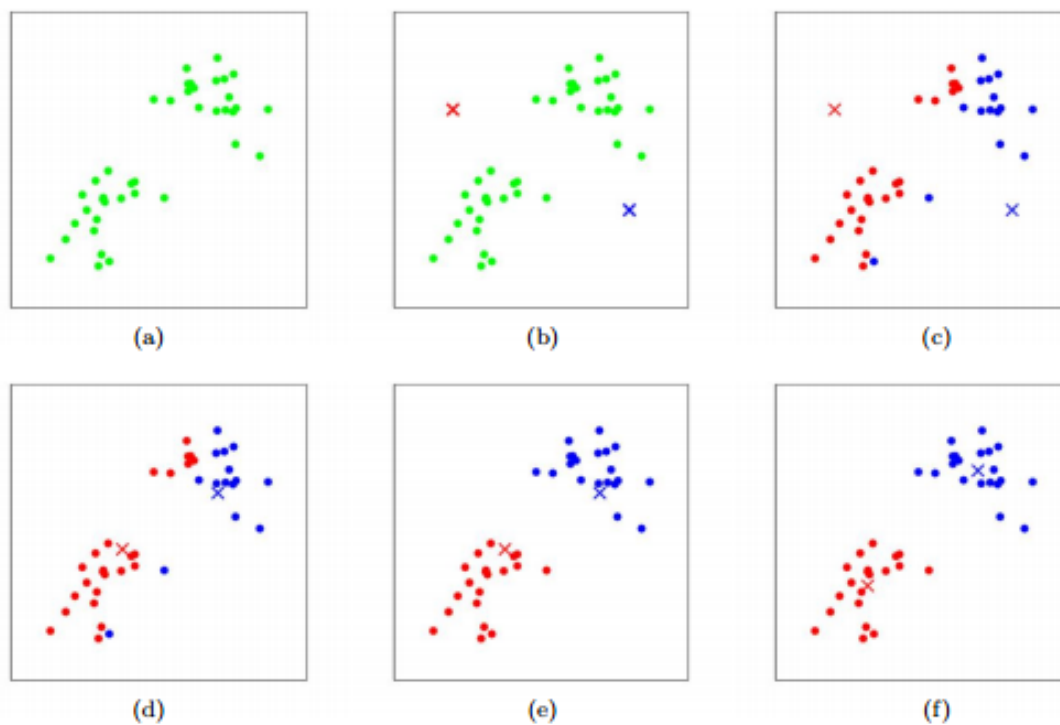
V druhé fázi je naopak $b_{n,j}$ pevné a minimalizujeme s ohledem na ϕ_j . Proto zderivujeme funkci ρ podle ϕ_j a tuto derivaci položíme rovnu 0, tzn.

$$2 \sum_{n=1}^N b_{n,j} (\mathbf{x}_n - \phi_j) = 0. \quad (2.19)$$

Načež po osamostatnění ϕ_j získáváme konečný tvar

$$\phi_j = \frac{\sum_{n=1}^N b_{n,j} \mathbf{x}_n}{\sum_{n=1}^N b_{n,j}}, \quad (2.20)$$

kde jmenovatel představuje celkový počet bodů patřících do j -tého shluku. Vzorec (2.20) lze ovšem interpretovat také jako střední hodnotu (anglicky: mean) všech bodů příslušících do shluku j , odtud plyne i název "K-means".



Obr. 2.5: Vizualizace algoritmu K-means. Tréninková data jsou vyobrazena jako tečky a středy shluků jako křížky. Na obrázku (a) jsou vykreslena původní data. V (b) je možné vidět náhodně vybrané počáteční středy a (c) – (f) ilustrují úvodní dvě iterace algoritmu. [15]

Kapitola 3

Praktická část - experimenty

V této kapitole se budeme zabývat praktickou částí, ve které si nejdříve předsavíme příznaky, bez kterých by naše algoritmy strojového učení nefungovaly. Dále budou uvedeny tři hlavní experimenty nejdříve na syntetických datech a později i na reálných. Všechny tyto experimenty byly programovány v jazyce Python za použití knihoven numpy, numba, matplotlib, scipy a hmmlearn.

3.1 Příznaky (Features)

Ve strojovém učení a rozpoznávání vzorů se pod pojmem "příznak" (feature) rozumí individuální měřitelná vlastnost nebo charakteristika pozorovaného jevu. Výběr těchto příznaků je naprosto zásadní pro efektivní rozpoznávací, regresní a klasifikační algoritmy. Čím relevantnější a charakterističtější příznak, tím lépe jsme schopni docílit větší přesnosti modelu. Na druhou stranu vynechání zbytečných, případně méně důležitých příznaků zase snižuje složitost modelu a urychluje jeho trénink. Nejčastější forma příznaku je číselná hodnota, avšak při rozpoznávání syntetických vzorků se hojně používají i písmena, slova nebo grafy.

Selekci těchto příznaků je možné demonstrovat na následujícím příkladu. Předpokládejme, že bychom chtěli předvídat typ domácího mazlíčka, jež si někdo koupí.

Do příznaků můžeme zahrnout například věk osoby, pohlaví, jméno, bydlení (byt, dům, ...), rodinný příjem, vzdělání a počet dětí. Je zřejmé, že většina těchto příznaků nám může při předvídaní pomoci, ale některé jako třeba vzdělání nebo jméno jsou zjevně méně důležité.

jméno	věk	pohlaví	bydlení	příjem	počet dětí	vzdělání
Karel	25	muž	byt	30.000	0	středoškolské
Petr	30	muž	dům	45.000	2	vysokoškolské
Jana	42	žena	byt	23.000	1	základní
Miloš	51	muž	dům	29.000	1	středoškolské

...

Tabulka 3.1: Vzorová tabulka příznaků k demonstračnímu příkladu

3.1.1 První a druhá derivace

Prvním příznakem použitým při klasifikaci je první derivace. Jelikož jsou k dispozici pouze jednotlivé body, nelze použít analytický vzorec pro derivace funkce $f(x) : \mathbb{R} \rightarrow \mathbb{R}$, tedy konkrétně

$$\frac{df(x)}{dx} = f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}. \quad (3.1)$$

Namísto toho se musí počítat numericky, a to za použití centrální diference druhého řádu pomocí (3.2) a v krajních bodech pomocí jednostranných diferencí prvního nebo druhého řádu (3.3).

$$\hat{f}'_k = \frac{f(x_{k+1}) - f(x_{k-1}))}{2h} \quad (3.2)$$

$$\hat{f}'_0 = \frac{f(x_1) - f(x_0)}{h} \quad \text{a} \quad \hat{f}'_n = \frac{f(x_n) - f(x_{n-1}))}{h} \quad (3.3)$$

Dalším použitým příznakem je druhá derivace, kterou lze je snáze získat použitím výše zmíněných vzorců (3.2) a (3.3) na již jednou zderivovaný signál.

3.1.2 Savitzky-Golay filtr

Na obrázku 1.3 v první kapitole bylo možné vidět, že naše data získaná z detektoru jsou zatížena velkým šumem. Proto je vhodné pokusit se tento signál nějak vyhladit. Za tímto účelem byl mezi příznaky vybrán Savitzky-Golay filtr, jež je digitálním filtrem dobře přizpůsobeným pro vyhlazování dat. S-G filtry byly zpočátku použity k zobrazení relativních šířek a výšky spektrálních čar v zašuměných spektrometrických datech. (převzato z [8])

Digitální filtr aplikovaný na stejnoměrně rozložená data, tzn. $f_i = f(t_i)$, kde $t_i = t_0 \cdot \Delta$, $i \in \mathbb{Z}$ a Δ je konstanta, nahrazuje každou hodnotu f_i hodnotou g_i , jež lineární kombinací určitého počtu nejbližších sousedů bodu f_i ,

$$g_i = \sum_{n=-n_L}^{n_R} c_n f_{i+n} = (\mathbf{c} * \mathbf{f})_i, \quad (3.4)$$

kde \mathbf{c} jsou tzv. konvoluční koeficienty, n_L je počet použitých bodů vlevo a n_R vpravo.

Hlavní myšlenou S-G filtru je aproximace funkce uvnitř pohybujícího se okna pomocí polynomu namísto konstanty. Pro každou hodnotu f_i proložíme všech $n_L + n_R + 1$ bodů, uvnitř pohyblivého okna, polynomem pomocí metody nejmenších čtverců a nastavíme g_i na hodnotu polynomu na i -té pozici. Jelikož nevyužíváme hodnoty polynomu v jiných bodech, měli bychom tedy pro f_{i+1} udělat celou proceduru znovu. Naštěstí díky tomu, že metoda nejmenších čtverců pro výpočet využívá pouze lineární maticovou inverzi a koeficienty proloženého polynomu jsou sami o sobě také lineární, můžeme veškeré prokládání vypočítat dopředu a pomocí binárního vektoru lze pak vše dopočítat lineární kombinací.

Potřebujeme tedy proložit polynom řádu M konkrétně $a_0 + a_1 i + \dots + a_M i^M$ hodnotami f_{-n_L}, \dots, f_{n_R} . Matice pro nejmenší čtverec má tvar

$$A_{ij} = i^j \quad i = -n_L, \dots, n_R \quad \text{a} \quad j = 0, \dots, M \quad (3.5)$$

vektro koeficientů polynomu \mathbf{a} lze pak vypočítat jako

$$\mathbf{a} = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot (\mathbf{A}^T \cdot \mathbf{f}). \quad (3.6)$$

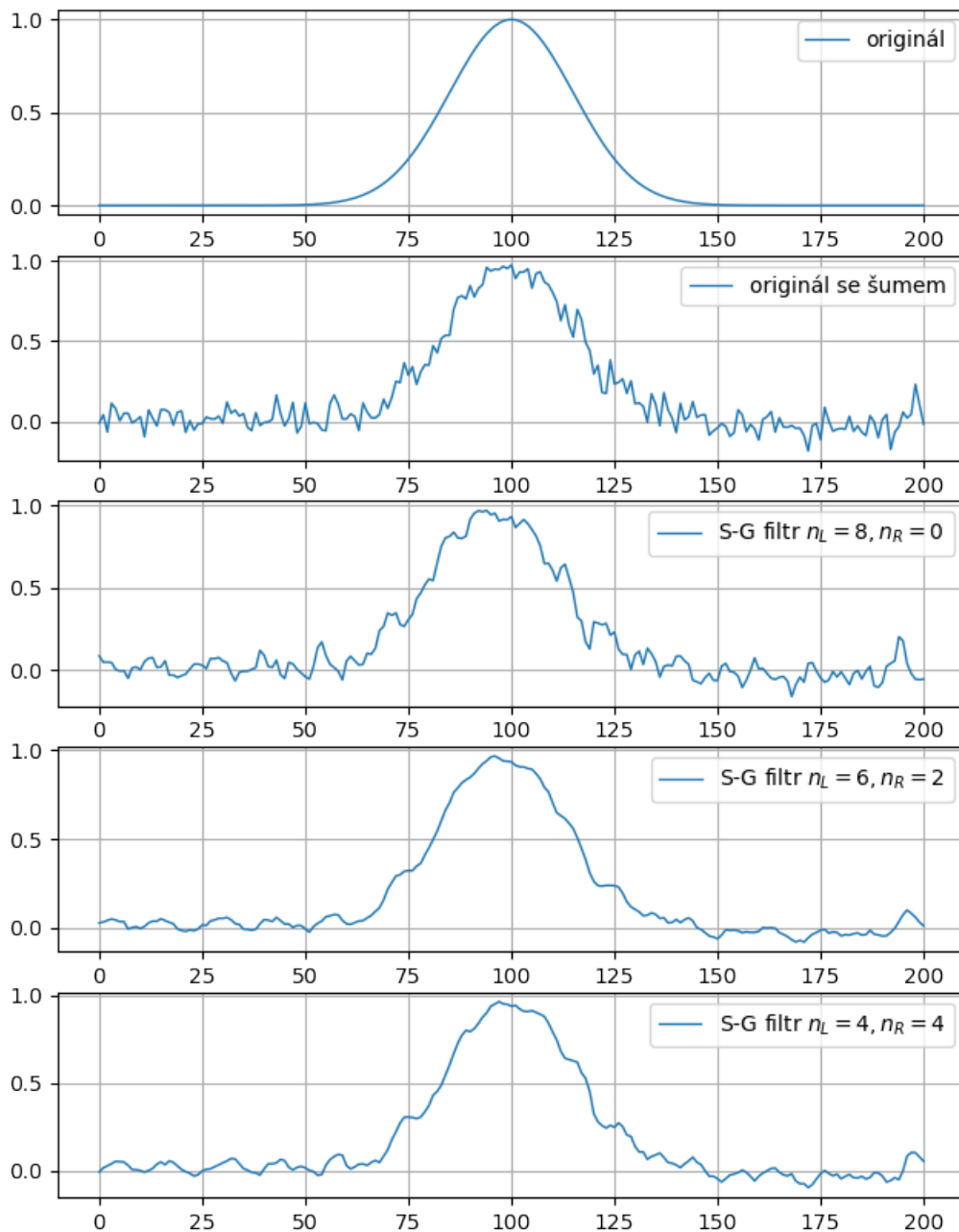
Nakonec vypočítáme i vektro koeficientů \mathbf{c} pomocí

$$\mathbf{c} = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \quad (3.7)$$

a po dosazení zpět do (3.4) získáváme konečně vyhlazenou funkci g_i .

S-G filtr je pro řešení našeho problému výhodný také z důvodu, že po mírné úpravě lze takto počítat i vyhlazené derivace signálu. Tato úprava sočívá v záměně derivace funkce f za derivaci polynomu a to díky vlastnosti konvoluce

$$(c * f') = (c' * f). \quad (3.8)$$



Obr. 3.1: Savitzky-Golay filtr aplikovaný na gaussovu křivku s délkou úseku 9 ($n_L + n_R + 1 = 9$). Na obrázcích je patrné, že posunutím bodu, v němž probíhá aproximace, balancujeme mezi zpožděním a velikostí šumu.

3.1.3 Klouzavý průměr

Dalším vybraným příznakem je klouzavý průměr. Klouzavý průměr je diskretní lineární filtr s konečnou dobou odezvy, který slouží k vyhlazení signálu. Nadále ho budeme značit jako \tilde{X}_t . Necht' máme vektor naměřených hodnot $\mathbf{X} = (x_1, x_2, \dots, x_n)$, pak definuji klouzavý průměr pro $t \in 1, 2, \dots, n$ jako

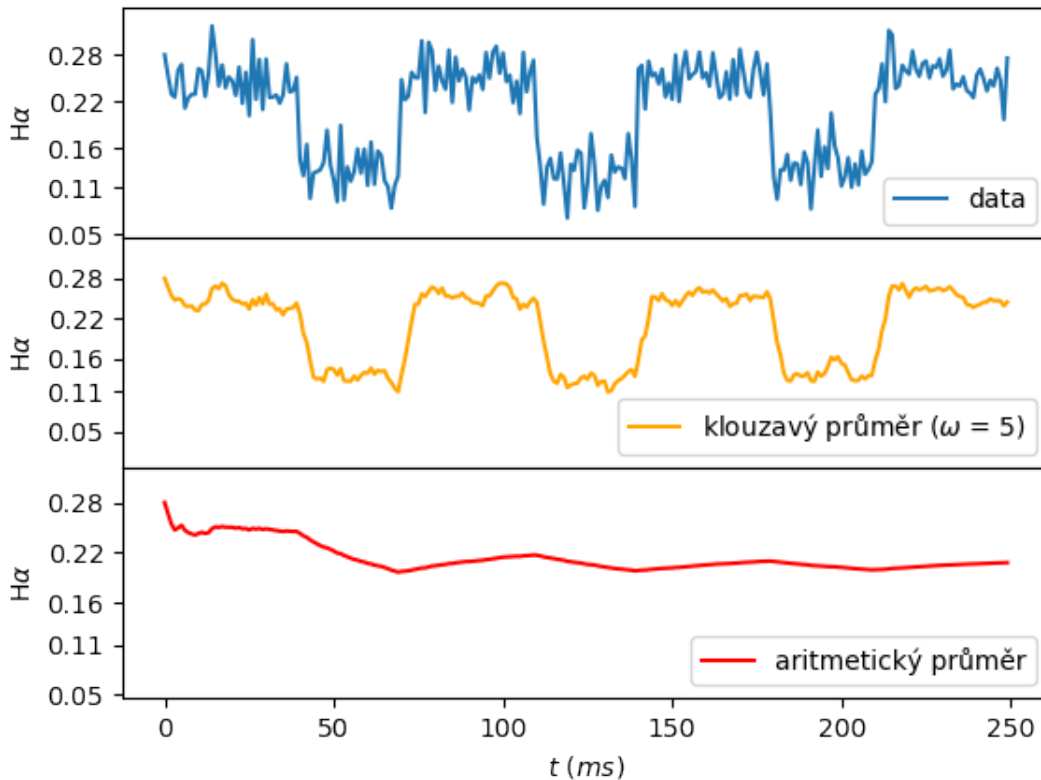
$$\tilde{X}_t = \frac{1}{\tilde{\omega}} \sum_{k=t-\tilde{\omega}}^t x_k, \quad (3.9)$$

kde $\tilde{\omega} = \min\{\omega, t\}$, přičemž ω je délka úseku (doba odezvy). Pak vektor $\tilde{\mathbf{X}} = (\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_n)$ je příznak vektoru \mathbf{X} .

Ve skutečnosti se jedná o standartní aritmetický průměr (3.10), jež je aplikovaný pouze na úsek dat konečné délky.

$$\bar{X}_n = \frac{1}{n} \sum_{k=1}^n x_k, \quad (3.10)$$

Mezi příznaky byl vybrán, protože předpokládáme, že okamžitá hodnota je závislá na předchozích datech. Důvod, proč využíváme jen konečně dlouhý úsek předcházejících hodnot je ten, že ze zákona velkých čísel aritmetický průměr konverguje ke střední hodnotě, a tedy ke konstantě. To znamená, že postupem času budou mít rozdílná data v odlišných stavech stejnou hodnotu příznaku. Z čehož plyne, že takovýto příznak by jen zkresloval a znepřesňoval výsledek, viz Obr. 3.2.



Obr. 3.2: Rozdíl mezi klouzavým a aritmetickým průměrem aplikovaným na syntetická data

3.1.4 Exponenciální klouzavý průměr

Již dříve jsme se zmínili, že předpokládáme závislost na předchozích hodnotách. Nicméně je zřejmé, že hodnoty naměřené s velkým časovým rozestupem na sebe mají mnohem menší vliv než ty, jež jsou naměřeny bezprostředně za sebou. Proto dalším vybraným příznakem je tedy exponenciální klouzavý průměr S_t .

Nechť $X = (x_1, x_2, \dots, x_n)$ je vektor naměřených hodnot, pak definuji váhový součet zleva pro $t \in 1, 2, \dots, n$ jako

$$S_t = \frac{1}{\tilde{\omega}} \sum_{k=t-\tilde{\omega}}^t g_{t-k} \cdot x_k, \quad (3.11)$$

kde $\tilde{\omega} = \min\{\omega, t\}$, ω je délka úseku a g_m je váhová funkce tvaru $g_m = \gamma^m$, přičemž $\gamma \in (0, 1)$. Pak vektor $S = (S_1, S_2, \dots, S_n)$ je příznak vektoru X . Díky exponenciální váhové funkci g_m , jsme tedy schopni snížit důležitost více vzdálených dat, což byl náš záměr.

3.1.5 Klouzavý rozptyl

Ve statistice a teorii pravděpodobnosti se pod pojmem rozptyl rozumí střední hodnota kvadrátu odchylky od střední hodnoty náhodné veličiny. Bývá reprezentován symbolem $Var(X)$ nebo σ^2 a definován vzorcem

$$Var(X) = E[(X - E[X])^2] \quad (3.12)$$

pro stejně rozdělené diskrétní náhodné veličiny $X = (x_1, x_2, \dots, x_n)$ můžeme tento vzorec přepsat do tvaru

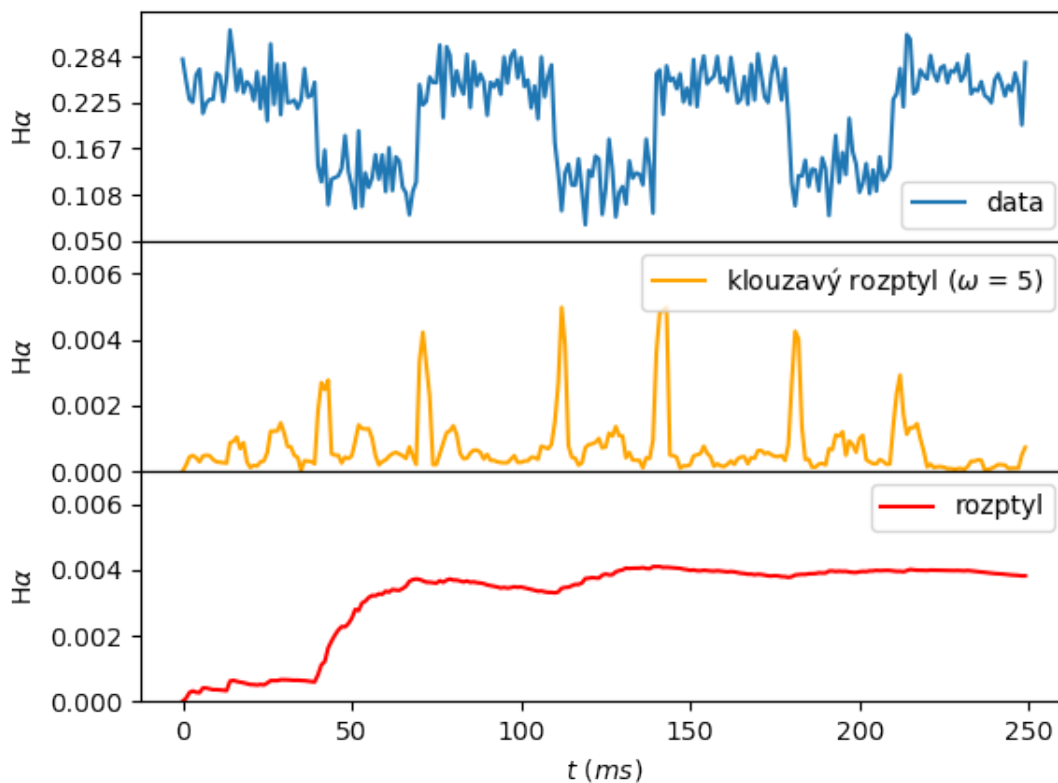
$$Var(X) = \frac{1}{n} \sum_{k=1}^n (x_k - \bar{X}_n)^2. \quad (3.13)$$

Bohužel, rozptyl nemůžeme jako příznak použít ze stejného důvodu jako aritmetický průměr, protože s postupem času bude různým stavům přiřazovat stejnou hodnotu. Proto zde využijeme místo aritmetického průměru již dříve definovaný klouzavý průměr a výslednou veličinu budu dále nazývat klouzavým rozptylem a značit D_t .

Nechť $X = (x_1, x_2, \dots, x_n)$ je vektor naměřených hodnot, pak definuji úsekový rozptyl pro $t \in 1, 2, \dots, n$ jako

$$D_t = \frac{1}{\tilde{\omega}} \sum_{k=t-\tilde{\omega}}^t (x_k - \tilde{X}_t)^2, \quad (3.14)$$

kde $\tilde{\omega} = \min\{\omega, t\}$, w je opět délka úseku a \tilde{X}_t je klouzavý průměr (3.9). Pak vektor $D = (D_1, D_2, \dots, D_n)$ je příznak vektoru X .



Obr. 3.3: Rozdíl mezi úsekovým a normálním rozptylem aplikovaným na syntetická data

3.2 Skrytý Markovův model vs K-means na syntetických datech

V této podkapitole se budeme zabývat prvním experimentem, a to porovnáním skrytého Markovova modelu a shlukovou metodou K-means. Skrytý Markovův model budeme při tomto experimentu používat ve formě učení bez učitele.

3.3 Skrytý Markovův model vs K-means na reálných datech

3.3.1 Detekce H-modu

3.3.2 Obecné datasety

Kapitola 4

Výsledky

4.1 Způsoby vyhodnocení výsledků

V této podkapitole se budeme věnovat způsobům jak vyhodnocovat kvalitu námy zvolených metod. Nejčastěji používaná je přesnost (accuracy), jako prvotní orientační náhled na kvalitu modelu je dostačující. Ovšem nelze jí nekompromisně věřit v každé situaci. Z tohoto důvodu budou použity i další metriky, jako precision, recall a F-míra.

4.1.1 Confusion Matrix (matice záměn)

Confusion matrix je ve strojovém učení velmi dobře známa. Představuje jakousi tabulku s přesně daným rozložením, kterou lze využít k vizualizaci či popisu výkonosti daného modelu. Dává nám přehled o chybách, kterých se model dopouští a navíc i druh těchto chyb. Využívá se standardně při učení s učitelem, kde jsou k dispozici označená data.

Nejprve je třeba vybrat testovací data, u kterých jsou nám známy výsledky. Na tyto data je následně aplikován již natrénovaný model. Výstup je pak porovnán se skutečnými výsledky a zaznamenám právě do confusion matrix. Z ní jsme pak schopni vypočítat většinu výkonostních metrik.

Výsledky klasifikace	Skutečné výsledky		
	skutečný stav 0	skutečný stav 1	skutečný stav 2
predikovaný stav 0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$
predikovaný stav 1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$
predikovaný stav 2	$C_{2,0}$	$C_{2,1}$	$C_{2,2}$

Obr. 4.1: Confusion matrix \mathbb{C} pro případ 3 skupin. Diagonální prvky $C_{0,0}$, $C_{1,1}$ a $C_{2,2}$ udávají kolikrát byl konkrétní stav predikován správně. Prvek $C_{i,j}$, kde $i, j \in \{0, 1, 2\}$ a $i \neq j$, udává kolikrát byl predikován stav i a ve skutečnosti se jednalo o stav j .

4.1.2 Přesnost

Přesnost (anglicky accuracy) udává poměr mezi správně klasifikovanými stavy vůči celkovému počtu bodů.

$$\text{Accuracy} = \frac{\text{correct}}{\text{all}} = \frac{C_{0,0} + C_{1,1} + C_{2,2}}{\sum_{i,j=0}^2 C_{i,j}} \quad (4.1)$$

Přesnost má bohužel jednu velkou nevýhodu, kterou ukážeme na příkladu. Uvažujme proces nabývajících pouze dvou stavů, kde jeden stav nastává mnohem častěji než druhý. Měřením jsme získali dataset obsahující 200 bodů, ve kterém se druhý stav vyskytl pouze 10 krát. Když byl pak na tyto data použit již natrénovaný model, výstupní sequence stavů byly samé jedničky (první stav na všech bodech). Pokud bychom spočítali přesnost takového modelu, vyšlo by 95%. Ale i přes takto vysokou přesnost model neodhalil ani jeden výskyt druhého stavu. Což je v případě, že je tento druhý stav důležitý, naprosto nedostačující.

4.1.3 Precision

Precision neboli preciznost udává v procentech, jak moc se dá zvolenému modelu věřit. Prozrazuje nám tedy jaká je pravděpodobnost, že když model předpoví nějaký stav, tak tento stav opravdu nastal. Precision stavu $i \in \{0, 1, 2\}$ lze dopočítat z confusion matrix pomocí vzorce

$$\text{Precision}_i = \frac{C_{i,i}}{\sum_{j=0}^2 C_{i,j}} \quad (4.2)$$

4.1.4 Recall

Recall udává, jaký podíl všech skutečných stavů, které model odhalí. Vysoká hodnota recallu značí správnost rozpoznání dané skupiny. Recall stavu $i \in \{0, 1, 2\}$ lze opět dopočítat z confusion matrix pomocí vzorce

$$\text{Recall}_i = \frac{C_{i,i}}{\sum_{j=0}^2 C_{j,i}} \quad (4.3)$$

4.1.5 F míra

F-míra známá jako F1-score nebo F measure je harmonický průměr mezi precision a recall. F-míra dosahuje maximální hodnoty 1 (100%) právě tehdy, když oba precision i recall jsou nejlepší. Pro stav $i \in \{0, 1, 2\}$ je F-míra definována jako

$$\text{F-míra}_i = \frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} = \frac{C_{i,i}}{\sum_{j=0}^2 C_{j,i} + \sum_{j=0}^2 C_{i,j}} \quad (4.4)$$

4.2 Skrytý Makrovův model vs K-means na syntetických datech výsledky

4.3 Skrytý Markovův model vs K-means na reálných datech - výsledky

4.3.1 Detekce H-modu

4.3.2 Obecné datasety

Závěr

Text závěru....

Literatura

- [1] C. M. Bishop, Pattern recognition and machine learning. Springer, New York, 2013.
- [2] M. Kikuchi, K. Lackner, M. Q. Tran, Fusion physics. International Atomic Energy Agency, Vienna, 2012.
- [3] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 1989, 257-286.
- [4] N. Privault, Understanding Markov Chains: Examples and Applications. Springer, 2013.
- [5] G. D. Forney, The Viterbi Algorithm. Proceedings of the IEEE 61(3) 1973, 268-278.
- [6] A.J. Viterbi, "Error bounds for convolutional codes a an asymptotically optimum decoding algorithm". IEEE Transactions on Information Theory 13 (2),1967, 260–269.
- [7] P. Harrington, Machine Learning in Action. Minning Publications Co. Greenwich, 2012.
- [8] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.F. Flannery, Numerical Recepies in C: The Art of Scientific Computing (Second Edition). New York: Cambridge University Press. ISBN 0-521-43108-5, 1992.
- [9] R.J. Goldston, P.H. Rutherford. Introduction to Plasma Physics. Taylor and Francis. ISBN 978-0-7503-0183-1, 1995.
- [10] R. Pánek, O. Bilyková, V. Fuchs, M. Hron, P. Chráska, P. Pavlo, J. Stöckel, J. Urban, V. Weinzettl, J. Zajac, F. Žáček, "Reinstallation of the COMPASS-D tokamak in IPP ASCR". Czechoslovak Journal of Physics. ISSN 1572-9486, 2006.
- [11] F. Wagner, "A quarter-century of H-mode studies". Plasma Physics and Controlled Fusion. 49: B1. Bibcode:2007PPCF...49....1W. doi:10.1088/0741-3335/49/12B/S01 2007.
- [12] https://lib.ugent.be/fulltxt/RUG01/001/458/765/RUG01-001458765_2011_0001_AC.pdf
- [13] http://doks.xios.be/doks/do/files/FiSe8ae680b43c26317b013c953679dd020b/200800040_12.pdf?recordId=Sxh
- [14] http://www.ipp.cas.cz/vedecka_struktura_ufp/tokamak/tokamak_compass/index.html
- [15] <http://stanford.edu/~cpiech/cs221/img/kmeansViz.png>
- [16] https://www.researchgate.net/profile/Janos_Egert/publication/283617947/figure/fig1/AS:314833057665024@14of-the-COMPASS-tokamak-left-and-its-vacuum-chamber-with-diagnostic-ports-right.png