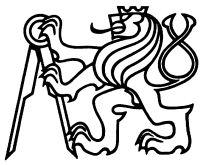
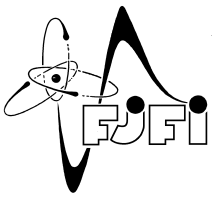


České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská



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 nejden 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

Název práce:

Těžba dat z experimentů na tokamaku COMPASS

Autor: Matěj Zorek

Obor: Matematické inženýrstvý

Zaměření: Aplikované matematicko-stochastické metody

Druh práce: Bakalářská práce

Vedoucí práce: Ing. Vít Škvára, Ústav fyziky plazmatu, AV ČR Za Slovankou 1782/3 182 00 Praha 8

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

Abstrakt: Abstrakt max. na 10 řádků.

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

Title:

Data mining on the COMPASS tokamak experiments

Author: Matěj Zorek

Abstract: Max. 10 lines of English abstract text.

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 Podrobnější popis problematiky . . . 14
1.4 Strojové učení . . . 15
1.4.1 Klasifikace vs clustering . . . 16

2 Teoretická část **17**
2.1 Skrytý Markovův model . . . 17
2.1.1 Viterbiho algoritmus . . . 19
2.2 K-means Clustering . . . 21

3 Postup práce a výsledky **23**
3.1 Rysy (Features) . . . 23
3.1.1 První a druhá derivace . . . 23
3.1.2 Savitzky-Golay filtr . . . 24
3.1.3 Klouzavý průměr . . . 25
3.1.4 Exponenciální klouzavý průměr . . . 26
3.1.5 Klouzavý rozptyl . . . 26
3.2 Způsoby vyhodnocení výsledků . . . 28
3.2.1 Přesnost . . . 28
3.2.2 Správnost . . . 28
3.2.3 Recall . . . 28
3.2.4 F míra . . . 28

Závěr **29**

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 zařízeníxh 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 budu 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 clusteringová metoda K-means.

Ve třetí a poslední kapitole jsou postupně popsány rysy 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 a jeho dvě hlavní skupiny.

1.1 Plazma

Plazma je jedním ze čtyř základních skupenství. Jedná se o ionizovaný plyn, jehož atomy jsou rozděleny na pozitivní ionty a naegativní 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. Obykle 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.

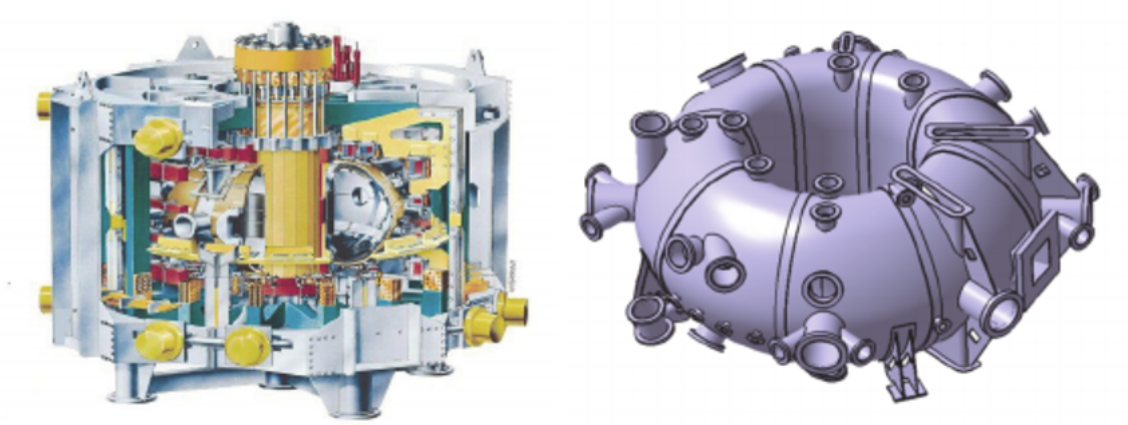
1.2 Tokamak COMPASS

Název tokamak je zkratkou původního ruského názvu toroidalnaja kamera s magnitnymi katuškami (toroidní komora s magnetickými cívkami). V podstatě se jedná o experimentální zařízení využívné 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 fůze. Pokud by se podařilo fůzy 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 fůze 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 nabytými částicemi plazmatu.

Tokamak COMPASS je umístěn v Praze Ládví na Ústavu fyziky plazmatu Akademie věd České republiky již od roku 2004. Původně byl umístěn a provozován ve Velké Británii pod UKAEA (UK Atomic Energy Autority) 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 tokuamaku 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. [11]

1.3 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 reakci na delší dobu a téměř nemožné udržet jí dlouhodoběji.

Druhým stavem je již dříve zmínění H-mód. V toto stavu je možné lépe kontrolovat chování plazmy a především jí držet v rovnováze po delší dobu. Tento stav je také standartním referenčním režimem budoucího tokamaku ITER.

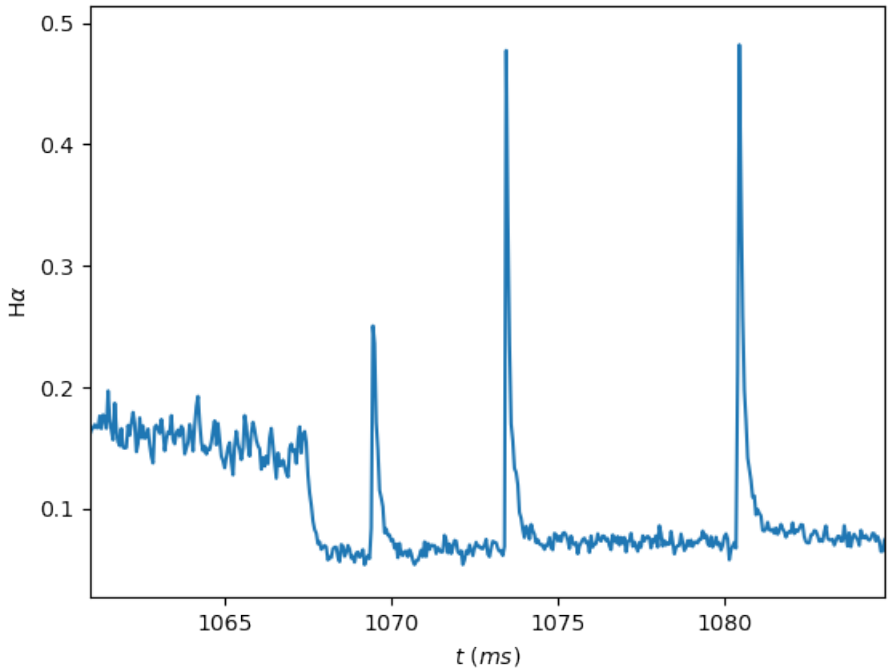
Třetím v řadě je ELM (edge-localized mode). ELMy jsou v podstatě narušující nestability, k nímž dochází na okraji plazmy. ELMy se navíc mohou vyskytovat pouze během H-módu. Posledním stavem je disrupce. Disrupce představuje fyzikální jev, během něhož dochází k přetržení nebo náhlé zrátě udžení plazmatu.

Aby bylo možné správně porozumět těmto jevům je zapotřebí velkého množství informací. Příkladem může být teplota během jednotlivých stavů atd. Bohužel většina detektorů a měřících 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[1] a metody strojového učení.

[1] H_α je specifická červená spektrální čára vyzařovaná vodíkem s vlnovou délkou $656,28nm$

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, nevyzařuje plazma tolik vodíku jako v L-módu. Jednotlivé peaky jsou pak ELMy. Ve vákuu, 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 pěkný ELM. V praxi však signály vypadají spíše jako na Obr. 1.3, kde už některé stavy nejsou jednoznačně viditelné.



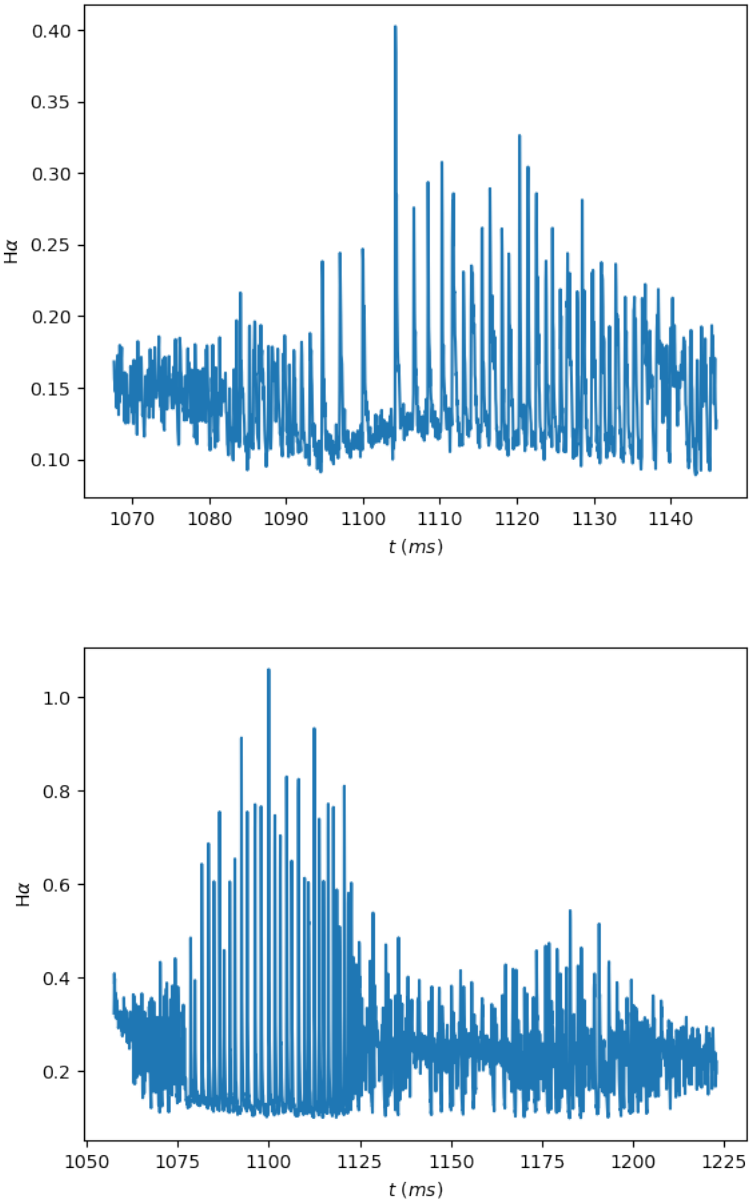
Obr. 1.2: Ukázka ideálně vypadajícího záření H_α zaznamenaného detektrorem. (číslo výstřelu: 15364)

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

Strojové učení dělíme do dvou hlavních skupin. První z nich je známo jako učení bez učitele (unsupervised learning), které úzce souvisí s těžbou dat. Vyznačuje se tím, že algoritmus nebo metoda pracuje zásadně s neoznačenými daty tzn. neznáme výsledky.

Druhou skupinou 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 skupiny mají svá jedinečná uplatnění, jako jsou například: regrese, klasifikace, clustering atd.



Obr. 1.3: Záření H_α častější případy. (horní obrázek č.v.: 7880, spodní obrázek č.v.: 13484)

1.4.1 Klasifikace vs clustering

Klasifikace a clustering jsou dvě strany jedné mince. Obě metody slouží k rozdělování pozorovaných dat do různých skupin. Toto členění probíhá na základě analýzy vlastností, podobností nebo nějakého skrytého vzoru. Zatímco klasifikace je jednoznačnou ukázkou učení s učitelem, při kterém algoritmus analyzuje jednotlivé jedince každé skupiny, aby odhalil, proč jsou právě v dané skupině. Clustering 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).

Kapitola 2

Teoretická část

V této kapitole se budeme zabývat teorií ohledně dvou námi použitých metod strojového učení. Nejprve 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 clusteringovou metodu K-means.

2.1 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. Přesněji se jedná o dvojnásobně zapuštěný stochastický proces podložený dalším stochastickým procesem, který není pozorován, je tedy skrytý. Nicméně tento skrytý proces může být pozorován skrze jiné stochastické procesy, jež poskytují posloupnost pozorování.

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.

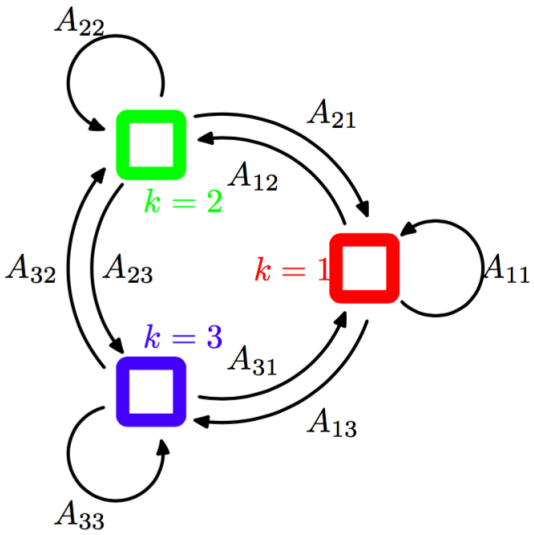
Pro lepší představu si tento model ukážeme na příkladě uren a míčků. Předpokládejme, že v místnosti je N velkých skleněných uren. V každé urně je velký počet barevných míčků. Předpokládejme, že máme M odlišných barev míčků. Fyzikální proces pro získání pozorování je následující. Džin je v místnosti a 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 diskrétní. Nicméně, když prozkoumáme jednorázový řez modelu, vidíme že odpovídá Mixture distribution (viz kapitola 9 v [1]) s hustotou pravděpodobnosti danou $\mathbb{P}(\boldsymbol{x}|\boldsymbol{z})$. Proto ho můžeme interpretovat jako rozšíření Mixture 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 diskrétní multinomické proměnné $\boldsymbol{z}_n$ popisující složku směsi, jež je zodpovědná za generování příslušného pozorování $\boldsymbol{x}_n$. Od této chvíle budeme předpokládat, že skrytý proces je Markovův, tzn. splňuje Markovu vlastnost (Markov property)[4]. Z tohoto předpokladu nyní plyne, že budoucí stav skryté proměnné $\boldsymbol{z}_n$ závisí pouze na předchozím stavu $\boldsymbol{z}_{n-1}$ skrze podmíněnou pravděpodobnost $\mathbb{P}(\boldsymbol{z}_n|\boldsymbol{z}_{n-1})$. Pak zavedeme matici přechodů $\mathbb{A}$ danou předpisem $A_{i,j} \equiv \mathbb{P}(\boldsymbol{z}_{n,j}|\boldsymbol{z}_{n-1,i} = 1)$, 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

$$\mathbb{P}(z_n|z_{n-1}, \mathbb{A}) = \prod_{i=1}^{K}\prod_{j=1}^{K} A_{i,j}^{z_{n-1,i}z_{n,j}}. \tag{2.1}$$



Obr. 2.1: 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 $\pi \equiv \mathbb{P}(z_{1,j} = 1)$ a tedy

$$\mathbb{P}(z_1|\pi) = \prod_{j=1}^{K} \pi_j^{z_{1,j}}, \tag{2.2}$$

kde $\sum_j \pi_j = 1$. Kdybychom chtěli matici $\mathbb{A}$ vysvětlit i jinak, mohli bychom toho docílit graficky pomocí diagramu na Obr. 2.1. 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. 2.2.

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

$$\mathbb{P}(\boldsymbol{x}_n|z_n, \Theta) = \prod_{j=1}^{K} \mathbb{P}(\boldsymbol{x}_n|\Theta_j)^{z_{n,j}}, \tag{2.3}$$

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

$$\mathbb{P}(\boldsymbol{x}_n|z_n) = \prod_{j=1}^{K} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \Sigma_j)^{z_{n,j}} = \prod_{j=1}^{K} \left(\frac{1}{(2\pi)^{R/2}} \frac{1}{|\Sigma_j|^{1/2}} exp\left\{ -\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_j) \right\} \right)^{z_{n,j}} \tag{2.4}$$

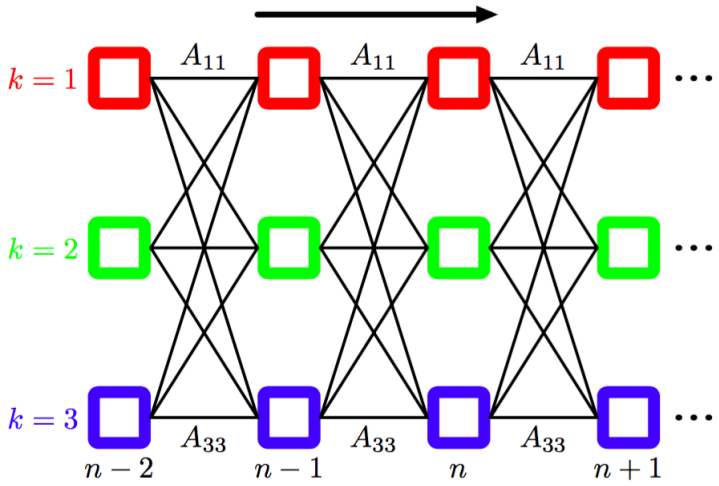
pokud prvky $\boldsymbol{x}_n$ jsou spojité.

Nyní se zaměříme na homogenní modely, pro které všechna podmíněná rozdělení řídící 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

$$\mathbb{P}(\boldsymbol{X}, \boldsymbol{Z}|\widetilde{\boldsymbol{\Theta}}) = \mathbb{P}(z_1|\pi)\prod_{n=2}^{N}\mathbb{P}(z_n|z_{n-1}, \mathbb{A})\prod_{m=1}^{N}\mathbb{P}(\boldsymbol{x}_m|z_m, \Theta), \tag{2.5}$$

kde $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$, $\boldsymbol{Z} = (z_1, ..., z_N)$ a $\widetilde{\boldsymbol{\Theta}} = (\pi, \mathbb{A}, \Theta)$ jsou parametry řídícího modelu.



Obr. 2.2: Stavový diagram Obr. 2.1 rozvinutý s časem. Každý sloupec diagramu odpovídá jedné skryté proměnné z_n. [1]

2.1.1 Viterbiho algoritmus

Tento algoritmus navrhl Andrew Viterbi, již v roce 1967, 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.

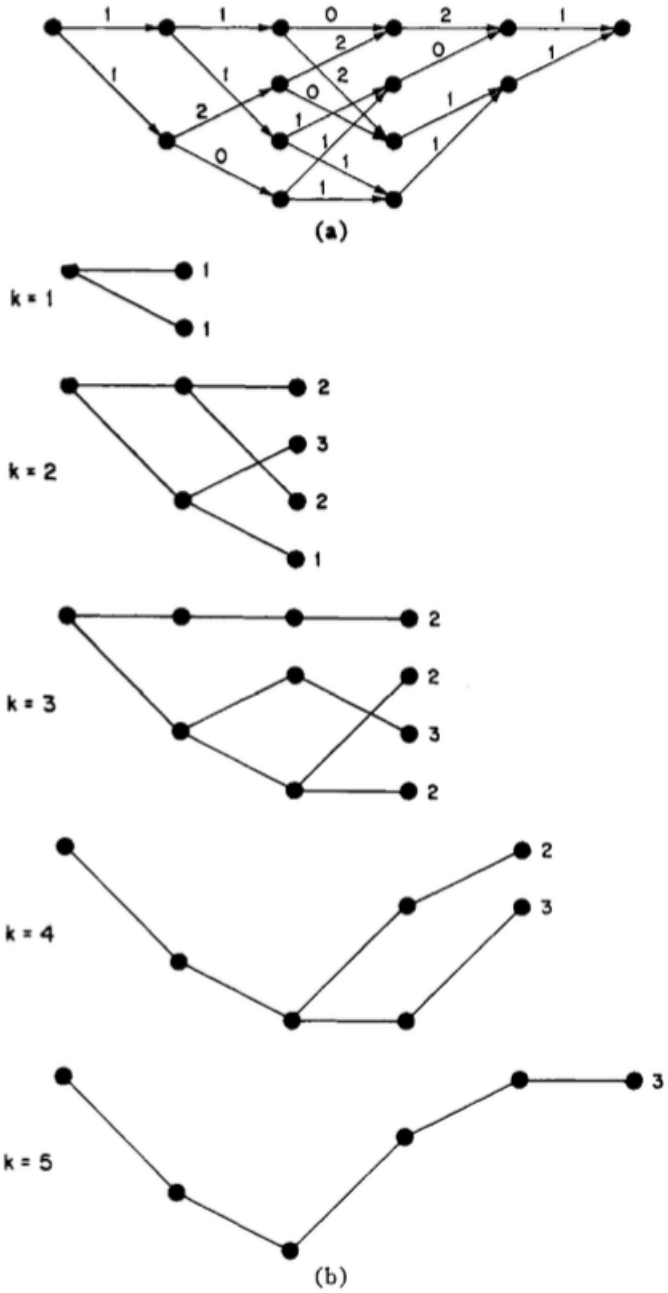
V nejobecnější podobě se na Viterbiho alogritmus můžeme dívat jako na řešení problému maximálního aposteriorního pravděpodobnostního odhadu posloupnosti skrytých stavů konečného diskrétního Markova procesu. Tento problém je formálně identický s problémem hledání nejkratší cesty grafem. Posloupnost pozorování $\boldsymbol{X}$ každé cesty může byt určena jako délka úměrná $-\ln\mathbb{P}(\boldsymbol{Z}, \boldsymbol{X})$, kde $\boldsymbol{Z}$ je seqvence stavů spojená s příslušnou cestou. Tento poznatek nám dovoluje řešit problém hledání posloupnosti stavů, pro které je $\mathbb{P}(\boldsymbol{Z}, \boldsymbol{X}) = \mathbb{P}(\boldsymbol{Z}|\boldsymbol{X})\mathbb{P}(\boldsymbol{X})$ maximální, jako problém hledání cesty jejíž délka $-\ln\mathbb{P}(\boldsymbol{Z}, \boldsymbol{X})$ je minimální. Poněvadž $\ln\mathbb{P}(\boldsymbol{Z}, \boldsymbol{X})$ je monotonní funkcí $\mathbb{P}(\boldsymbol{Z}, \boldsymbol{X})$ a každá cesta odpovídá právě jedné posloupnosti stavů. Následně díky platnosti Markovy vlastnosti můžeme přepsat $\mathbb{P}(\boldsymbol{Z}, \boldsymbol{X})$ jako

$$\mathbb{P}(\boldsymbol{Z}, \boldsymbol{X}) = \mathbb{P}(\boldsymbol{X}|\boldsymbol{Z})\mathbb{P}(\boldsymbol{Z}) = \mathbb{P}(z_1)\left[\prod_{n=2}^{N}\mathbb{P}(z_n|z_{n-1})\right]\prod_{n=1}^{N}\mathbb{P}(\boldsymbol{x}_n|z_n). \tag{2.6}$$

Po zlogaritmování (2.6) můžeme vidět, že celková délka cesty odpovídající libovolnému $\boldsymbol{Z}$ je

$$-\ln \mathbb{P}(\boldsymbol{Z}, \boldsymbol{X}) = \left[\sum_{n=2}^{N} -\ln \mathbb{P}(z_n|z_{n-1}) - \ln \mathbb{P}(\boldsymbol{x}_n|z_n)\right] - \ln \mathbb{P}(z_1) - \ln \mathbb{P}(\boldsymbol{x}_1|z_1), \tag{2.7}$$

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



Obr. 2.3: (a) Mřížový diagram s délkami přechodů. M = 4, K = 5. (b) Rekurzivní hledání nejkratší cesty pomocí Viterbiho algoritmu.

Ve chvíli, kdy jsme našli nejpravděpodobnější cestu, a tím pádem i združené rozdělení $\mathbb{P}(\boldsymbol{Z}, \boldsymbol{X})$, 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 řetězce, 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 řetězce.

2.2 K-means Clustering

K-means clustering je typ strojového učení bez učitele, 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 $(\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ náhodné R rozměrné veličiny $\boldsymbol{X}$ a chceme je rozdělit do nějakých 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 vektory $\boldsymbol{\phi}_j$, kde $j \in 1, .., K$. Tyto vektory představují středy našich shluků a naším cílem se nyní stává nalezení množiny $\{\boldsymbol{\phi}_j\}$, tak aby bylo splněno, že součet čtverců vzdáleností každého bodu shluku k nejbližšímu vektoru $\boldsymbol{\phi}_j$ je minimální. Jinými slovy, potřebujeme minimalizovat objektovou funkci ρ definovanou jako

$$\rho = \sum_{n=1}^{N} \sum_{j=1}^{K} b_{n,j} \, \| \boldsymbol{x}_n - \boldsymbol{\phi}_j \|^2, \tag{2.8}$$

kde proměnné $b_{n,j} \in \{0, 1\}$ příslušící každému bodu $\boldsymbol{x}_n$ 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. Nejprve vybereme, nejlépe náhodně, počáteční vektory $\boldsymbol{\phi}_j$. V první fázy bereme $\boldsymbol{\phi}_j$ 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

$$b_{n,j} = \begin{cases} 1 & \text{pokud } argmin_m \|\boldsymbol{x}_n - \boldsymbol{\phi}_m\|^2 = j \\ 0 & \text{jinde.} \end{cases} \tag{2.9}$$

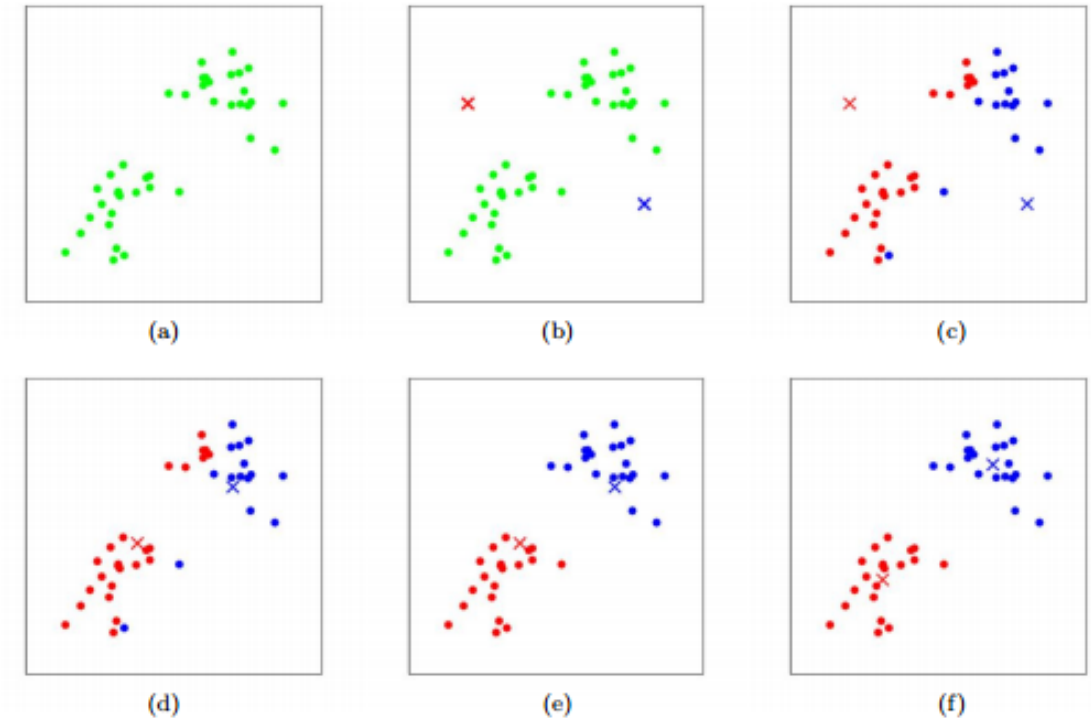
V druhé fázi je naopak $b_{n,j}$ pevné a minimalizujeme s ohledem na $\boldsymbol{\phi}_j$. Proto zderivujeme funkci ρ podle $\boldsymbol{\phi}_j$ a tuto derivací položíme rovnu 0, tzn.

$$2 \sum_{n=1}^{N} b_{n,j}(\boldsymbol{x}_n - \boldsymbol{\phi}_j) = 0. \tag{2.10}$$

Načež po osamostatnění $\boldsymbol{\phi}_j$. získáváme konečný tvar

$$\boldsymbol{\phi}_j = \frac{\sum_{n=1}^{N} b_{n,j} \boldsymbol{x}_n}{\sum_{n=1}^{N} b_{n,j}}, \tag{2.11}$$

kde jmenovatel představuje celkový počet bodů patřících do j-tého shluku. Vzorec (2.11) lze ovšem interpretovat také jako střední hodnotu (anglicky: mean) všech bodů příslušících do shluku j, odtud plyne i název "K-means".



Obr. 2.4: 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*) − (*d*) ilustrují úvodní dvě iterace algoritmu. [10]

Kapitola 3

Postup práce a výsledky

3.1 Rysy (Features)

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

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

Do rysů můžeme zahrnout například věk osoby, pohlaví, jméno, bydlení (byt, dům, ...), rodinný příjem, vzdělání a počet dětí. Je zřejmě, že většina těchto rysů nám může při předvídání 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 rysů k demonstračnímu příkladu

3.1.1 První a druhá derivace

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

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

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).

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

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

Dalším použitým rysem 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 velikým šumem. Proto není od věci pokusit se tento signál nějak vyhladit. Za tímto učelem byl mezi rysy 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.

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 hodotu f_i lineární kombinací g_i nebo sama sebe a určitým počtem nejbližších sousedů,

$$g_i = \sum_{n=-n_L}^{n_R} c_n f_{i+n} = (\boldsymbol{c} \otimes \boldsymbol{f})_j, \tag{3.4}$$

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

Hlavní myšlenou S-G filtru je nalezení koeficientů c_n tak, aby se zachovávaly momenty vyšších řádů a 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 vešeré prokládání vypočítat dopředu a pomocí binárního vektoru lze pak vše dopočítat linerání kombinací.

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

$$A_{ij} = i^j \quad i = -n_L, ..., n_R \quad a \quad j = 0, ..., M \tag{3.5}$$

koeficienty polynomu $\boldsymbol{a}$ lze pak vypočítat jako

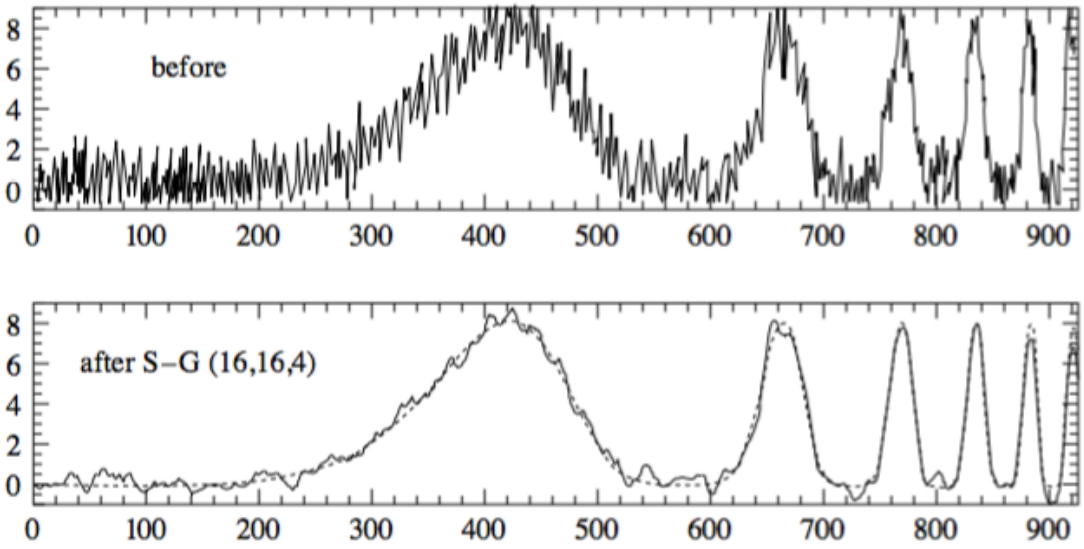
$$\boldsymbol{a} = (\mathbb{A}^T \cdot \mathbb{A})^{-1} \cdot (\mathbb{A}^T \cdot \boldsymbol{f}). \tag{3.6}$$

Nakonec vypočítáme i koeficienty $\boldsymbol{c}$ pomocí

$$\boldsymbol{c} = (\mathbb{A}^T \cdot \mathbb{A})^{-1} \cdot \mathbb{A}^T \tag{3.7}$$

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.



Obr. 3.1: Savitzky-Golay filtr. Na horním obrázku jsou syntetická data s bílým gaussovským šumem. Na spodním je již vyhlazený set získaný aplikací S-G filtru s parametry $n_L = 16, n_R = 16$ a stupeň polynomu $M = 4$. [7]

3.1.3 Klouzavý průměr

Dalším vybraným rysem je klouzavý průměr. Klouzavý průměr je diskrétní lineární filtr s konečnou dobou odezvy, který slouží k vyhlazení signálu. Nadále ho budeme značit jako $\widetilde{X}_t$. Nechť máme vektor naměřených hodnot $\boldsymbol{X} = (x_1, x_2, ..., x_n)$, pak definuji klouzavý průměr pro $t \in 1, 2, ..., n$ jako

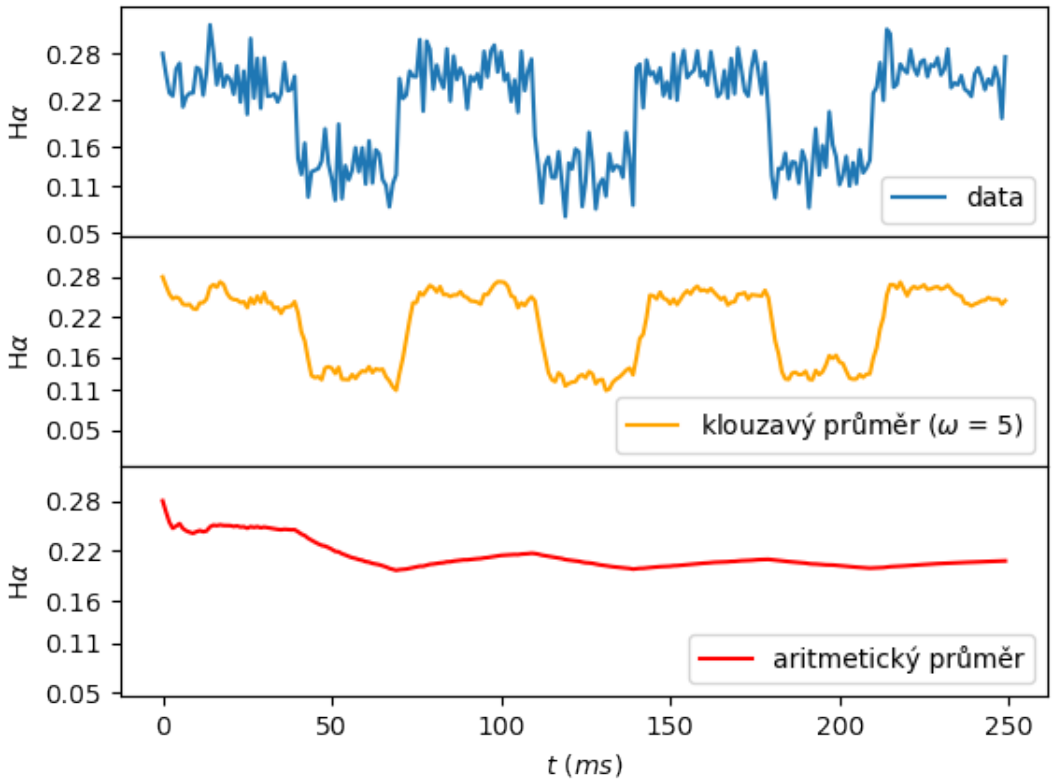
$$\widetilde{X}_t = \frac{1}{\tilde{\omega}} \sum_{k=t-\tilde{\omega}}^{t} x_k, \tag{3.8}$$

kde $\tilde{\omega} = min\{\omega, t\}$, přičemž ω je délka úseku. Pak vektor $\widetilde{\boldsymbol{X}} = (\widetilde{X}_1, \widetilde{X}_2, .., \widetilde{X}_n)$ je rys vektoru $\boldsymbol{X}$.

Ve skutečnosti se jedná o standartní aritmetický průměr (3.9), jež je aplikovaný pouze na úsek dat konečné délky.

$$\overline{X}_n = \frac{1}{n} \sum_{k=1}^{n} x_k, \tag{3.9}$$

Mezi rysy 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 rysu. Z čehož plyne, že takovýto rys by jen zkresloval a znepřesňoval výsledek, viz Obr. 3.2.



Obr. 3.2: Rozdíl mezi klouzavým a aritmerickým 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 rysem je tedy exponenciální klouzavý průměr S_t.

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

$$S_t = \frac{1}{\tilde{\omega}} \sum_{k=t-\tilde{\omega}}^{t} g_{t-k} \cdot x_k, \tag{3.10}$$

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 $\boldsymbol{S} = (S_1, S_2, .., S_n)$ je rys vektoru $\boldsymbol{X}$. 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

$$Var(X) = E[(X - E[X])^2] \tag{3.11}$$

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

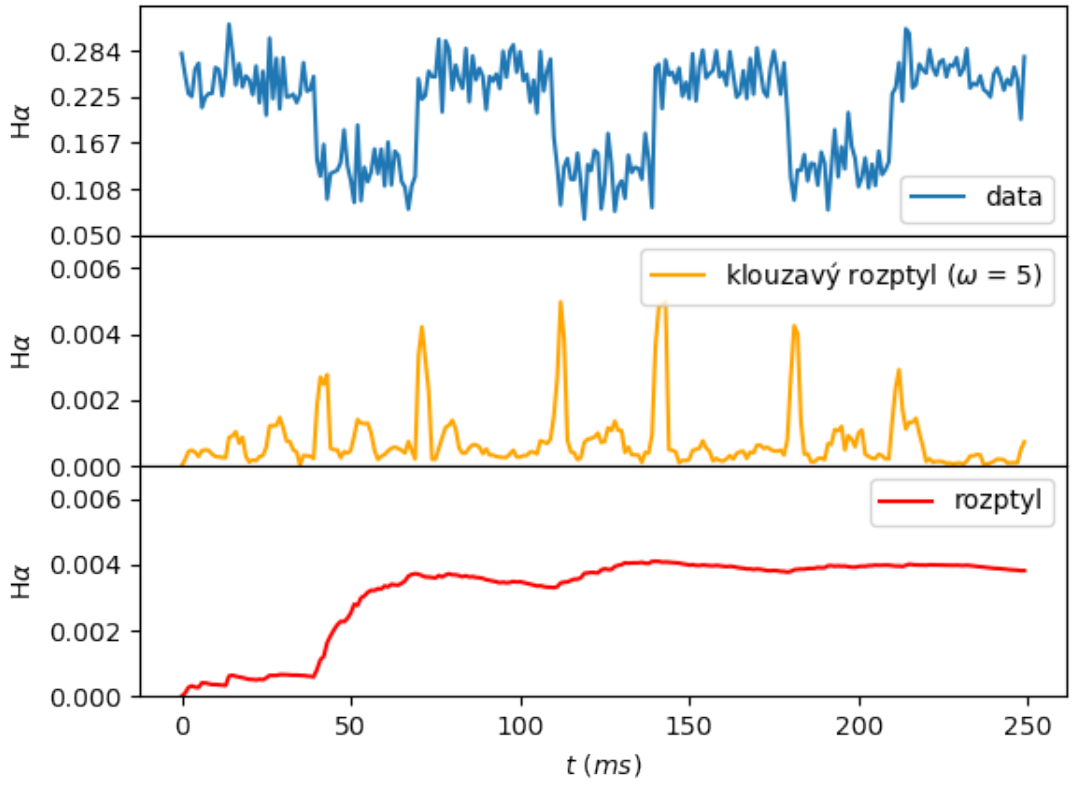
$$Var(X) = \frac{1}{n}\sum_{k=1}^{n}(x_k - \overline{X}_n)^2. \tag{3.12}$$

Bohužel, rozptyl nemůžeme jako rys 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 $\boldsymbol{X} = (x_1, x_2, ..., x_n)$ je vektor naměřených hodnot, pak definuji úsekový rozptyl pro $t \in 1, 2, ..., n$ jako

$$D_m = \frac{1}{\tilde{\omega}}\sum_{k=t-\tilde{\omega}}^{t}(x_k - \widetilde{X}_t)^2, \tag{3.13}$$

kde $\tilde{\omega} = min\{\omega, t\}$, w je opět délka úseku a $\widetilde{X}_t$ je klouzavý průměr (3.8). Pak vektor $\boldsymbol{D} = (D_1, D_2, ..., D_n)$ je rys vektoru $\boldsymbol{X}$.



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

3.2 Způsoby vyhodnocení výsledků

3.2.1 Přesnost

3.2.2 Správnost

3.2.3 Recall

3.2.4 F míra

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] P. Harrington, Machine Learning in Action. Minning Publications Co. Greenwich, 2012.

[7] 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.

[8] R.J. Goldston, P.H. Rutherford. Introduction to Plasma Physics. Taylor and Francis. ISBN 978-0-7503-0183-1, 1995.

[9] http://www.ipp.cas.cz/vedecka_struktura_ufp/tokamak/tokamak_compass/index.html

[10] http://stanford.edu/ cpiech/cs221/img/kmeansViz.png

[11] https://www.researchgate.net/profile/Janos_Egert/publication/283617947/figure/fig1/AS:314833057665024@14
of-the-COMPASS-tokamak-left-and-its-vacuum-chamber-with-diagnostic-ports-right.png