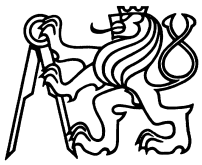
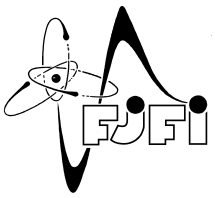


České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská



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: V této bakalářské práci se zabýváme metodami strojového učení, které využíváme ke klasifikaci stavů plazmatu v tokamaku COMPASS. Nejdříve jsme vysvětlili signál H_α, pojem plazma a popsali stavy, ve kterých se může vyskytovat. V teoretické části jsme představili skrytý Markovův model a shlukovací algoritmus K-means. V další kapitole jsou prezentovány příznaky k oběma metodám a navrženy experimenty. První experiment využívá k trénování a testování syntetická data, zatímco při druhém experimentu jsou použity skutečná data z tokamaku. V poslední kapitole jsou obě experimenty vyhodnoceny pomocí zadefinovaných metrik. Výsledkem práce je nalezení nejlepší metody pro klasifikaci stavů plazmatu, která je schopna pracovat v reálném čase.

Klíčová slova: expectation-maximization algoritmus, K-means, plazma, skrytý Markovův model, strojové učení, tokamak COMPASS, Viterbiho algoritmus

Title:

Data mining on the COMPASS tokamak experiments

Author: Matěj Zorek

Abstract: This thesis deals with methods of machine learning which will be applied for classification of plasma states in the COMPASS tokamak. Firstly, we explained H_α lines, a term plasma. Then we described states in which plasma can occur. Further we introduce hidden Markov model and K-means clustering in theoretical part. In next chapter, we presented features for both methods and proposed some experiments. First experiment uses synthetic data for training and testing, but second one applies real data from tokamak. In final chapter results of experiments are evaluated by defined metrics. The end results of this work is the best method for classifying states of plasma which is able to work in real time.

Key words: COMPASS tokamak, expectation-maximization algorithm, hidden Markov model, K-means clustering, machine learning, plasma, Viterbi algorithm by commas

Obsah

Úvod

V dnešní moderní společnosti funguje téměř vše na elektřinu a nároky stále rostou. Řešením by mohlo být ovládnutí termojaderné fúzní reakce. Abychom tento potenciál mohly naplno využít, potřebujeme tuto reakci udržet dlouhodoběji. Jedním z nepokročilejších zařízeních sloužících k jejímu výzkumu jsou tokamaky. Znalost okamžitého režimu udržení 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 představen diskrétní Markovův model, který poté rozšíříme do skrytý Markovův model. Dále je vysvětlen Expectation–maximization algoritmus sloužící k výpočtu parametrů tohoto modelu a Viterbiho algoritmus používaný k nalezení posloupnosti nejpravděpodobnějších stavů. Na konci této kapitoly prezentována shlukovací metoda K-means.

Ve třetí kapitole jsou postupně popsány příznaky používané při aplikaci obou metod. Následuje postupu práce, ve kterém jsou vysvětleny oba provedené experimenty. První z nich byl proveden na syntetických datech a při druhém jsme již použili skutečná data z tokamaku COPMASS.

V poslední kapitoly je uveden seznam metrik, sloužících k vyhodnocení úspěšnosti výsledků a kvality metod. Díle je zde popsána technika křížové validace. Kapitola je zakončena výsledky z obou experimentů.

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. V podstatě se jedná o shluk ionizovaných částic a elektronů s unikátními vlastnostmi. Výměna nábojů mezi ionty a elektrony zapříčiňuje vznik elektrického pole a proudění nabitých částic vyvolává magnetické pole [9]. Na rozdíl od zbylých tří skupenství se volně, za normálních podmínek, na zemském povrchu nevyskytuje. Paradoxně však 99% veškeré pozorovatelné 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 plazmatu. Poté se během srážek těchto elektronů s atomy uvolňují další elektrony a tento kaskádový proces postupuje, dokud plyn nedosáhne požadovaného stupně ionizace (převzato z [9]).

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.

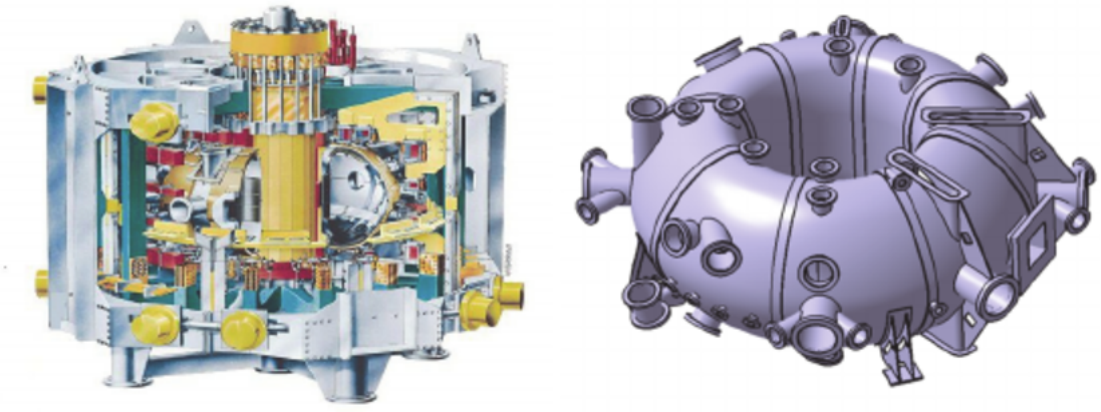
1.2 Tokamak COMPASS

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 fůzi 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 vychýlení 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 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 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. [19]

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 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 Wagnerem 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 nímž dochází na okraji plazmy. ELMy se navíc mohou vyskytovat pouze během H-módu.

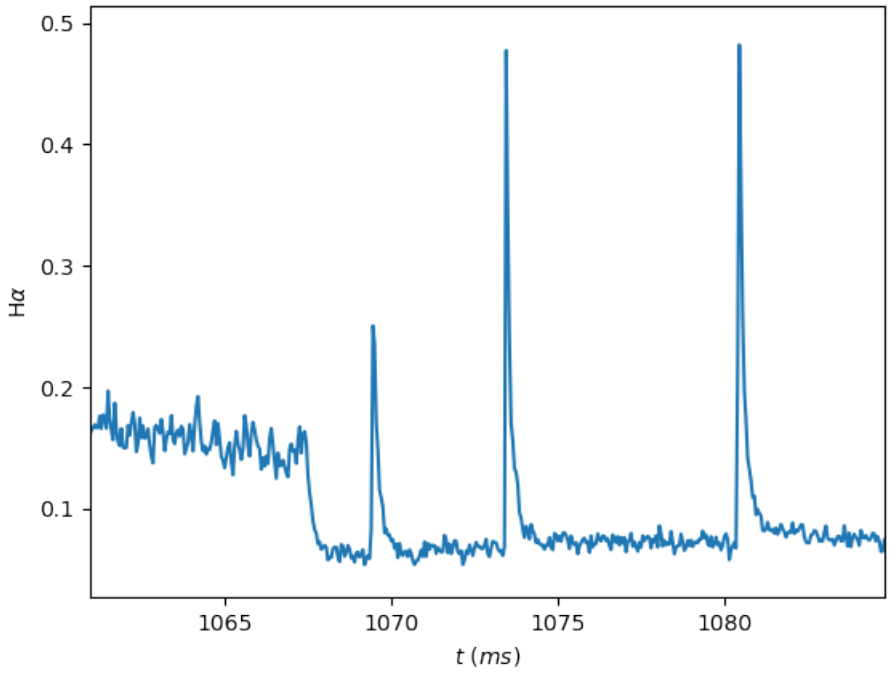
Posledním stavem je disrupce. Disrupce představuje fyzikální jev, během něhož dochází k přetržení nebo náhlé zrátě udžení plazmatu.

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 a metody strojového učení.

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]).

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 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 detektrorem. (číslo výstřelu: 15364)

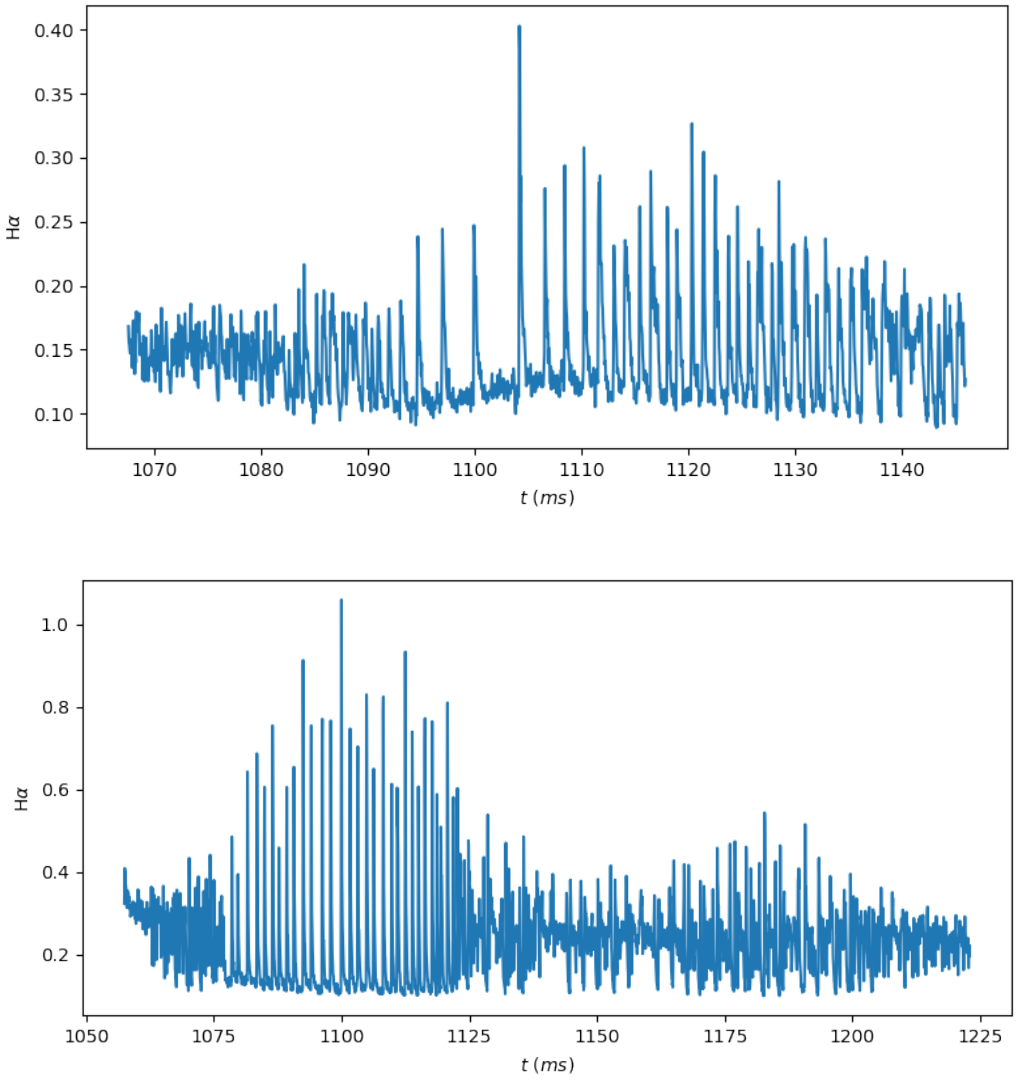
1.4 Strojové učení

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

V této práce 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_α častější případy. (horní obrázek č.v.: 7880, spodní obrázek č.v.: 13484)

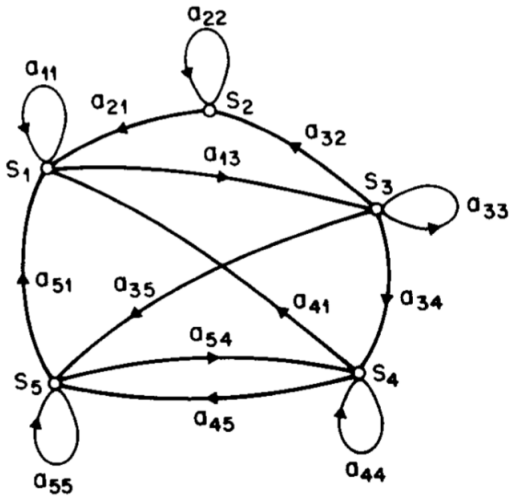
Kapitola 2

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 Expectation–maximization algoritmus a Viterbiho algoritmus, které slouží k výpočtům již zmíněného modelu. 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 diskrétních stavů. Tuto množinu budeme značit jako $\mathbb{S} = \{S_1, S_2, S_3, ..., S_N\}$.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: Markovuv řetězec s 5 stavy a přechody mezi nimi. [3]

Nyní označme časy t odpovídající změnám stavů jako $t = 1, 2, 3, ...$ a stav v čase t označme jako q_t. Pro pravděpodobnostní popis takovéhoto systému je třeba znát současný stav stejně tak jako předchozí stavy. Pokud budeme uvažovat speciální případ jímž je Markovův řetězec prvního řádů, 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 $S_1, S_2, S_3, ..., S_i, S_j \in \mathbb{S}$ platí

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

(převzato z [4]).

Pokud je navíc tento systém nezávislý na čase t platí

$$a_{i_j} = \mathbb{P}(q_{t+1} = S_J | q_t = S_i), \qquad 1 \leq i, j \leq N, \tag{2.2}$$

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

$$\sum_{J}^{N} a_{i,j} = 1. \tag{2.3}$$

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 diskrétního Markova modelu. Tyto modely se bohužel liší v jedné podstatné věci. Zatímco u standartního diskrétního 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 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 směsové distribution (viz kapitola 9 v [1]) s hustotou pravděpodobnosti danou $\mathbb{P}(\boldsymbol{x}|\boldsymbol{z})$. Proto ho můžeme interpretovat jako rozšíření směsového modelu, kde výběr složky směsi, pro každé pozorování, není nezávislý, ale závisí na volbě složek z předchozího pozorování.

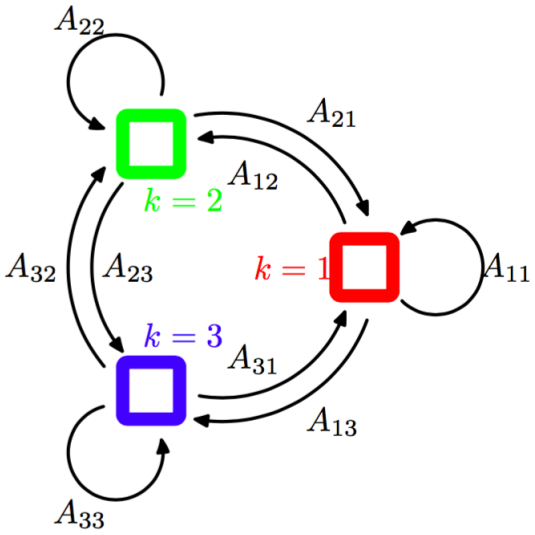
Jako v případě standartního Mixture modelu, skryté proměnné jsou diskrétní multinomické proměnné $\boldsymbol{z}_n$ popisující složku směsi, jež je zodpovědná za generování příslušného pozorování $\boldsymbol{x}_n$. Od této chvíle budeme předpokládat, že skrytý proces je Markovův, tzn. splňuje Markovu vlastnost. 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 $\mathbb{P}(z_n|z_{n-1})$. Pak zavedeme matici přechodů $\mathbb{A}$ danou předpisem

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

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

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



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

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

a tedy

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

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) $\mathbb{P}(\boldsymbol{x}_n|\boldsymbol{z}_n, \Theta)$, kde Θ představuje soubor parametrů řídícího rozdělení. Pro pozorované proměnné $\boldsymbol{x}_n$ se rozdělení $\mathbb{P}(\boldsymbol{x}_n|\boldsymbol{z}_n, \Theta)$ skládá z K-dimenzionálního vektoru odpovídajícího K potenciálním stavům $\boldsymbol{z}_n$. Emisní pravděpodobnosti můžeme pak zapsat jako

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

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

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

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

$$\boldsymbol{\mu} = \mathbb{E}\boldsymbol{X} = (\mathbb{E}X_1, \mathbb{E}X_2, ..) \tag{2.10}$$

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

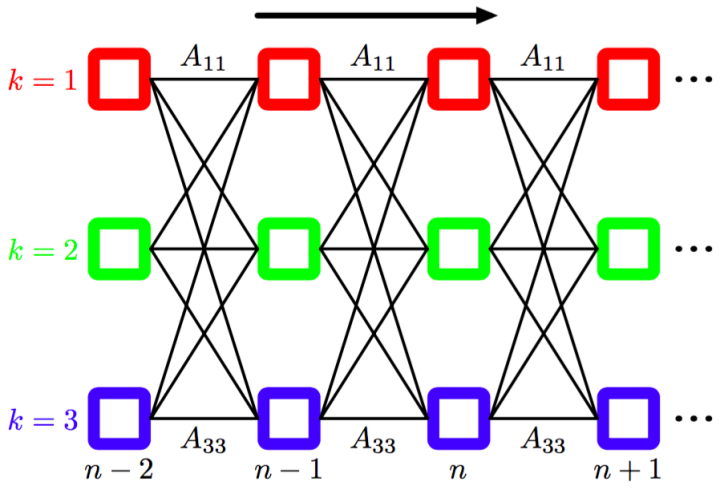
$$\Sigma = \Big(Cov(X_i, X_j) \Big)_{i,j} = \Big(\mathbb{E}[(X_i - \mathbb{E}X_i) \cdot (X_j - \mathbb{E}X_j)] \Big)_{i,j}. \tag{2.11}$$

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

$$\mathbb{P}(\boldsymbol{X}, \boldsymbol{Z}|\widetilde{\boldsymbol{\Theta}}) = \mathbb{P}(z_1|\pi) \prod_{n=2}^{N} \mathbb{P}(z_n|z_{n-1}, \mathbb{A}) \prod_{m=1}^{N} \mathbb{P}(\boldsymbol{x}_m|z_m, \Theta), \tag{2.12}$$

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

2.2.1 Expectation–maximization algoritmus

2.2.2 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 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

$$\mathbb{P}(\boldsymbol{Z}, \boldsymbol{X}) = \mathbb{P}(\boldsymbol{Z}|\boldsymbol{X})\mathbb{P}(\boldsymbol{X}) \tag{2.13}$$

maximální, jako problém hledání cesty jejíž délka

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

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, pak díky platnosti Markovy vlastnosti můžeme přepsat $\mathbb{P}(\boldsymbol{Z}, \boldsymbol{X})$ jako

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

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

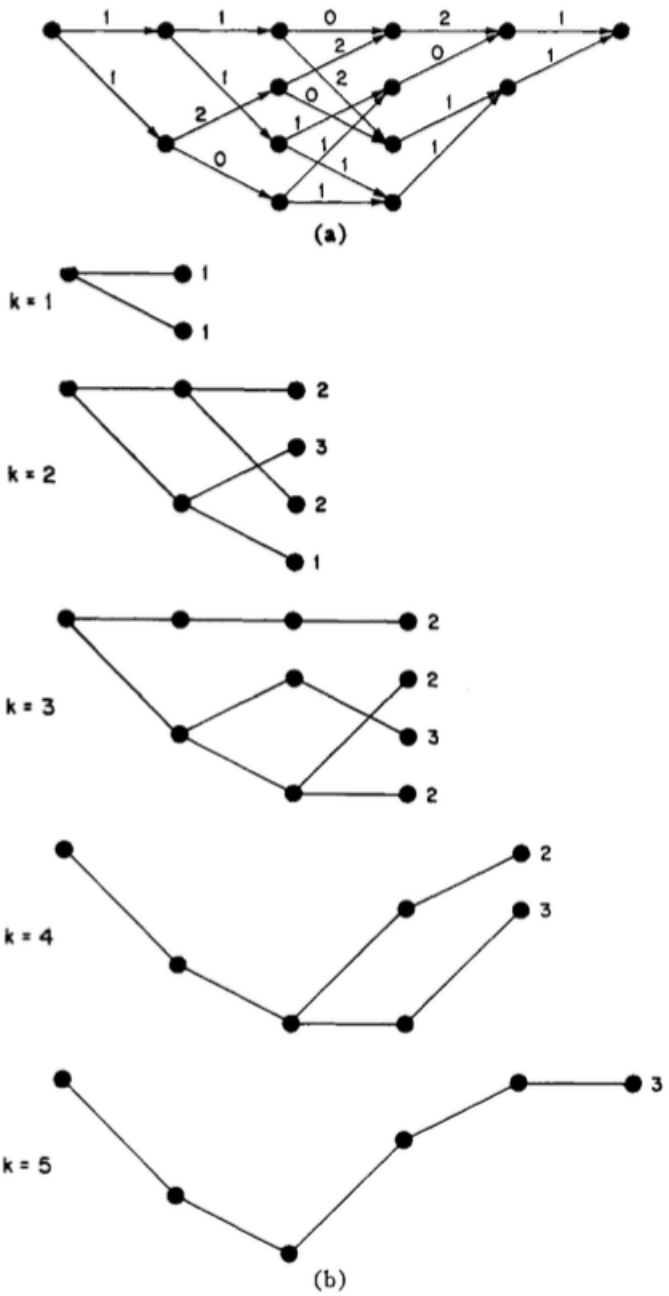
$$-\ln \mathbb{P}(\boldsymbol{Z}, \boldsymbol{X}) = \left[\sum_{n=2}^{N} -\ln \mathbb{P}(z_n|z_{n-1}) - \ln \mathbb{P}(\boldsymbol{x}_n|z_n)\right] - \ln \mathbb{P}(z_1) - \ln \mathbb{P}(\boldsymbol{x}_1|z_1), \tag{2.16}$$

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.

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 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 ($= -\ln \mathbb{P}(z_i|z_{i-1}) - \ln \mathbb{P}(\boldsymbol{x}_i|z_i)$). 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 $(\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ náhodné R rozměrné veličiny $\boldsymbol{X}$ 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 $\boldsymbol{\phi}_j$, kde $j \in 1, .., K$. Tyto centroidy jsou vektory představují 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 $\{\boldsymbol{\phi}_j\}$, tak aby bylo splněno, že součet čtverců vzdáleností každého bodu shluku k nejbližšímu vektoru $\boldsymbol{\phi}_j$ je minimální. Jinými slovy, potřebujeme minimalizovat účelovou funkci ρ definovanou jako

$$\rho = \sum_{n=1}^{N} \sum_{j=1}^{K} b_{n,j} \, \|\boldsymbol{x}_n - \boldsymbol{\phi}_j\|^2, \tag{2.17}$$

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, ve kterém dochází k potupné minimalizaci účelové funkce, dokud nenastane konvergence. 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

$$b_{n,j} = \begin{cases} 1 & \text{pokud } argmin_m \|\boldsymbol{x}_n - \boldsymbol{\phi}_m\|^2 = j \\ 0 & \text{jinde.} \end{cases} \tag{2.18}$$

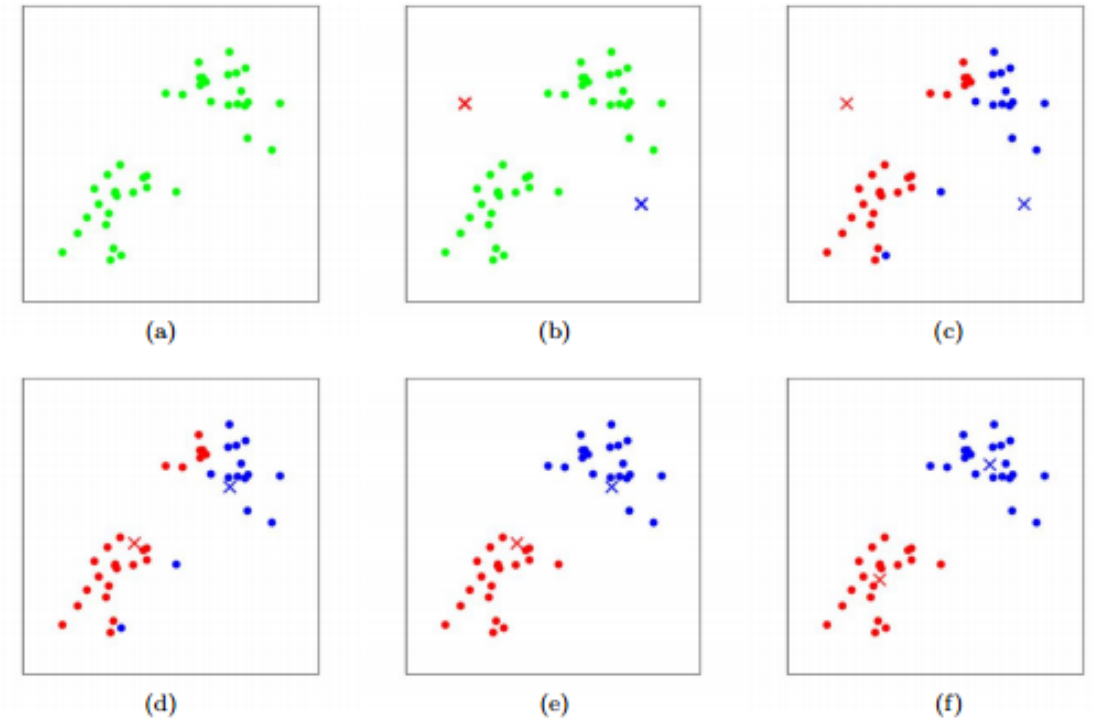
V druhé fázi je naopak $b_{n,j}$ pevné a minimalizujeme s ohledem na $\boldsymbol{\phi}_j$. Proto zderivujeme funkci ρ podle $\boldsymbol{\phi}_j$ a tuto derivací položíme rovnu 0, tzn.

$$2 \sum_{n=1}^{N} b_{n,j}(\boldsymbol{x}_n - \boldsymbol{\phi}_j) = 0. \tag{2.19}$$

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

$$\boldsymbol{\phi}_j = \frac{\sum_{n=1}^{N} b_{n,j} \boldsymbol{x}_n}{\sum_{n=1}^{N} b_{n,j}}, \tag{2.20}$$

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*) − (*d*) ilustrují úvodní dvě iterace algoritmu. [18]

Kapitola 3

Praktická část - experimenty

V této kapitole se budeme zabývat praktickou částí, ve které si nejdříve představíme 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. Ostatní funkce včetně příznaků jsou napsány v modulu Classification.py, kromě metody K-means, která má vlastní stejnojmenný modul.

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 zřejmě, že většina těchto příznaků 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 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 $f(x) : \mathbb{R} \longrightarrow \mathbb{R}$, 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).

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

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

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 velikým šumem. Proto je vhodné pokusit se tento signál nějak vyhladit. Za tímto učelem 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 hodotu f_i hodnotou g_i, jež lineární kombinací určitého počtu nejbližších sousedů bodu f_i,

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

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

Hlavní myšlenou S-G filtru je 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

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

vektro koeficientů polynomu $\boldsymbol{a}$ lze pak vypočítat jako

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

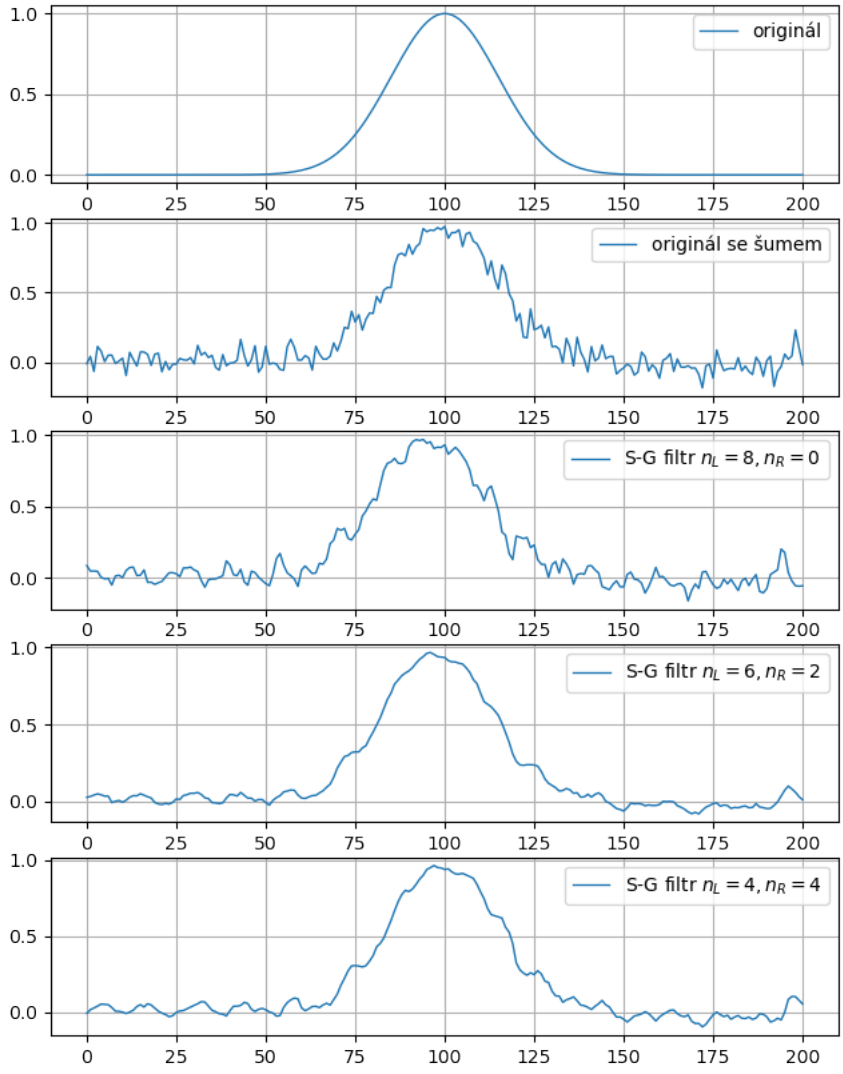
Nakonec vypočítáme i vektro koeficientů $\boldsymbol{c}$ pomocí

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

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. Tato úprava sočívá v záměně derivace funkce f za derivaci polynomu a to díky vlastnosti konvoluce

$$(c * f') = (c' * f). \tag{3.8}$$



Obr. 3.1: Savitzky-Golay filtr aplikovaný na gaussovu křivu 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 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

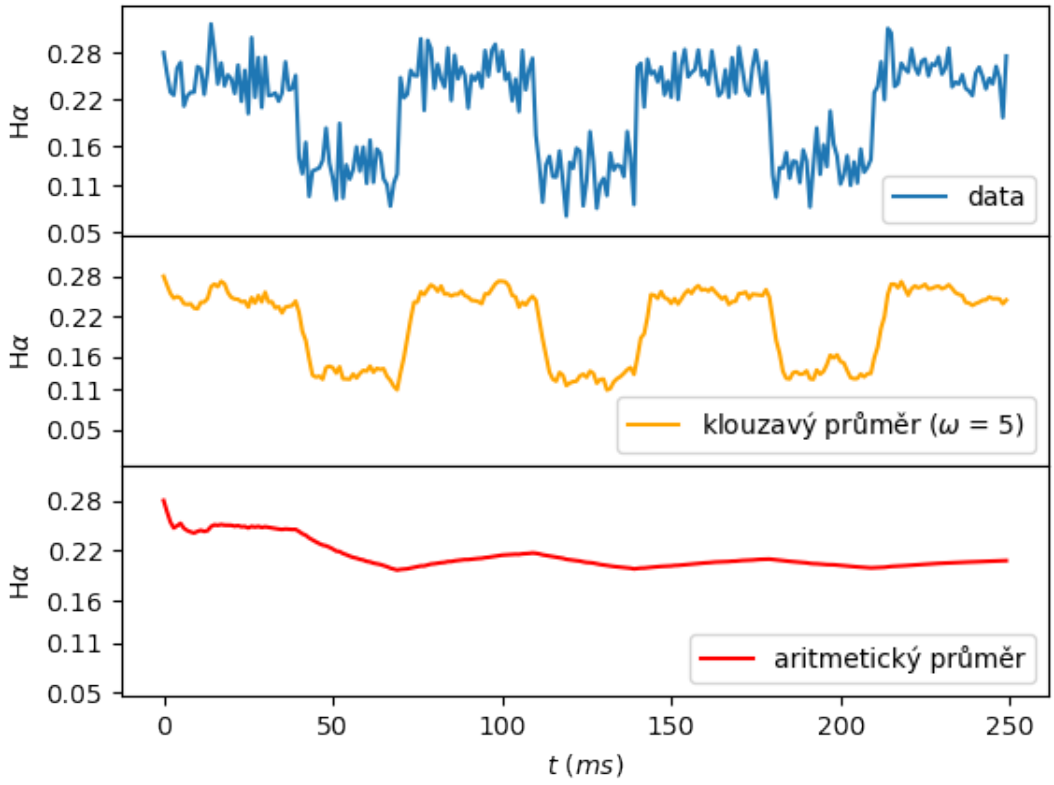
$$\widetilde{X}_t = \frac{1}{\tilde{\omega}} \sum_{k=t-\tilde{\omega}}^{t} x_k, \tag{3.9}$$

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

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

$$\overline{X}_n = \frac{1}{n} \sum_{k=1}^{n} x_k, \tag{3.10}$$

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 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 příznakem 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

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

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 $\boldsymbol{S} = (S_1, S_2, .., S_n)$ je příznak 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

$$Var(X) = E[(X - E[X])^2] \tag{3.12}$$

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

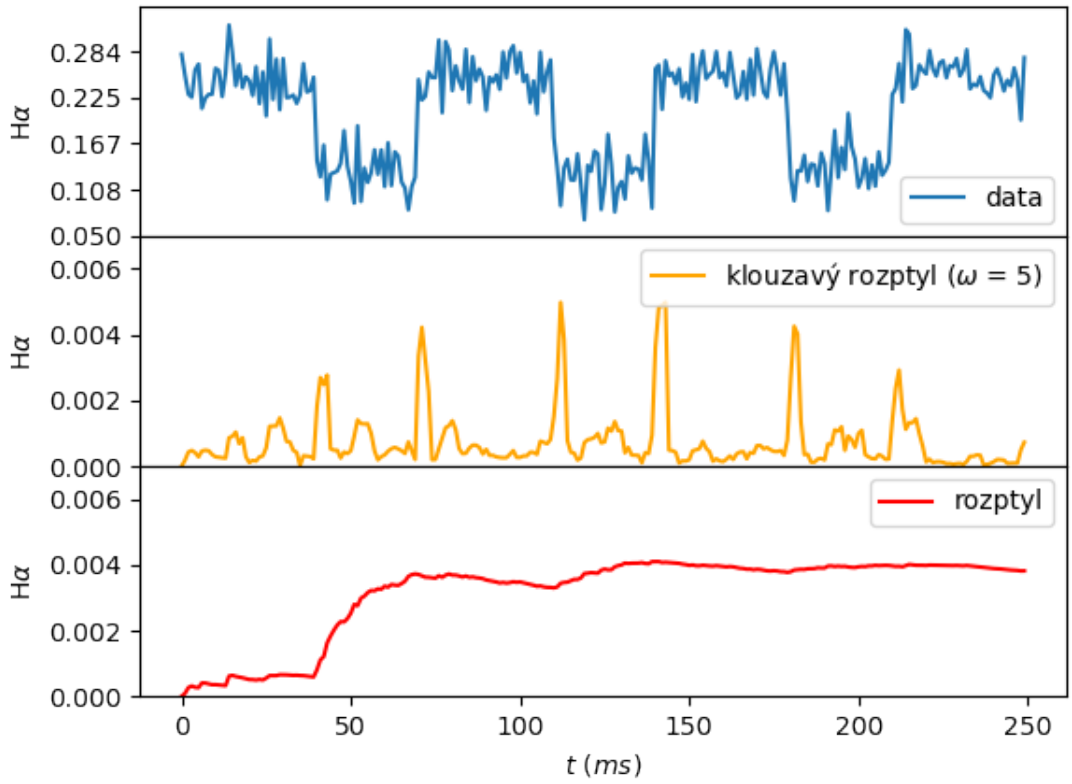
$$Var(X) = \frac{1}{n} \sum_{k=1}^{n} (x_k - \overline{X}_n)^2. \tag{3.13}$$

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

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

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



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

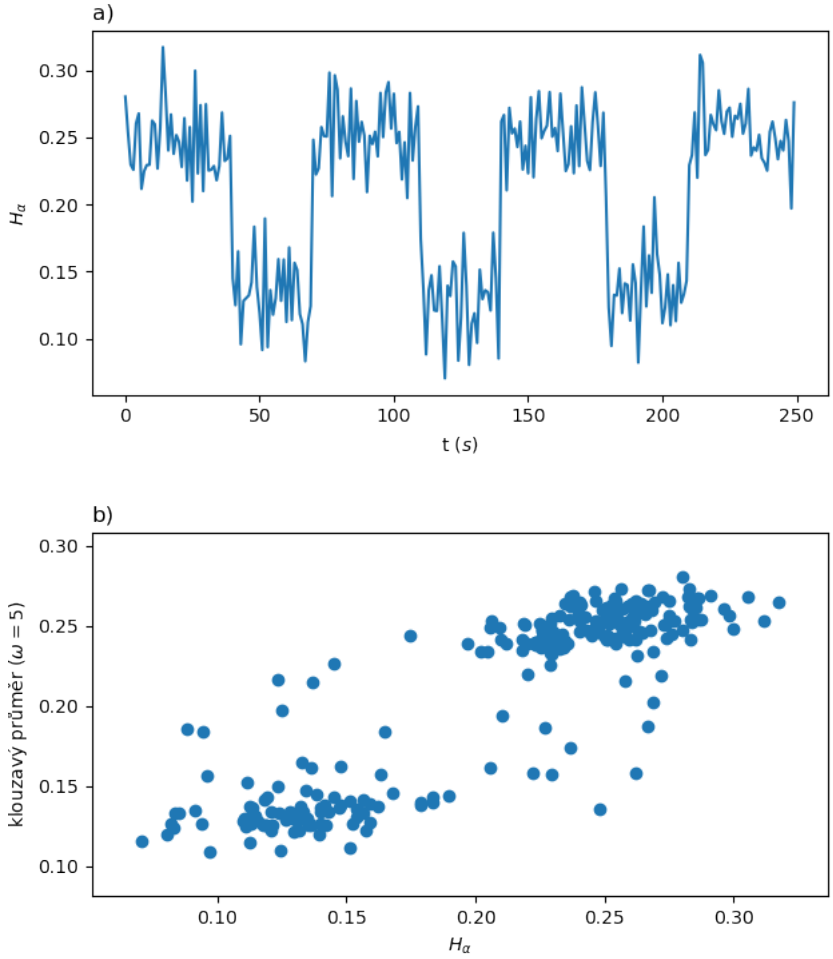
3.2 Experiment na syntetických datech

V této podkapitole se budeme zabývat prvním experimentem, a to porovnáním skrytého Markovova modelu se shlukovou metodou K-means na syntetických datech. Hlavním účelem tohoto experimentu je vyzkoušet obě metody na snadněji rozpoznatelných datech, otestování správného fungování všech funkcí a v neposlední řadě odstranění případných závažných chyb.

Skrytý Markovův model budeme v tuto chvíli používat ve formě učení bez učitele. Využijeme k tomu knihovnu hmmlearn [14]. Při všech našich experimentech budeme předpokládat, že příznaky mají ve všech stavech Gaussovo rozdělení. Díky tomu lze tento model popsat pomocí vzorce (2.9) a navíc můžeme využít předimplementovanou třídu GaussianHMM, která při výpočtu skrytého Markovova modelu využívá právě toho, že jsou emisní pravděpodobnosti Gaussovské. Abychom tento model mohli korektně použít, musíme si uvědomit, co vlastně chceme počítat. Naším úkolem je učit tři stavy plazmatu v tokamaku (H-mód, L-mód, ELM). Tyto tři stavy budou reprezentovány třemi skrytými stavy z_1, z_2 a z_3. Vektory $\boldsymbol{x}_n$ jsou vektory příznaků, kde každá složka vektoru odpovídá hodnotě jednoho příznaku v bodě n.

V případě metody K-means nejsou třeba žádné dodatečné předpoklady kromě jediného, a to, že se data dají rozdělit do shluků. Tato metoda se dá úspěšně použít jen ve chvíli, kdy známe počet hledaných shluků. V našem případě to ovšem není problém, protože budeme hledat přesně tři shluky, jeden pro každý stav. Dalo by se namítat, že tato metoda se nehodí k práci s časovými řadami. Nicméně, na Obr.

3.4 b) je vidět, že když pomineme časovou složku signálu, na osu x vyneseme záření H_α a na osu y například klouzavý průměr, pak vzniknou dva shluky, se kterými si K-means snadno poradí.



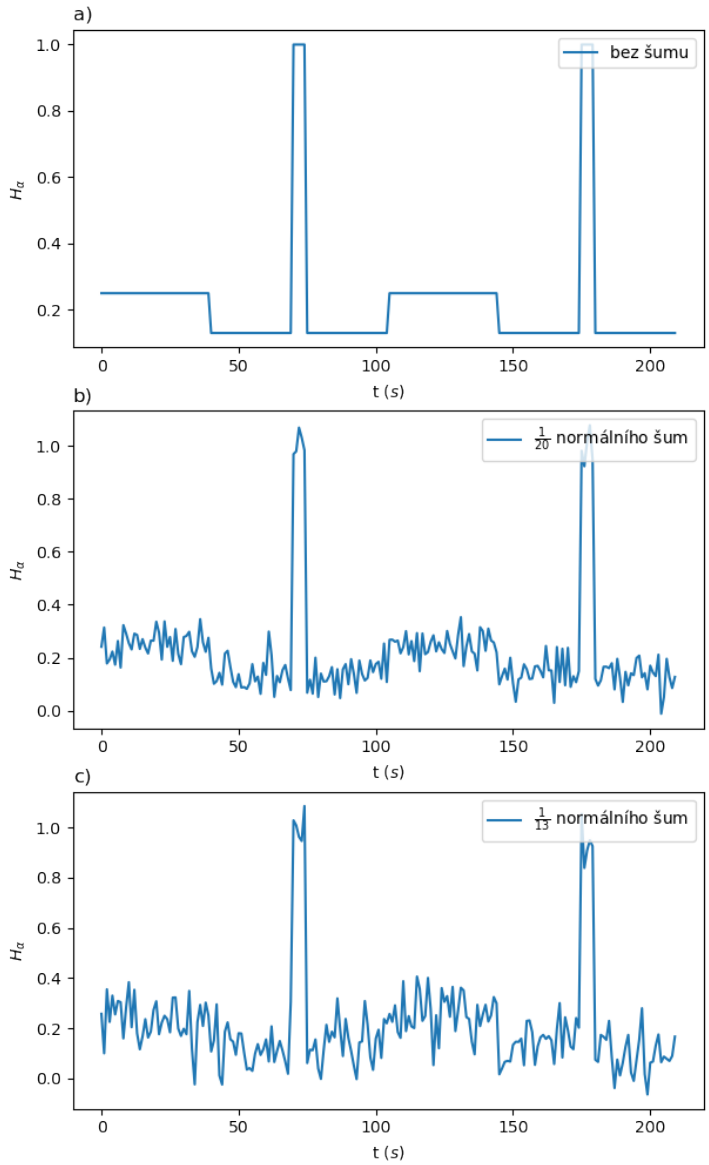
Obr. 3.4: Na obrázku a) je vyobrazen syntetický signál se dvěma stavy a na obrázku b) je tento signál bez časové složky.

Pro tento experiment jsme vytvořili jednoduchý syntetický signál se třemi stavy Obr. 3.5 a), na který aplikujeme Gaussovský bílý šum. Gaussovský bílý šum je statistický šum, který má hustotu pravděpodobnosti odpovídající normálnímu rozdělení $\mathcal{N}(0, 1)$ [15]. Jelikož se velikost toho šumu ukázala pro naše data příliš vysoká, bylo třeba šum trochu zmenšit. Využívat budem tedy $\frac{1}{20}$ a $\frac{1}{13}$ jeho normální velikosti viz Obr. 3.5.

Poté budeme na těchto datech testovat následující příznaky: první derivace, druhá derivace, klouzavý průměr, exponenciální klouzavý průměr a klouzavý rozptyl. Chtěli bychom najít nejlepší kombinaci těchto příznaků spolu s takovou délkou úseku (ω), abychom měli co nevětší přesnost.

Během tohoto experimentu byla objevena jedna velká nevýhoda učení bez učitele. Problém vyvstal ve chvíli, kdy jsme chtěli porovnat výsledky obou metod se skutečným řešením. Metody pracující bez učitele si na počátku každého tréninku potřebují inicializovat skupiny, do nichž budou později rozdělovat data, a poněvadž jsou tyto skupiny voleny náhodně, tak jsou pokaždé očíslovány jinak. Předpokládejme, že ve skutečném řešení jsou stavy číslovány následovně L-mód = 1, H-mód = 0 a ELM = 2. Od výstupu

metody bychom tedy očekávali, že nám bude tyto stavy číslovat stejně, ale to není možné. Jelikož metodě nesdělujeme do jaké skupiny který bod patří, tak šance, že budou tyto stavy očíslovány správně, je 1 ku 6. A to je právě zásadní problém při ověřování výsledků, protože se může stát, že body jsou roztříděny do skupin správně, ale přesnost je nulová. Tento problém byl vyřešen pomocí funkce "srovnej", která vyzkouší všechny permutace, jež mohou nastat $(0, 1, 2; 1, 0, 2; ...)$, a vrátí tu s největší přesností.

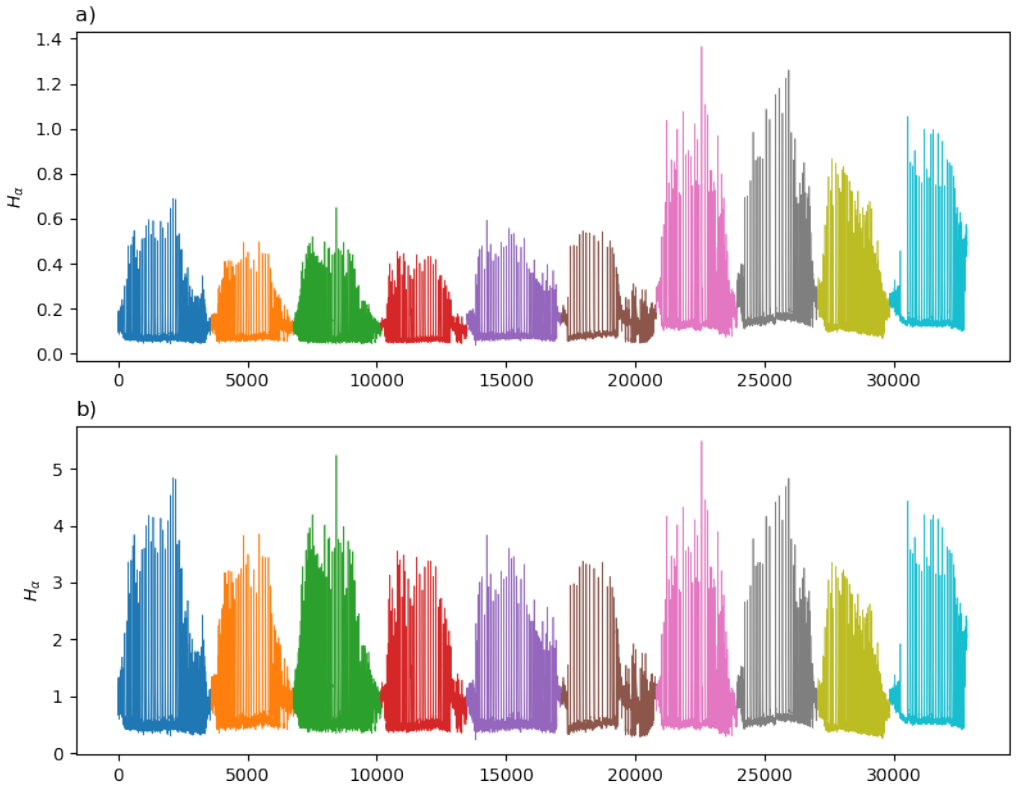


Obr. 3.5: Náhled syntetických signálů použitých při experimentu. Na obrázku a) je základní šablona signálu. Na obrázcích b) a c) je již zašuměný signál s různými velikostmi šumu.

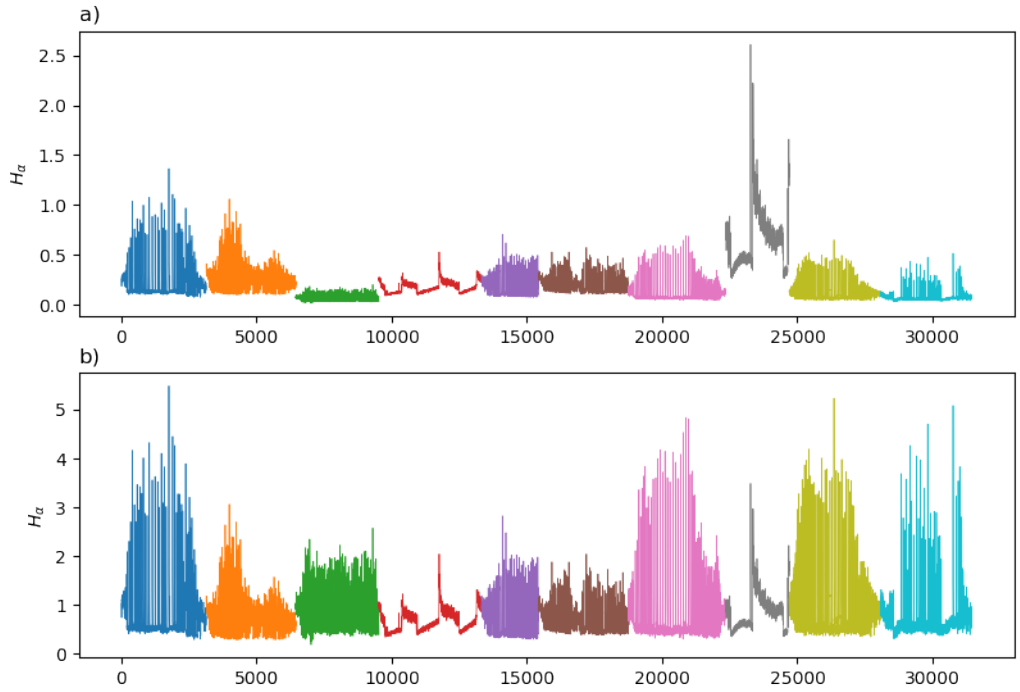
3.3 Experiment na reálných datech

V tomto experimentu se budeme opět zabývat porovnáním našich dvou metod s tím rozdílem, že nyní je budeme trénovat a testovat na reálných datech z tokamaku COMPASS a k výpočtu první a druhé derivace bude použit Savitzky-Golay filtr. Skrytý Markovův model bude nejdříve použit standartně bez učitele a později ho modifikujeme k použití ve formě učení s učitelem. Algoritmus K-means bude použit totožným způsobem jako při předchozím experimentu.

Nejprve je potřeba získat data z databáze. Používat budeme data zaznamenávána fotonásobičem. Tyto data jsou však měřena s příliš vysokou frekvencí a pro simulaci výpočtů v reálném čase je tedy třeba jejich převzorkování. Poněvadž počítač má 50 *ms* na jednotlivé výpočty, bude nám stačit vzít každý stý bod původního signálu. Vybírali jsme pouze data, která byla už někým zkontrolována, abychom se vyhnuli špatně označeným signálům, jež by mohli ovlivnit výsledky. Poté byly vybrány dva datasety skládající se každý z deseti signálů. První dataset je tvořen signály, na kterých jsou hlavně H-módy a ELMy (Obr. 3.6). Druhý dataset obsahuje spíše obecnější data (Obr. 3.7). Poslední důležitý krok, který je třeba s daty udělat, je normalizace, protože naměřená data jsou ovlivněna velikostí napětí na fotonásobiči. K normalizaci používáme velmi jednoduchý proces, při kterém spočítáme průměr několika prvních bodů a tímto průměrem, pak vydělíme celý signál.



Obr. 3.6: Dataset tvořený signály s velkým počtem H-módů a ELMů. Na obrázku a) jsou signály před normalizací a na obrázku b) jsou signály po normalizaci. (Čísla výstřelů: 15304, 15311, 15312, 15318, 15349, 15364, 15546, 15547, 15578, 15617.)



Obr. 3.7: Dataset tvořený obecnými signály. Na obrázku a) jsou signály před normalizací a na obrázku b) jsou signály po normalizaci. Při porovnání a) a b) je dobře patrná důležitost normalizace. (Čísla výstřelů: 15546, 13484, 11282, 5823, 15128, 9735, 15304, 16824, 15312, 11733.)

Na rozdíl od předchozího experimentu použijeme více délek úseků, na kterých bude možno počítat exponenciální klouzavý průměr a klouzavý průměr. Při testech se ukázalo, že tyto průměry počítané na kratších úsecích zlepšují detekci ELMů a naopak průměry delších úseků zase napomáhají hledání H-módů. Přidáním více délek se pokusíme vylepšit schopnost nalezení obou z nich.

3.3.1 Způsob modifikace skrytého Markovova modelu

Z předchozího experimentu víme, že skrytý Markovův model funguje ve formě bez učitele poměrně dobře. Ovšem byla by škoda nevyužít při klasifikaci i naší předběžnou znalost správných výsledků. V knihovně hmmlearn není bohužel naimplementována možnost využití této znalosti přímo, jako je to u standartních metod učení s učitelem, kde při tréninku zadáváte uspořádanou dvojici vektoru příznaků a číslo skupiny, do které tento vektor patří. To však neznamená, že znalost výsledků nelze použít vůbec. Ve třídě GaussianHMM se úvodní inicializace parametrů (tj. střední hodnoty, kovarianční matice, přechodová matice a pravděpodobnosti prvního stavu) provádí náhodně, ale může být také zadána uživatelem. Jelikož víme, do jakého stavu, který vektor příznaků patří, není náročné tyto parametry vypočítat dopředu.

Nejdříve roztřídíme tréninkové vektory příznaků podle stavů, poté spočítáme střední hodnotu každého příznaku v daném stavu. Kovarianční matici vytvoříme z jednotkové matice tak, že za diagonální prvky dosadíme rozptyly jednotlivých příznaků. Poslední parametr, který modelu určíme dopředu, je pravděpodobnost prvního stavu. Jelikož všechna data byla oříznuta tak, aby začínala v L-módu, pak

pravděpodobnost prvního stavu je rovna 1 pro L-mód a pro ostatní stavy je 0. Při tréninku samozřejmě umožňujeme tyto hodnoty měnit a upravovat, protože se jedná pouze o předběžný odhad. Předpočítání parametrů není ovšem poslední možnost, jak lze model ještě vylepšit. Víme navíc totiž, že ELMy mohou nastávat pouze v H-módu, tzn. $\mathbb{P}(\text{ ELM} \mid \text{L-mód}) = 0$. Proto po natrénování modelu, ještě před samotnou klasifikací testovacích dat, upravíme tedy ještě přechodovou matici (2.4) tím, že vynulujeme pravděpodobnost výše zmíněného přechodu.

Díky této modifikaci nyní odpadá i nepříjemná povinnost permutovat výstup modelu podle skutečných výsledků. Protože model bude od této chvíle číslovat stavy podle toho, v jakém pořadí jsme mu je předpočítali.

Kapitola 4

Výsledky

V této kapitole se budeme zabývat výsledky obou našich experimentů. Nejdříve si předsavíme způsoby, jimiž budeme výsledky vyhodnocovat. Poté budou zhodnoceny experimenty představené v předchozí kapitole.

4.1 Způsoby vyhodnocení výsledků

V této podkapitole se budeme věnovat způsobům, jak vyhodnocovat kvalitu námi 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 dašlí 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 výkonnosti daného modelu. Dává nám přehled o chybám, kterých se model dopouští a navíc i druh těchto chyb. Využívá se standartně při učení s učitelem, kde jsou k dispozici označená data.

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

4.1.2 Přesnost

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

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

Přesnost má bohužel jednu velkou nevýhodu, kterou ukážeme na příkladu. Uvažujme proces nabývající 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 seqvence stavů byly samé jedničky (první stav na všech bodech). Pokud bychom spočítali přesnost takovéhoto 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.

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 tří stavů. Diagonální prvky $C_{0,0}$, $C_{1,1}$ a $C_{2,2}$ udávají kolikrát byl konkrétní stav predikován správně. Prvek $C_{i,j}$, kde $i, j \in 0, 1, 2 \wedge i \neq j$, udává, kolikrát byl predikován stav i a ve skutečnosti se jednalo o stav j.

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

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

4.1.4 Recall

Recall udává, jaký podíl všech skutečných stavů, jež 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

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

4.1.5 F míra

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

$$\text{F-míra}_i = \frac{2 \cdot Precision_i \cdot Recall_i}{Precision_i + Recall_i} = \frac{C_{i,i}}{\sum_{j=0}^{2} C_{j,i} + \sum_{j=0}^{2} C_{i,j}} \tag{4.4}$$

4.1.6 Křížová validace

Křížová validace (anglicky: Cross-fold validation) je technika používaná ve statistické analýze, která umožňuje odhadnout, jak přesně bude model fungovat v praxi [16]. Používá se hlavně tehdy, když je cílem predikce nebo klasifikace. Hlavním cílem je otestovat schopnost modelu klasifikovat neznámá data, která nebyla použita při jeho odhadu, a případně upozornit na problém. V našem případě je tímto problémem, kterému chceme předejít, tzv. overfitting. Výsledkem overfittingu je model, který funguje pouze na natrénovaných datech a jakmile ho nasadíme do praxe, stává se nepoužitelným.

My tuto techniku budeme používat následovně. Nejdříve vybereme dataset skládající se z deseti signálů, poté budeme vždy na devíti z nich trénovat a desátý testovat (takzvaně do "kříže"). Nakonec výsledky zprůměrujeme a tím získáme požadované konečné výsledky, které budeme dále vyhodnocovat.

4.2 Výsledky experimentu na syntetických datech

Nyní probereme výsledky našeho prvního experimentu. Na syntetických datech si vedly obě metody velice dobře. Skrytý Markovův model měl pro první velikost šumu nejlepší výsledky pouze s jedním příznakem (samotná data jsou do modelu přidávána vždy, proto je mezi příznaky nepočítáme), a to s exponenciálním klouzavým průměrem, který byl použit na úsek délky 5 (viz Tab. 4.1). Pro data s druhou velikostí šumu, konkrétně pro $\frac{1}{13}$, se k předchozímu příznaku přidala ještě druhá derivace (viz Tab. 4.2). Tato kombinace si ovšem vedla dobře i v předchozím případě, a proto bychom jí mohli prohlásit za nejlepší kombinaci příznaků pro skrytý Markovův model na syntetických datech.

K-means si vedli o něco hůře, ale přesnost 93% pro první šum je velice dobrá a průměrná F-míra je dokonce 95%. Stejně jako u předchozího modelu nám pro nalezení nejlepšího výsledku stačí pouze jeden příznak. V tomto případě se jím stal klouzavý průměr s délkou úseku 5. Nejlepšími příznaky pro větší šum se ukázala kombinace exponenciálního klouzavého průměru a normálního klouzavého průměru s přesností 88% a průměrnou F-mírou 91%.

Důležitým poznatkem plynoucím z tohoto experimentu je, že obě metody lze využít k řešení problému, kterým se tato práce zabývá. A navíc je lze použít i v reálném čase, což byl další z požadavků. Dalším zjištěním je, že při větších šumech se první a druhá derivace stává spíše kontraproduktivní. Důvodem je způsob, kterým obě derivace počítáme. Pokud totiž použijeme centrální diference na signál s velkým šumem, pak výsledkem je signál s ještě větším. Při dalším experimentu proto nahradíme centrální diference Savitzky-Golay filtrem.

index	3	1	19	34	11
Kombinace příznaků	(0, 0, 1, 0, 0)	(0, 0, 0, 1, 0)	(1, 0, 1, 0, 0)	(0, 0, 1, 0, 0)	(0, 1, 1, 0, 0)
Délka úseku	5	5	5	6	5
Accuracy	0.942	0.940	0.933	0.932	0.919
F míra H-mód	0.946	0.945	0.938	0.937	0.924
F míra L-mód	0.929	0.927	0.919	0.918	0.904
F míra ELM	1.000	1.000	1.000	1.000	1.000
F míra průměrná	0.959	0.957	0.953	0.952	0.943
Precision H-mód	0.998	0.998	0.998	0.998	0.998
Precision L-mód	0.870	0.866	0.853	0.851	0.826
Precision ELM	1.000	1.000	1.000	1.000	1.000
Recall H-mód	0.900	0.897	0.885	0.883	0.860
Recall L-mód	0.998	0.998	0.998	0.998	0.998
Recall ELM	1.000	1.000	1.000	1.000	1.000

Tab. 4.1: Výsledky skrytého Markovova modelu aplikovaného na syntetická data ($\frac{1}{20}$ normálního šumu). Nejlepších pět kombinací příznaků a délek úseku seřazených podle průměrné F-míry. Uspořádaná pětice (0, 1, 1, 0, 0) $\approx$ (první derivace, druhá derivace, exponenciální klouzavý průměr, klouzavý průměr, klouzavý rozptyl) udává, které příznaky byly použity (na jejich pozici je 1) a které ne (na jejich pozici je 0). Délka úseku (ω) je při tomto experimentu pro všechny průměry a rozptyl stejná.

index	11	3	27	9	42
Kombinace příznaků	(0, 1, 1, 0, 0)	(0, 0, 1, 0, 0)	(1, 1, 1, 0, 0)	(0, 1, 0, 1, 0)	(0, 1, 1, 0, 0)
Délka úseku	5	5	5	5	6
Accuracy	0.928	0.927	0.927	0.924	0.918
F míra H-mód	0.933	0.932	0.932	0.929	0.923
F míra L-mód	0.913	0.911	0.911	0.909	0.902
F míra ELM	1.000	1.000	1.000	1.000	1.000
F míra průměrná	0.948	0.948	0.948	0.946	0.942
Precision H-mód	0.993	0.991	0.991	0.993	0.992
Precision L-mód	0.847	0.847	0.847	0.841	0.829
Precision ELM	1.000	1.000	1.000	1.000	1.000
Recall H-mód	0.880	0.880	0.880	0.873	0.863
Recall L-mód	0.990	0.988	0.988	0.990	0.990
Recall ELM	1.000	1.000	1.000	1.000	1.000

Tab. 4.2: Výsledky skrytého Markovova modelu aplikovaného na syntetická data ($\frac{1}{13}$ normálního šumu). Nejlepších pět kombinací příznaků a délek úseku seřazených podle průměrné F-míry.

index	1	17	2	9	5
Kombinace příznaků	(0, 0, 0, 1, 0)	(1, 0, 0, 1, 0)	(0, 0, 0, 1, 1)	(0, 1, 0, 1, 0)	(0, 0, 1, 1, 0)
Délka úseku	5	5	5	5	5
Accuracy	0.929	0.928	0.928	0.927	0.926
F míra H-mód	0.936	0.936	0.936	0.935	0.934
F míra L-mód	0.909	0.907	0.907	0.906	0.905
F míra ELM	1.000	1.000	1.000	1.000	1.000
F míra průměrná	0.948	0.948	0.948	0.947	0.946
Precision H-mód	0.956	0.956	0.954	0.954	0.952
Precision L-mód	0.885	0.882	0.884	0.882	0.882
Precision ELM	1.000	1.000	1.000	1.000	1.000
Recall H-mód	0.918	0.917	0.918	0.917	0.917
Recall L-mód	0.935	0.935	0.932	0.932	0.930
Recall ELM	1.000	1.000	1.000	1.000	1.000

Tab. 4.3: Výsledky algoritmu K-means aplikovaného na syntetická data ($\frac{1}{20}$ normálního šumu). Nejlepších pět kombinací příznaků a délek úseku seřazených podle průměrné F-míry.

index	5	6	13	21	36
Kombinace příznaků	(0, 0, 1, 1, 0)	(0, 0, 1, 1, 1)	(0, 1, 1, 1, 0)	(1, 0, 1, 1, 0)	(0, 0, 1, 1, 0)
Délka úseku	5	5	5	5	6
Accuracy	0.884	0.884	0.882	0.882	0.880
F míra H-mód	0.898	0.898	0.897	0.897	0.894
F míra L-mód	0.849	0.849	0.847	0.846	0.846
F míra ELM	0.989	0.989	0.989	0.989	0.989
F míra průměrná	0.912	0.912	0.911	0.911	0.910
Precision H-mód	0.905	0.905	0.904	0.903	0.909
Precision L-mód	0.839	0.839	0.837	0.838	0.826
Precision ELM	1.000	1.000	1.000	1.000	1.000
Recall H-mód	0.892	0.892	0.890	0.892	0.880
Recall L-mód	0.860	0.860	0.858	0.855	0.868
Recall ELM	0.980	0.980	0.980	0.980	0.980

Tab. 4.4: Výsledky algoritmu K-means aplikovaného na syntetická data ($\frac{1}{13}$ normálního šumu). Nejlepších pět kombinací příznaků a délek úseku seřazených podle průměrné F-míry.

4.3 Výsledky experimentu na reálných datech

V této podkapitole vyhodnotíme výsledky z experimentu na reálných datech. Nejdříve zhodnotíme kvalitu skrytého Markovova modelu bez učitele, následovat bude jeho modifikovaná verze. Posledním v pořadí je algoritmus K-means. Nakonec všechny tři porovnáme mezi sebou. Výsledy v tabulkách byly získány technikou křížové validace a jedná se tedy o průměrné výsledky.

Stejně jako na syntetických datech, tak i na reálných si skrytý Markovův model bez učitele vedl velice dobře. V Tab. 4.5 jsou uvedeny výsledky modelu trénovaného na datech z Obr. 3.6 a v Tab. 4.6 jsou výsledky tréninku dat z Obr. 3.7.

Během testů jsme se rozhodli dát modelu možnost využít klouzavých průměrů na více úsecích o různé délce. Upořádaná čtveřice délky úseku, jež je patrná ve všech následujících tabulkách, představuje tyto různé úseky, kde na prvních třech pozicích jsou úseky ω, kterých mohou využívat ony průměry. Poslední hodnota udává úsek, který je využit pro výpočet klouzavého rozptylu.

I přes tuto skutečnost shledáváme, že nejlepší výsledky z obou datasetů jsou s exponenciálním klouzavým průměrem počítaným pouze s jednou délkou úseku. Liší se od sebe jen v délce tohoto úseku. Nejlepší kombinace příznaků je pro oba datasety kombinace S-G filtru, který využíváme na $n_L + n_R + 1 = 9$, exponenciálního klouzavého průměru s $\omega_{ekp} = 16$ a klouzavým rozptylem s $\omega_{kr} = 14$ (resp. 11).

index	2843	2844	2831	2830	2832
Kombinace příznaků	(1, 0, 1, 0, 1)	(1, 0, 1, 0, 1)	(1, 0, 1, 0, 1)	(1, 0, 1, 0, 1)	(1, 0, 1, 0, 1)
Délky úseků	(0, 0, 16, 14)	(0, 0, 16, 15)	(0, 0, 14, 14)	(0, 0, 14, 13)	(0, 0, 14, 15)
Accuracy	0.833	0.832	0.831	0.832	0.830
F míra H-mód	0.901	0.900	0.901	0.903	0.899
F míra L-mód	0.726	0.721	0.727	0.732	0.719
F míra ELM	0.782	0.787	0.775	0.768	0.784
F míra průměrná	0.803	0.803	0.801	0.801	0.801
Precision H-mód	0.951	0.963	0.951	0.940	0.961
Precision L-mód	0.606	0.594	0.603	0.613	0.591
Precision ELM	0.843	0.843	0.841	0.843	0.844
Recall H-mód	0.857	0.846	0.858	0.869	0.846
Recall L-mód	0.911	0.921	0.919	0.912	0.923
Recall ELM	0.732	0.741	0.722	0.709	0.735

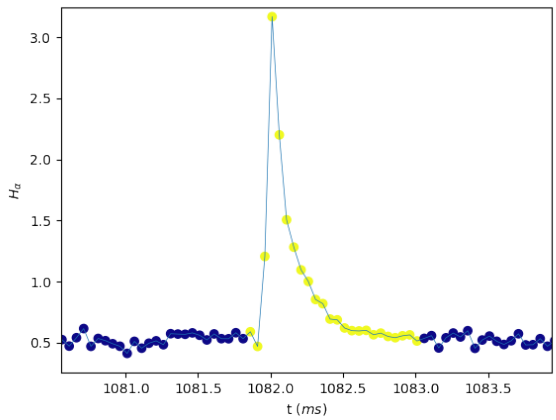
Tab. 4.5: Výsledky skrytého Markovova modelu (bez učitele) aplikovaného na reálná data z Obr. 3.6. Nejlepších pět kombinací příznaků a délek úseku seřazených podle průměrné F-míry. Uspořádaná pětice (0, 1, 1, 0, 0) ≈ (první derivace, druhá derivace, exponenciální klouzavý průměr, klouzavý průměr, klouzavý rozptyl) udává, které příznaky byly použity (na jejich pozici je 1) a které ne (na jejich pozici je 0).

index	2840	2828	2841	2827	2842
Kombinace příznaků	(1, 0, 1, 0, 1)	(1, 0, 1, 0, 1)	(1, 0, 1, 0, 1)	(1, 0, 1, 0, 1)	(1, 0, 1, 0, 1)
Délky úseku	(0, 0, 16, 11)	(0, 0, 14, 11)	(0, 0, 16, 12)	(0, 0, 14, 10)	(0, 0, 16, 13)
Accuracy	0.769	0.766	0.767	0.767	0.765
F míra H-mód	0.807	0.806	0.804	0.807	0.802
F míra L-mód	0.690	0.692	0.685	0.696	0.685
F míra ELM	0.589	0.583	0.592	0.576	0.593
F míra průměrná	0.695	0.694	0.694	0.693	0.693
Precision H-mód	0.847	0.842	0.855	0.833	0.866
Precision L-mód	0.609	0.615	0.606	0.617	0.602
Precision ELM	0.672	0.669	0.672	0.667	0.676
Recall H-mód	0.780	0.783	0.770	0.791	0.759
Recall L-mód	0.852	0.848	0.848	0.851	0.858
Recall ELM	0.592	0.587	0.599	0.571	0.600

Tab. 4.6: Výsledky skrytého Markovova modelu (bez učitele) aplikovaného na reálná data z Obr. 3.7. Nejlepších pět kombinací příznaků a délek úseku seřazených podle průměrné F-míry.

Další na řadě je skrytý Markovův model modifikovaný. Od standartní verze modelu bez učitele se liší větší přesností u obou datasetů. Je tedy zřejmé, že předpočítání parametrů má kladný dopad na kvalitu modelu. U datasetu z Obr. 3.6 se přesnost zvýšila o $3,5\%$ a průměrná F-míra vzrostla dokonce o $5,5\%$ viz. Tab. 4.7. Ona modifikace má i další výhody, např. už není třeba přerovnávat výsledky, protože skupiny potažmo stavy jsou určeny správně díky lepší inicializaci a dokonce se snížila i doba potřebná k tréninku.

Co se týká druhého datasetu z Obr. 3.7 je navýšení přesnosti a F-míry téměř totožné viz Tab. 4.8. Jedním z důvodů tohoto zpřesnění je lepší schopnost rozpoznat ELM, což je dobře patrné u jeho recallu, kde je nárůst u nejlepších kombinací 18%. Navíc je zde využit plný potenciál z vícero úseků pro klouzavé průměry.



Obr. 4.2: Oříznutá část signálu zobrazující jeden ELM. Žluté tečky symbolizjí body ELMu a modré jsou body příslušící H-módu. Data jsou označena na základě informací z databáze. (Číslo výstřelu: 15311)

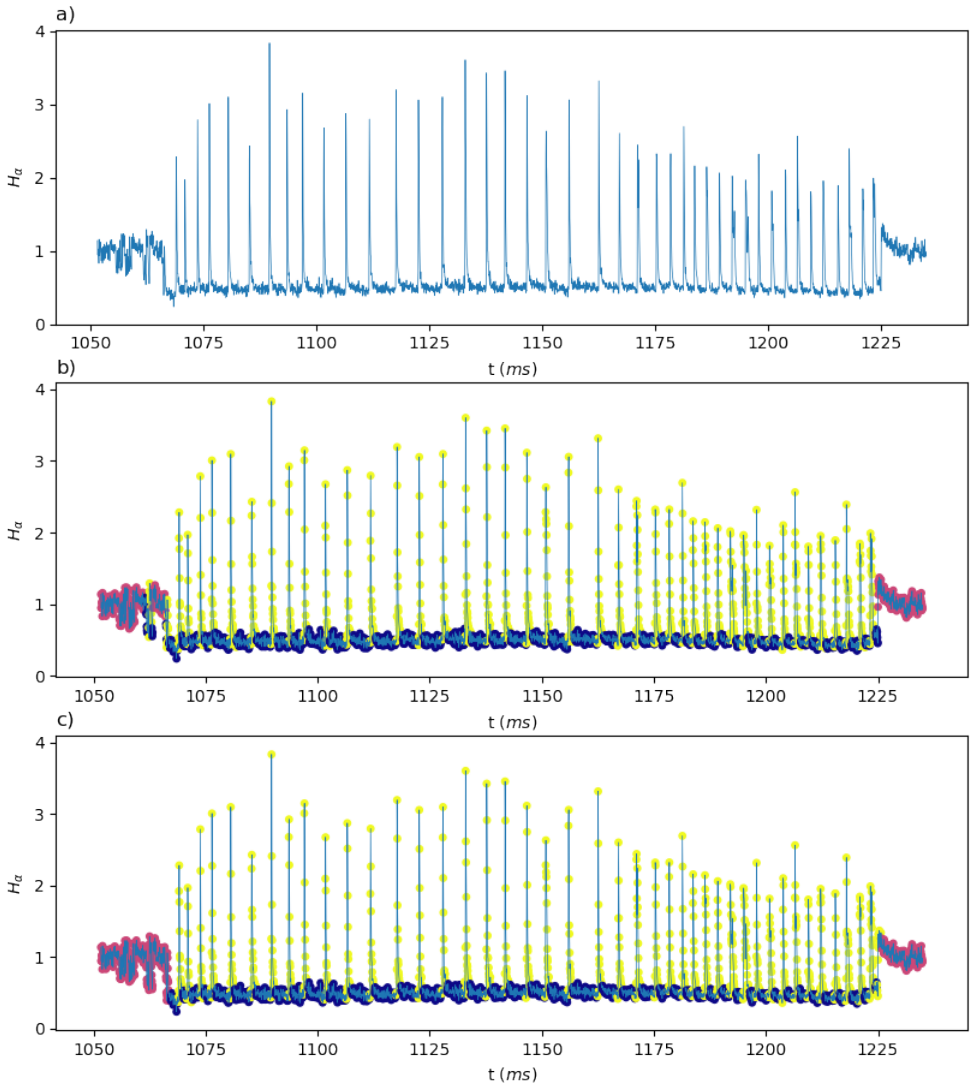
Konkrétně detekce ELMů je onou příčinou, jež nejvíce snižuje kvalitu modelů. Děje se tak díky jeho specifickému průběhu. Když se podrobně podíváme na Obr. 4.2, můžeme vidět, že konec ELMu odpovídá spíše charakteru H-módu, a právě v tom často modely chybují. Takto vzniklá chyba neovlivňuje příliš přesnost, protože v porovnání s ostatními stavy se jedná o zanedbatelný počet bodů, ale velice ovlivňuje jeho F-míru. Proto také hodnotíme kvalitu modelu hlavně podle F-míry.

index	4118	4106	3950	2932	3998
Kombinace příznaků	(1, 1, 1, 1, 1)	(1, 1, 1, 1, 1)	(1, 1, 1, 1, 1)	(1, 0, 1, 1, 1)	(1, 1, 1, 1, 1)
Délky úseků	(6, 10, 16, 16)	(6, 10, 14, 16)	(4, 8, 12, 16)	(4, 10, 14, 16)	(4, 10, 14, 16)
Accuracy	0.868	0.867	0.866	0.867	0.866
F míra H-mód	0.897	0.897	0.895	0.898	0.896
F míra L-mód	0.851	0.851	0.855	0.848	0.849
F míra ELM	0.826	0.823	0.820	0.823	0.823
F míra průměrná	0.858	0.857	0.856	0.856	0.856
Precision H-mód	0.922	0.919	0.910	0.922	0.919
Precision L-mód	0.895	0.902	0.893	0.893	0.884
Precision ELM	0.788	0.787	0.796	0.791	0.793
Recall H-mód	0.875	0.877	0.881	0.877	0.876
Recall L-mód	0.828	0.824	0.835	0.825	0.833
Recall ELM	0.872	0.867	0.849	0.866	0.861

Tab. 4.7: Výsledky modifikovaného skrytého Markovova modelu aplikovaného na reálná data z Obr. 3.6. Nejlepších pět kombinací příznaků a délek úseku seřazených podle průměrné F-míry.

index	2206	2557	2593	2314	2665
Kombinace příznaků	(1, 0, 0, 1, 1)	(1, 0, 1, 0, 1)	(1, 0, 1, 0, 1)	(1, 0, 0, 1, 1)	(1, 0, 1, 0, 1)
Délky úseků	(4, 8, 16, 16)	(4, 8, 16, 16)	(4, 10, 16, 16)	(6, 8, 16, 16)	(6, 8, 16, 16)
Accuracy	0.812	0.811	0.809	0.808	0.810
F míra H-mód	0.793	0.793	0.791	0.791	0.793
F míra L-mód	0.764	0.754	0.749	0.752	0.746
F míra ELM	0.684	0.686	0.687	0.683	0.686
F míra průměrná	0.747	0.744	0.743	0.742	0.742
Precision H-mód	0.858	0.869	0.871	0.857	0.877
Precision L-mód	0.798	0.765	0.758	0.768	0.745
Precision ELM	0.640	0.637	0.639	0.641	0.638
Recall H-mód	0.749	0.740	0.736	0.744	0.736
Recall L-mód	0.779	0.782	0.781	0.779	0.784
Recall ELM	0.771	0.776	0.779	0.769	0.776

Tab. 4.8: Výsledky modifikovaného skrytého Markovova modelu aplikovaného na reálná data z Obr. 3.7. Nejlepších pět kombinací příznaků a délek úseku seřazených podle průměrné F-míry.



Obr. 4.3: Příklad klasifikace. Na obrázku *a*) je vyobrazen signál H_α zaznamenaný fotonásobičem. Na obrázku *b*) jsou barevně označeny stavy podle databáze a na obrázku *c*) jsou stavy podle modifikovaného skrytého Markovova modelu. Fialová barva odpovídá L-módu, modrá H-módu a žlutá ELMu. (Číslo výstřelu: 15349)

Poslední metoda, kterou je K-means, se projevila jako nehorší. Vzhledem k tomu, že se jedná o poměrně jednoduchý algoritmus, tak i jeho přesnost není zase tak špatná. Nejvíce vypovídající, průměrná F-míra, je u prvního datasetu 64, 6% viz Tab. 4.9, ale u druhého datasetu je to sotva 54, 6% viz Tab. 4.10. Příčina toho se skrývá v opět detekci ELMu.

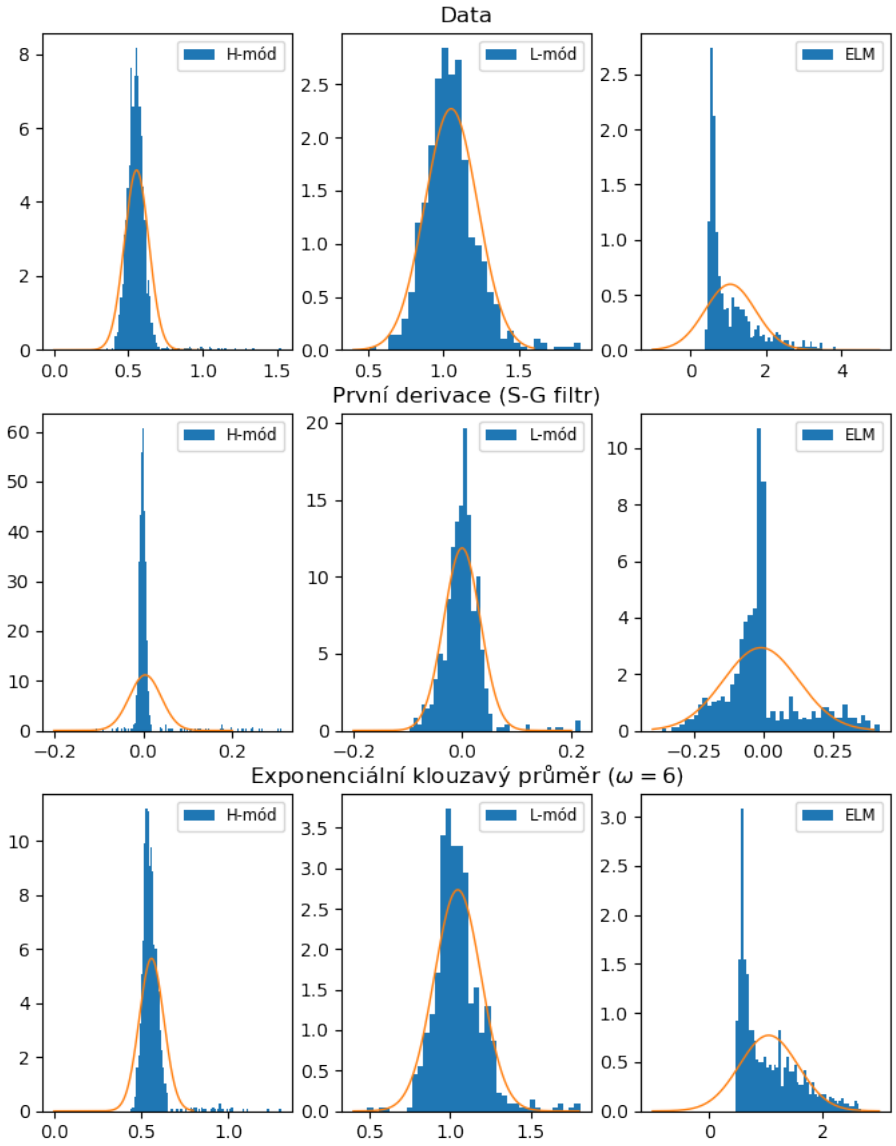
index	2988	856	4054	2989	857
Kombinace příznaků	(1, 0, 1, 1, 1)	(0, 0, 1, 1, 1)	(1, 1, 1, 1, 1)	(1, 0, 1, 1, 1)	(0, 0, 1, 1, 1)
Délky úseků	(6, 8, 12, 12)	(6, 8, 12, 12)	(6, 8, 12, 12)	(6, 8, 12, 13)	(6, 8, 12, 13)
Accuracy	0.734	0.735	0.734	0.735	0.734
F míra H-mód	0.859	0.859	0.859	0.860	0.860
F míra L-mód	0.699	0.699	0.699	0.697	0.697
F míra ELM	0.379	0.379	0.378	0.379	0.379
F míra průměrná	0.646	0.645	0.645	0.645	0.645
Precision H-mód	0.769	0.769	0.769	0.771	0.771
Precision L-mód	0.581	0.581	0.581	0.578	0.578
Precision ELM	0.949	0.949	0.949	0.947	0.946
Recall H-mód	0.981	0.981	0.981	0.981	0.981
Recall L-mód	0.890	0.890	0.890	0.891	0.890
Recall ELM	0.241	0.241	0.240	0.241	0.241

Tab. 4.9: Výsledky algoritmu K-means aplikovaného reálná data z Obr. 3.6. Nejlepších pět kombinací příznaků a délek úseku seřazených podle průměrné F-míry.

index	3134	4200	3135	4201	4202
Kombinace příznaků	(1, 0, 1, 1, 1)	(1, 1, 1, 1, 1)	(1, 0, 1, 1, 1)	(1, 1, 1, 1, 1)	(1, 1, 1, 1, 1)
Délky úseků	(0, 10, 12, 14)	(0, 10, 12, 14)	(0, 10, 12, 15)	(0, 10, 12, 15)	(0, 10, 12, 16)
Accuracy	0.684	0.684	0.683	0.683	0.683
F míra H-mód	0.750	0.750	0.750	0.750	0.751
F míra L-mód	0.495	0.495	0.494	0.494	0.494
F míra ELM	0.393	0.393	0.393	0.393	0.392
F míra průměrná	0.546	0.546	0.546	0.546	0.545
Precision H-mód	0.664	0.664	0.665	0.665	0.666
Precision L-mód	0.526	0.526	0.527	0.527	0.528
Precision ELM	0.612	0.612	0.611	0.611	0.61
Recall H-mód	0.875	0.875	0.875	0.875	0.874
Recall L-mód	0.569	0.568	0.568	0.568	0.568
Recall ELM	0.353	0.353	0.353	0.353	0.352

Tab. 4.10: Výsledky algoritmu K-means aplikovaného na reálná data z Obr. 3.7. Nejlepších pět kombinací příznaků a délek úseku seřazených podle průměrné F-míry.

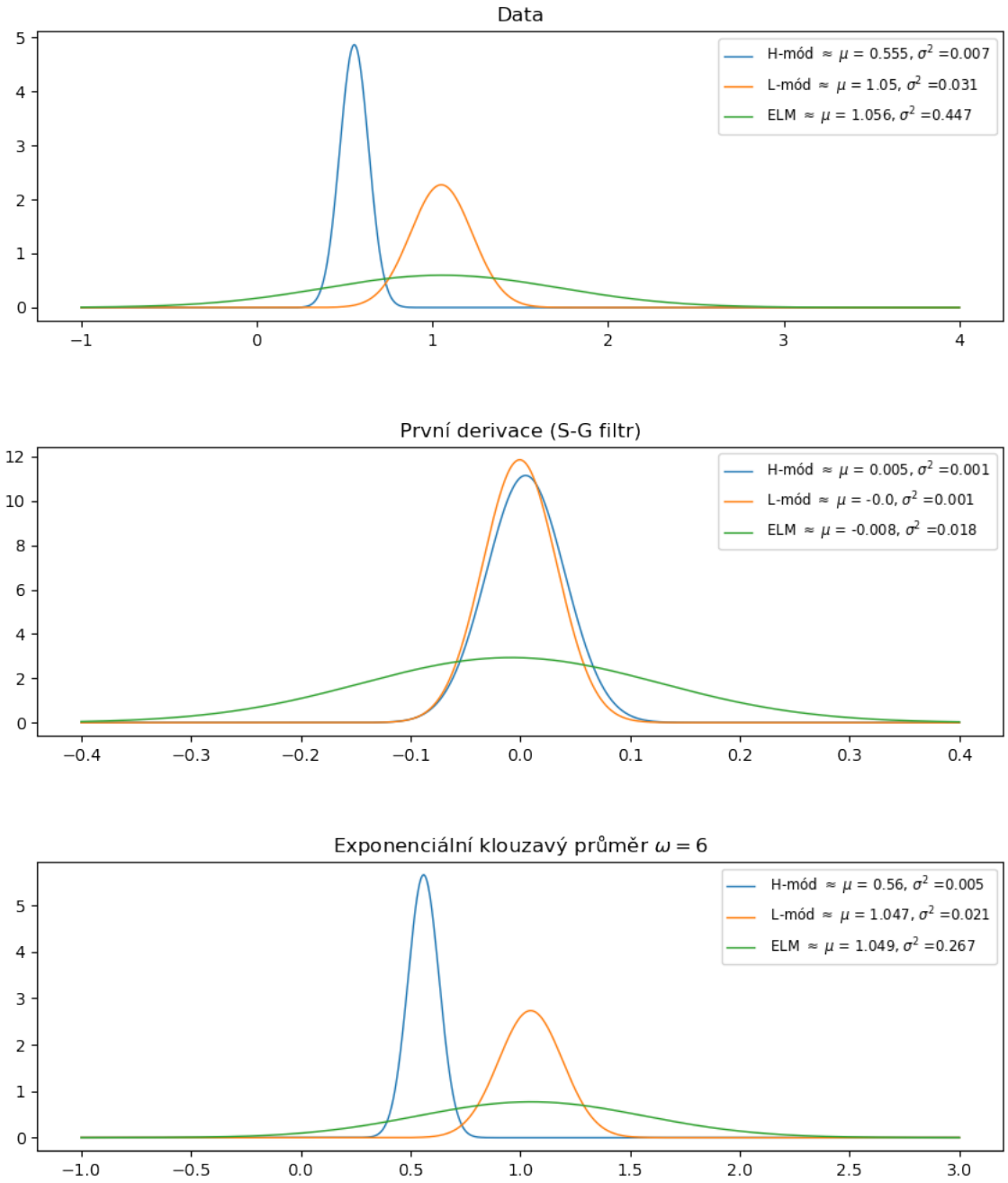
Důvodem, proč jsme ze skrytého Markovova modelu nezískali ještě lepší výsledky, je především to, že příznaky nemají Gaussovo rozdělení. V předchodí kapitole jsme uvedli, že předpokládáme toto rozdělení u všech přínaků v každém stavu, abychom mohli využít predimplementovanou třídu GaussianHMM využívající při výpočtu Gaussovy emisní pravděpodobnosti. Když ovšem vytvoříme histogramy příznaků pro jednotlivé stavy, pak je jednoznačně patrné, že Gaussovské nejsou (viz Obr. 4.4). I když Gaussovy křivky neodpovídají přímo histogramům, lze je u skrytého Markova využít, protože model především potřebuje, aby se tyto křivky nepřekrývaly a z většiny příznaků tomu tak je (viz Obr. 4.5).



Obr. 4.4: Histogramy příznaků. První tři histogramy přdstavují naměřené hodnoty H_α roztříděny podle stavů. Na dalších histogramech jsou dva příznaky, kontrétně první derivace (Savitzky-Golay filtr) a exponenciální klouzavý průměr. Oranžové křivky jsou Gaussovy křivky, jež by měli odpovídat oněm příznakům. (Číslo výstřelu: 15311)

Pro vylepšení modelu by se případně dalo uvažovat o jiných emisních pravděpodobnostech. Mohli bychom třeba použít třídu GMMHMM, která sice opět využívá Gaussových křivek, ale připoušítí i smíšené Gaussovky. Tímto způsobem lze řešit například nevyhovující křivky pro příznaky ELMu, jež se

od histogramů liší nejvýznamněji. Další možností je rozdělit ELMy na dva samostatné stavy (začátek a konec). Nebo se pokusit nalézt odpovídající rozdělení všech stavů a napsat vlastní knihovnu, který by pracovala přímo s těmito rozděleními.



Obr. 4.5: Gaussovy křivky příznaků. (Číslo výstřelu: 15311)

Závěr

V této práci jsme se zabývali klasifikací stavů plazmatu uvnitř tokamaku COMPASS. K tomu jsme využívali metody strojového učení, jmenovitě skrytý Markovův model a K-means clustering. Nejdříve jsme se seznámili se zářením H_α, tokamakem COMPASS a fyzikální podstatou jevů, které v něm při výbojích vznikají. Poté jsme zadefinovali diskrétní Markovův model, jež jsme později rozšířili do skrytého Markovova modelu a dále jsme popsali dva důležité algoritmy potřebné k jeho výpočtu, jimiž jsou expectation–maximization a Viterbiho algoritmus. Následně jsme představili shlukovou metodu K-means, kterou jsme pak sami implementovali v jazyce Python.

V praktické části jsme postupně zadefinovali derivaci, Savitzky-Golay filtr, klouzavý průměr, exponenciální klouzavý průměr a nakonec klouzavý rozptyl. Pak jsme naplánovali dva experimenty, na kterých byly později obě metody testovány. První experiment byl prováděn na syntetických datech zašuměných Gaussovským bílým šumem. Při tomto experimentu jsme ukázali, že vybrané metody lze využít ke klasifikaci takovýchto dat. Pro druhý experiment jsme z databáze vybrali dva datasety skládající se z deseti signálů a začali používat více délek úseků pro klouzavé průměry. Navíc byl zde poprvé uveden způsob modifikace skrytého Markovova modelu, ve kterém předpočítáváme některé parametry dopředu.

V poslední kapitole jsme představili způsoby, které jsme poté používali při vyhodnocování kvality modelů. Abychom se vyhnuli případnému overfittingu, rozhodli jsme se pro techniku křížové validace, kde trénování a testování probíhalo v poměru 9:1. Při prvním experimentu jsme zjistili, že nejdůležitějším příznakem se stal exponenciální klouzavý průměr počítaný na úseku délky 5, který se pro obě velikosti šumu umístil nejlépe. V obou případech tato kombinace dosáhla více než $92,7\%$ přesnosti a $94,8\%$ průměrné F míry (viz Tab. 4.1 a 4.2). K-means dopadli o něco hůře, přičemž kombinace exponenciálního klouzavého průměru a klouzavého průměru na úseku délky 5 překročili $91,2\%$ F-míry (viz Tab. 4.3 a 4.4).

Při druhém experimentu se opět potvrdila převaha skrytého Markovova modelu, který si v obou formách, standartní bez učitele a modifikované s učitelem, vedl mnohem lépe, než K-means. Skrytý Markovův model dosáhl u prvního datasetu nejlepší přesnosti 83% a F-míry 80% (viz Tab. 4.5), zatímco u druhého datasetu přesnost i F-míra výrazně klesly (viz Tab. 4.6). Modifikovaná verze tohoto modelu si ovšem vedla o něco lépe. U prvního (resp. druhého) datasetu jsme dosáhli zlepšení o $3,5\%$ (resp. $4,3\%$) v přesnosti a $5,5\%$ (resp. $15,3\%$) v průměrné F-míře (viz Tab. 4.7 a 4.8). Na základě výsledků bych chtěl tento model prohlásit za nejlepší způsob klasifikace stavů plazmatu ze všech, které jsme otestovali v této práci.

Největším problémem bránícím lepším výsledkům je skutečnost, že naše příznaky nemají Gaussovo rozdělení, které předpokládáme při výpočtu skrytého Markovova modelu. Za důkaz tohoto tvrzení lze považovat Obr 4.4, kde je odchýlení histogramů od předpovídaného rozdělení nejlépe patrné u ELMu. Pokud bychom chtěli případně model nějak vylepšit, stálo by za úvahu použití třídy GMMHMM (Gaussian Mixture Model Hidden Markov Model), ve které rozdělení příznaků vznikne smíšením několika

Gaussových rodělení dohromady. Další možností je rozlišovat ELM na dva samostatné stavy, nebo nalezení skutečného rozdělení příznaků a implementace vlastní knihovny.

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=Sxhl8ae680b43c26317b013c953679dd020a

[14] http://hmmlearn.readthedocs.io/en/latest/api.html#hmmlearn-hmm

[15] Tudor Barbu, *Variational Image Denoising Approach with Diffusion Porous Media Flow*. Abstract and Applied Analysis. 2013: 8. doi:10.1155/2013/856876, 2013.

[16] Kohavi, Ron, *A study of cross-validation and bootstrap for accuracy estimation and model selection*. Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann. 2 (12): 1137–1143. CiteSeerX 10.1.1.48.529, 1995.

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

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

[19] https://www.researchgate.net/profile/Janos_Egert/publication/283617947/figure/fig1/AS:314833057665024@1452073458382/View-of-the-COMPASS-tokamak-left-and-its-vacuum-chamber-with-diagnostic-ports-right.png