

České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství
Obor: Aplikace softwarového inženýrství



Kombinace prediktorů v odhadování parametrů

Mixing of Predictors in Parameter Estimation

BAKALÁŘSKÁ PRÁCE

Vypracoval: Yana Podlesna
Vedoucí práce: Ing. Miroslav Kárný, DrSc.
Rok: 2020

České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství

Akademický rok 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Yana Podlesna
Studijní program: Aplikace přírodních věd
Obor: Aplikace softwarového inženýrství
Název práce česky: Kombinace prediktorů v odhadování parametrů
Název práce anglicky: Mixing of Predictors in Parameter Estimation

Pokyny pro vypracování:

1. Seznamte se s bayesovským odhadováním parametrů.
2. Seznamte se zobecněním 1. využívající prediktor dat místo dat.
3. Navrhněte užití více prediktorů, vzniklých odhadováním jednoduchých parametrických modelů lišících se svými regresory, pro výpočetně zvládnutelné respektování mnoha atributů ovlivňujících předpovídání jedné proměnné.
4. Specializujte obecné řešení ad 3. pro případ jednoduchého parametrického modelu.
5. Implementujte získané algoritmy v jazyce MATLAB nebo Python a vyhodnoťte přínos Vašeho odhadování.

Doporučená literatura:

- [1] PETERKA, V. Bayesian System Identification. In: *Trends and Progress in System Identification*. Oxford: Pergamon Press, 1981, pp. 239-304.
- [2] KRACÍK, J., and KÁRNÝ, M. Merging of data knowledge in Bayesian estimation. In: *Proceedings of the Second Int. Barcelona: Conference on Informatics in Control, Automation and Robotics, INSTICC*, 2005, pp. 229-232.
- [3] QUINN, A., KÁRNÝ, M. and GUY, T.V. Optimal design of priors constrained by external predictors. In: *Journal of Approximate Reasoning*. 2017, 84: pp. 150-158.
- [4] RISH, I. An empirical study of the naive Bayes classifier, T.J. Watson Research Center, 2001.

Jméno a pracoviště vedoucího práce:

Ing. Miroslav Kárný, DrSc.
ÚTIA AVČR, v.v.i.

Jméno a pracoviště konzultanta:

Ing. František Hůla
ÚTIA AVČR, v.v.i.

.....
vedoucí práce

Datum zadání bakalářské práce: 11. 10. 2019

Termín odevzdání bakalářské práce: 7. 7. 2020

Doba platnosti zadání je dva roky od data zadání.

.....
garant oboru

.....
vedoucí katedry

.....
děkan

V Praze dne 11. 10. 2019

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracovala samostatně a použila jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne

.....
Yana Podlesna

Poděkování

Mé poděkování patří Ing. Miroslavu Kárnému, DrSc. za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování bakalářské práce věnoval. Tato práce byla podpořena grantem MŠMT LTC18075 a EU-COST Action CA16228.

Yana Podlesna

Název práce:

Kombinace prediktorů v odhadování parametrů

Autor: Yana Podlesna

Studijní program: Aplikace přírodních věd

Obor: Aplikace softwarového inženýrství

Druh práce: Bakalářská práce

Vedoucí práce: Ing. Miroslav Kárný, DrSc.
ÚTIA AVČR

Konzultant: Ing. František Hůla
ÚTIA AVČR

Abstrakt: Tato bakalářská práce se zabývá návrhem metody k řešení prokletí rozměrnosti vznikající v kvantitativním modelování složitých vzájemně propojených systémů. Jedná se o předpověďací modely, které jsou založené na diskrétním markovském rozhodovacím procesu. Předpovídání je založeno na odhadu parametrů modelu pomocí bayesovské statistiky. Tato práce obsahuje návod na zmenšení rozměrnosti dat, potřebných k předpovídání v systémech s velkým počtem stavů a akcí. Místo odhadu prediktoru závislého na všech parametrech metoda předpokládá užití několika prediktorů, které vznikají odhadováním parametrických modelů, předpokládajících závislost na různých regresorech. Vlastnosti chování navržené metody jsou ilustrovány simulačními experimenty.

Klíčová slova: prokletí rozměrnosti, bayesovské odhadování, předpovídání, markovský rozhodovací proces, rozhodování

Title:

Mixing of Predictors in Parameter Estimation

Author: Yana Podlesna

Abstract: This bachelor thesis deals with the design of the method for solving the curse of dimensionality arising in the quantitative modeling of complex interconnected systems. The employed predictive models are based on a discrete Markov process. Prediction is based on estimating model parameters using Bayesian statistics. This work contains method for reducing the amount of data needed for prediction in systems with a large number of occurring states and actions. Instead of estimating a predictor dependent on all parameters, the method assumes the use of several predictors, which arise from estimating parametric models based on dependences on different regressors. The behavioral properties of the proposed method are illustrated by simulation experiments.

Key words: curse of dimensionality, Bayesian estimation, prediction, Markov decision process, decision making

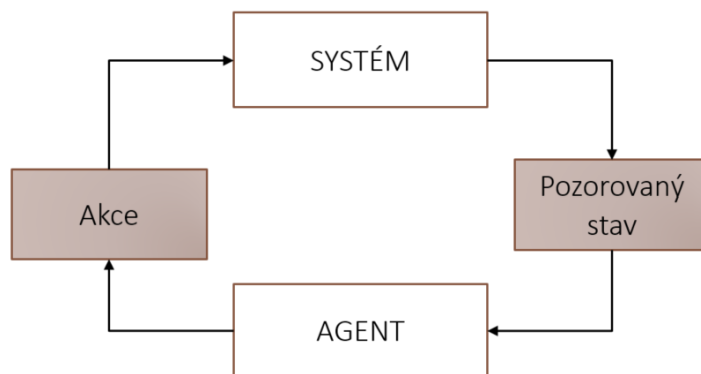
Obsah

Úvod	8
1 Přípravná kapitola	11
1.1 Bayesovský odhad pravděpodobnosti přechodu	12
1.2 Slučování dat	14
2 Návrh metody	16
2.1 Odhadování přechodové pravděpodobnosti	16
2.2 Kombinace prediktorů	19
3 Experimenty	22
3.1 Základní předpoklady experimentů	22
3.2 Organizace výpočtu	23
3.3 Výsledky	27
3.3.1 Systém EX1	27
3.3.2 Systém EX2	30
3.3.3 Systém EX3	32
3.3.4 Souhrnná diskuze	35
Závěr	36
Literatura	38
Přílohy	39
A Obsah přiloženého paměťového média	39

Úvod

Rozhodování je každodenní činnost pro běžného člověka. Každý den narážíme na situace, ve kterých si musíme vybrat. Některá rozhodnutí jsou jednoduchá, některá jsou mnohem složitější a potřebují hodně znalostí. Matematicky můžeme rozhodování definovat tak, jak jej běžně chápeme, avšak musíme uvažovat nějakou uzavřenou část světa, kterou nazýváme systém. Rozhodovací proces může být obecně popsán pomocí posloupnosti akcí a pozorovaných stavů v systému. Realizované stavy a akce, které systém ovlivňují, obecně můžeme nazývat regresory.

Každá provedená akce ovlivňuje systém a generuje závislý pozorovaný stav. Kvůli tomu, že různé akce vedou k různým pozorovaným stavům, je rozumné se ptát, která posloupnost akcí vede v průměru k nejvyššímu zisku. Při správném definování systému může rozhodovací algoritmus, agent, předpovídat další stav, který očekáváme po určité akci a tím ovlivňovat systém ve svůj prospěch. Na obrázku 1.1 je ukázána uzavřená smyčka propojující agenta a systém, který generuje stavy. Agent následně vybírá akce ovlivňující systém.



Obrázek 1: Uzavřená smyčka

Systémy skutečného světa mají tendenci být nejisté, což znamená, že jedna akce může vést k odlišnému stavu s různými pravděpodobnostmi. Tato pravděpodobnost přechodu obvykle není známa. Odhad pravděpodobnosti přechodu zde bude proveden na základě bayesovského přístupu. Teorie se opírá hlavně o [8]. Bayesovská statistika jako alternativa klasické nabízí příležitost získat kvalifikovaný odhad parametrů procesu dynamicky a to při malých počtech dat.

Cíl agenta je udělat nejlepší odhady pravděpodobností přechodu na základě dostupných údajů. Rozhodování za neurčitosti může být provedeno pomocí markovského rozhodovacího procesu za podmínek, kdy je možno brát v úvahu všechny akce a pozorované stavy, stejně jako vzájemnou souvislost. Potřebná teorie odhadování a předpovídání je popsána v kapitole 1.

Příklad. Představme si, že chceme pěstovat balkónová rajčata. Sledujeme za jakých podmínek rostou dobře nebo špatně. Víme, že jejich růst závisí na tom, jestli jsme rajčata zalévali a jestli venku svítilo slunce nebo ne. Tím pádem definujeme systém závislý na stavech: rajčata rostou dobře nebo špatně, zalitá nebo ne, svítilo slunce nebo bylo zataženo a akci: zalít nebo ne. Různé stavy se navzájem ovlivňují. Například když zaléváme rajčata potom co bylo zataženo, tak pravděpodobně nebudou růst dobře. Naopak pravděpodobnost přechodů ke stavu "rostou dobře" je velice pravděpodobná potom, co jsme rajčata polili a bylo slunečno.

V uvedeném příkladě se systém ovlivňuje jen několika jednoduchými stavy a akcemi. V takových systémech výpočet předpovědi není složitý ani rozsáhlý. Čím více akcí a stavů v systému máme, tím složitější je proces hledání nejlepšího rozhodnutí pro agenta.

Pro správnou předpověď je zapotřebí více údajů ohledně pravděpodobnosti přechodu každého stavu při každé akci. Například když chceme předpovídat růst akcií na trhu, musíme brát v úvahu všechno, co by mohlo akcie ovlivnit. Když se zamyslíme nad rozměrem těchto dat, jednoduše vidíme, že množství dat potřebných k odhadnutí dalšího stavu roste exponenciálně s počtem možností. Kvůli tomu předpověď v složitých systémech potřebuje velké množství dat, které často není k dispozici. Získávání takového množství dat může být poměrně dlouhé a drahé, často i nemožné. Tento problém je také nazýván prokletí rozměrnosti (angl. curse of dimensionality).

Kvantitativní modelování složitých vzájemně propojených systémů (nutné pro rozhodování) představuje závažný problém pro vědecký výzkum v různých oborech, včetně ekologie [11] a meteorologie [6], financí, neurobiologie a medicíny [1].

Tato bakalářská práce je zaměřena na řešení daného problému. Chceme-li ovlivnit systém žádoucím způsobem, musíme předpovídat, jaký bude následný stav po stavu, který byl pozorován a po akci, kterou jsme zvolili. K tomu je zapotřebí sestavit odpovídající předpovědčí model. Navržená metoda spočívá v rozdělení tohoto předpovědčího modelu na několik prediktorů, které vycházejí z odhadu jednoduchých parametrických modelů lišících se jejich regresory. Díky této metodě lze provést náročnější výpočty s respektováním mnoha atributů ovlivňujících predikci jedné proměnné.

Po přípravné kapitole 1 je v kapitole 2 uveden obecný návrh této metody. V sekci 2.1 lze najít detailnější popis konstrukce modelu odhadující přechodové pravděpodobnosti. Sekce 2.2 se zabývá rozdělením předpovědčího modelu na více prediktorů a jejich nasledované slučování dat.

V kapitole 3 je popsána simulace metody na jednoduchých systémech, ve kterých si pamatujeme 2 předchozí stavy, čili pracujeme s pamětí dva. Předpoklady experimentu jsou popsány v sekci 3.1. V sekci 3.2 je uveden algoritmus simulace. Dosažené

výsledky pro každý experiment jsou ukázány a diskutovány v sekci 3.3. Poté jsou shrnuty v Závěru.

Kapitola 1

Přípravná kapitola

Tato kapitola poskytuje informace o základních definicích a větách, které budou používané v této práci.

pojem	značení
přirozená čísla	\mathbb{N}
reálná čísla	\mathbb{R}
pravděpodobnost jevu a	$P(a)$
pravděpodobnost jevu a za podmínky jevu b	$P(a \mid b)$
s náleží do S	$s \in S$
x je úměrné y	$x \propto y$

Množinu stavů značíme $S = \{s_1, s_2, \dots, s_n\}$ pro všechny $n \in \mathbb{N}$.

Množinu akcí značíme $A = \{a_1, a_2, \dots, a_m\}$ pro všechny $m \in \mathbb{N}$.

Čas t je diskrétní veličina $t = 1, 2, \dots$

K matematickému popisu se používá diskrétní markovský rozhodovací proces, který poskytuje matematický rámec pro modelování rozhodování v situacích, kdy jsou výsledky zčásti náhodné a zčásti pod kontrolou uživatele.

Definice 1.1: Diskrétní markovský rozhodovací proces je uspořádaná čtveřice $(S, A, P(\cdot, \cdot), R(\cdot, \cdot))$, kde

- S je konečná množina stavů,
- A je konečná množina akcí,
- $P(s \mid a, s') = \Pr(s_{t+1} = s \mid a_t = a, s_t = s')$ je pravděpodobnost, že akce a ve stavu s' v čase t povede v čase $t + 1$ do stavu s ,
- $R(s \mid a, s')$ je okamžitý užitek dosažený po přechodu stavu na s ze stavu s' s pravděpodobností přechodu $P(s \mid a, s')$.

Teorie k markovským rozhodovacím procesům se opírá o [5] a [3]. Tato práce slouží pro libovolné R a jeho konkrétní volba není pro tuto práci důležitá.

Označme vektor $d_t = (s_t, a_t)$ jako dvojici s_t, a_t v určitém čase t a $d(t) = (d_t, d_{t-1}, \dots, d_1)$ jako množinu všech dvojic d_t, d_{t-1}, \dots, d_1 proběhlých vzhledem k určitému času t .

Předpoklad 1.2 Předpokládáme, že zkoumaný systém má konečnou paměť $p \in \mathbb{N}$, což znamená, že pravděpodobnost přechodu $P(s_t | a_t, d(t-1)) = P(s_t | a_t, d_{t-1}, \dots, d_{t-p})$.

Značíme $r_t = \{a_t, s_{t-1}, a_{t-1}, s_{t-2}, \dots, s_{t-p}, a_{t-p}\}$ regresní vektor jako podvýběr posledních $p \in \mathbb{N}$ dat.

Kroneckerovu delta-funkci značíme

$$\delta_{ij} = \begin{cases} 1, & \text{pro } i = j \\ 0, & \text{pro } i \neq j \end{cases} \quad (1.1)$$

Pro $x, y > 0$ značíme Beta funkci jako:

$$B(x, y) = \int_0^1 t^{x-1} (1-t)^{y-1} dt. \quad (1.2)$$

Definujeme Gama funkci pro libovolné komplexní číslo, kromě nuly a celých záporných čísel jako:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt. \quad (1.3)$$

Věta 1.3 (*Vztah mezi B a Γ funkcí*): $B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$.

Věta 1.4 (*Vlastnost Γ funkce*): $\Gamma(z+1) = z\Gamma(z)$.

Důkaz vět 1.3 a 1.4 lze najít v [2].

1.1 Bayesovský odhad pravděpodobnosti přechodu

Předpovídání stavu systému založíme na odhadu pravděpodobnosti přechodu k dalšímu stavu na základě historie, která je již známá. Jinak řečeno, předpokládáme, že posloupnost stavů z množiny stavů $S = \{s_1, s_2, \dots, s_n\}$ a posloupnost akcí z množiny $A = \{a_1, a_2, \dots, a_m\}$ jsou agentovi známy v určitém čase t . K předpovědi dalšího stavu potřebujeme rozumět souvislosti mezi předešlými stavy S a akcemi A , takzvané pravděpodobnosti přechodu. Avšak ve většině příkladů nejsou známy předem, proto musíme provést odhadování těchto parametrů za nejistoty.

Odhadování můžeme provést pomocí například klasické nebo bayesovské statistiky. Hlavní výhodou bayesovského odhadu je možnost sestavení modelu, když není k dispozici žádné (nebo malé) množství dat. Teorie a k bayesovskému odhadování je převzata z [8].

Pro výpočet pravděpodobnosti přechodu $P(d_t | d(t-1))$ bude používané bayesovské odhadování. Funkce P je nejprve parametrizována parametrem $\theta \in \Theta$. Tato závislost

je popsána jako $P(d_t \mid d(t-1), \theta)$, díky čemuž se zužuje odhad P jenom na odhad parametru θ .

Při bayesovském přístupu považujeme parametr θ za náhodnou veličinu, jejíž hodnotu sice nepozorujeme, ale jejíž rozdělení považujeme za známé. Pomocí vzorce úplné pravděpodobnosti lze vyjádřit pravděpodobnost přechodu pomocí integrálu přes parametr θ :

$$P(d_t \mid d(t-1)) = \int_{\Theta} P(d_t, \theta \mid d(t-1)) d\theta. \quad (1.4)$$

Věta 1.5 (*Řetězové pravidlo*): Nechť máme posloupnost náhodných veličin $\{x_k\}_{k=1}^n \in X$. Pak platí:

$$P(x_n, x_{n-1}, \dots, x_0) = \prod_{k=2}^n P(x_k \mid x_{k-1}, \dots, x_1) P(x_1).$$

Řetězové pravidlo lze aplikovat na pravděpodobnost přechodů:

$$P(d_t \mid d(t-1)) = \int_{\Theta} P(s_t \mid a_t, d(t-1), \theta) P(a_t \mid d(t-1), \theta) P(\theta \mid d(t-1)) d\theta. \quad (1.5)$$

Dále rozebereme každý člen integrálu zvlášť.

Generátor akcí $P(a_t \mid d(t-1), \theta)$ se řídí agentem. Agent vybírá akci, která bude ovlivňovat systém, proto generátor akcí nezná parametry systému. Pro něj platí tzv. přirozené podmínky řízení.

Předpoklad 1.6: (*Přirozené podmínky řízení*) Pokud agent současně sleduje a řídí systém, pak jeho rozhodnutí (akce), neposkytují žádné další informace o stavu systému a naopak. A platí:

$$P(a_t \mid d(t-1), \theta) = P(a_t \mid d(t-1)). \quad (1.6)$$

Díky předpokladu platí, že generátor akcí není závislý na parametru θ a lze ho vytknout za integrál.

Obecně je parametr θ reálný vektor, pak $P(\theta \mid d(t))$ je tzv. posteriorní hustota pravděpodobnosti. Tato hustota je opravována po každém kroku pozorování dat v uzavřené smyčce. Její úprava je reprezentována následující rovnicí, která vychází z obecnějšího Bayesova pravidla.

Věta 1.7: Nechť x a y jsou náhodné proměnné a X je množina hodnot x . Pak podmíněná pravděpodobnost $p(x \mid y)$ splňuje následující rovnost:

$$P(x \mid y) = \frac{P(y \mid x)P(x)}{\int_X P(y \mid x)P(x)dx},$$

a použitím věty pro náš případ:

$$P(\theta \mid d(t)) = \frac{P(d(t) \mid \theta)P(\theta)}{\int_{\Theta} P(d(t) \mid \theta)P(\theta)d\theta}. \quad (1.7)$$

$P(\theta)$ značíme apriorní hustotu pravděpodobnosti veličiny θ . Tato hustota pravděpodobnosti vyjadřuje apriorní informaci o možných hodnotách parametru θ . Může být

zvolena zcela objektivně, např. na základě zkušenosti s pozorováními z minulosti nebo na základě vnějších informací, např. z fyzikální podstaty problému. Možná je ale také subjektivní volba vyjadřující individuální názor na pravděpodobnosti výskytu jednotlivých hodnot parametru.

Věta 1.7 tak kombinuje apriorní informaci o parametru s informací obsaženou v pozorováních. To ukazuje, že akumulace systému a znalostí se dynamicky vyvíjí v závislosti na vybrané akci a následujícím stavu. Tato pravděpodobnost bude používána hlavně pro odhadování pravděpodobností přechodu v sekci 2.1.

Pravděpodobnost $P(s_t | a_t, d(t-1), \theta)$ a další výpočet aposteriorní hustoty pravděpodobnosti $P(\theta | d(t-1))$ bude popsán v kapitole 2.

1.2 Slučování dat

V dané sekci je uvedena teorie, která umožní kombinovat několik prediktorů. Teorie se hlavně opírá o [4] a [9].

Příklad. Představme si, že chceme předpovídat chování akcií na trhu. Někou dobu sledujeme, jak se chová vybraný systém a získáváme nějaké znalosti ohledně pravděpodobnosti přechodu k dalšímu stavu. Avšak pro kvalitní předpověď stále potřebujeme více dat. Kvůli tomu se obrátíme na poradce, který má další zkušenosti s chováním daného nebo podobného systému a má svojí představu o pravděpodobnosti dalšího stavu pro danou historii chování akcií. Potom rozšíříme svá data pomocí této informace a tím zlepšíme předpovídání.

Matematicky to můžeme popsat tak, že předpovědácí model agenta $P(d_t | d(t-1))$ chceme rozšířit nějakou další informací ohledně chování pozorovaného systému. Díky tomu dostaneme přesnější představu o zkoumaném systému. Tuto znalost převezmeme od tzv. poradce. Předpokládáme, že data poradce jsou v podobném tvaru jako data, která chceme doplnit. Například mohou být získána pomocí jeho vlastního předpovědácího modelu $M()$, který je vypočítán nezávisle na agentovi a má svojí představu o pravděpodobnosti dalšího stavu pro danou historii chování akcií.

Prakticky to proběhne tak, že obnovíme aposteriorní hustotu pravděpodobnosti agenta pomocí sloučení modelu agenta s modelem poradce s nastavením odpovídající váhy důvěry v . Potom na základě získaného odhadu aposteriorní hustoty pravděpodobnosti můžeme sestavit jejich sdruženou přechodovou pravděpodobnost $P(d_t | d(t-1))$. Slučování dat agenta a poradce pro odhad parametru bude probíhat pomocí následující věty:

Věta 1.8: Nechť platí, že $P(s_{t+1} | a_{t+1}, r_t, \theta)$ je parametrický model systému, M je hustota pravděpodobnosti přidávaných dat, $v \geq 0$ je jejich váha důvěry. Pak platí, že aposteriorní hustotu pravděpodobnosti $P(\theta | d(t), M, v)$ lze spočítat jako:

$$P(\theta | d(t), M, v) \propto P(\theta | d(t)) \exp\left\{v \int_s M(s | a_{t+1}, r_t) \ln(P(s | a_{t+1}, r_t, \theta))\right\}.$$

Odvození a důkaz věty lze najít v příloze [4].

Na rozdíl od M nemusí být příslušná váha důvěry k poradcovi v předem známá. Obvykle je subjektivně přiřazená účastníkem provádějícím parametrický odhad a vyjadřuje váhu, kterou dává přidanému modelu systému.

Díky váze důvěry můžeme nastavit, jak moc "věříme" poradcovi, což znamená nastavit vliv přidaných dat na aposteriorní hustotu pravděpodobnosti pro nejlepší odhad pravděpodobnosti přechodu systému. Taky lze například odhadnout váhu důvěry tak, aby poskytovala co nejlepší pomoc poradce pro konkrétní systém.

Kapitola 2

Návrh metody

V této kapitole je popsán návrh vlastní metody. Hlavní myšlenkou je zmenšení dimenze předpověďacího modelu zmenšením počtu odhadování pravděpodobnosti přechodů. Díky tomu zjednodušíme výpočet predikce regresorů pro modely s velkým počtem vyskytujících stavů a akcí a užitím odpovídajících prediktorů bez velké ztráty kvality předpovídání.

Zkoumáme předpovídání dalšího stavu systému na základě předchozí informace. Návrh prediktoru, předpovídání modelu, je uveden v sekci 2.1. V sekci 2.2 je potom rozpracován postup, jak kombinovat více prediktorů.

2.1 Odhadování přechodové pravděpodobnosti

Cílem agenta je předpovídat pravděpodobnosti přechodu od posledního stavu k dalšímu za podmínky určité akce v čase t . V sekci 1.2 jsme odvodili na čem je obecně závislý předpověďací model $P(d_t | d(t-1))$.

Podle (1.5) k výpočtu potřebujeme aposteriorní hustotu pravděpodobnosti, díky které lze kombinovat apriorní hustotu pravděpodobnosti a informaci obsaženou v pozorováních. Aposteriorní hustota pravděpodobnosti je potom ve tvaru (1.7).

Použitím řetězového pravidla (Věta 1.5) lze (1.7) rozepsat jako součin:

$$P(\theta | d(t)) \propto P(d(t) | \theta)P(\theta) = \prod_{k=1}^t P(d_k | d(t-1), \theta)P(\theta). \quad (2.1)$$

Podle předpokladu 1.2 uvažujeme systém s diskrétními stavy a akcemi. Data $d(t-1)$ v podmínce, lze za předpokladu konečné paměti 1.2 zapsat pomocí dříve zavedeného regresního vektoru r_{t-1} . V takovém případě jsou pravděpodobnostní funkce přechodu tabulky konečného rozměru. Takovou tabulku si můžeme představit jako vícerozměrné pole, jehož rozměr závisí na počtu regresorů. Neznámé, dle předpokladu časově neproměnné hodnoty jejich prvků, tvoří neznámý (mnohazměrový) parameter θ . V jistém stavu $s_t = s$ při jisté podmínce $a_t = a$ a $r_{t-1} = r$ z pravděpodobnostní

tabulky odhadují jistou hodnotu $\theta_{s|a,r}$, která musí být nezáporná a součet přes předpovídané stavy musí být jedna.

To lze interpretovat jako frekvence cest $r \rightarrow a \rightarrow s$, které byly měřeny do času t vůči celkovému počtu cest. Popis systému lze tedy parametrizovat pomocí hodnot pravděpodobnosti jako:

$$P(s_t | a_t, r_{t-1}, \theta) = \prod_{s,a,r} \theta_{s|a,r}^{\delta(s,s_t)\delta(a,a_t)\delta(r,r_{t-1})} \quad (2.2)$$

kde Kroneckerovy delty (1.1) zajišťují správnou volbu cesty (trojice s, a, r) podle které právě předpovídáme další stav. Například pro model závislý na jednom předchozím stavu $P(s_t | a_t, s_{t-1}, \theta)$ vnímáme pravděpodobnostní tabulku jako třírozměrné pole, ve kterém hledáme takovou matici, sloupec a řádek, které odpovídají konkrétnímu stavu a akci. Výpočet této tabulky se budeme snažit zjednodušit.

Použitím dané vlastnosti a vlastnosti součinu mocnin se stejným základem na (2.1) dostáváme:

$$P(\theta | d(t)) \propto \prod_{k=1}^t \prod_{s,a,r} \theta_{s|a,r}^{\delta(s,s_k)\delta(a,a_k)\delta(r,r_{k-1})} P(\theta) = \prod_{s,a,r} \theta_{s|a,r}^{\sum_{k=1}^t \delta(s,s_k)\delta(a,a_k)\delta(r,r_{k-1})} P(\theta). \quad (2.3)$$

Zavedeme pomocnou funkci (statistiku):

$$V_t(s, a, r) = V_{t-1}(s, a, r) + \delta(s, s_t)\delta(a, a_t)\delta(r, r_{t-1}). \quad (2.4)$$

Tato funkce (statistika) vyjadřuje počet cest $r \rightarrow a \rightarrow s$, které jsme dostali během pozorování do času t . Apriorní pravděpodobnost $P(\theta)$ vyjadřuje znalost o systému před počátkem sledování. Tuto znalost můžeme vyjádřit jako počáteční hodnotu statistiky $V_0(s, a, r) - 1$. Rozepíšeme (2.3) pomocí funkce V :

$$P(\theta | d(t)) \propto \prod_{s,a,r} \theta_{s|a,r}^{V_t(s,a,r)} \theta_{s|a,r}^{V_0(s,a,r)-1} = \prod_{s,a,r} \theta_{s|a,r}^{V_t(s,a,r)+V_0(s,a,r)-1}. \quad (2.5)$$

Věta 2.1 Pro systém s konečným počtem regresorů je aposteriorní hustota pravděpodobnosti $P(\theta | s_t, a_t, r_{t-1})$ v určitém čase t ve tvaru:

$$P(\theta | s_t, a_t, r_{t-1}) = \frac{\prod_{s,a,r} \theta_{s|a,r}^{V_t(s,a,r)+V_0(s,a,r)-1}}{\int_{\Theta} \prod_{s,a,r} \theta_{s|a,r}^{V_t(s,a,r)+V_0(s,a,r)-1} d\theta}, \quad (2.6)$$

kde množina Θ je definována jako:

$$\Theta = \left\{ \theta = (\theta_{s|a,r})_{s \in S, a \in A, r \in R} : \theta_{s|a,r} \geq 0, \sum_{s \in S} \theta_{s|a,r} = 1 \mid \forall a \in A, r \in R \right\}.$$

Potom integrační obor Θ je složitý simplex, díky čemuž jsou integrály beta funkce statistik parametrizujících integrovanou funkci.

Tímto způsobem lze obnovovat aposteriorní statistiky přepovídacího modelu v každém kroku předpovědi. Potom prediktor (1.5) je ve tvaru:

$$P(d_t | d(t-1)) \propto \int_{\Theta} P(s_t | a_t, d(t-1), \theta) P(a_t | d(t-1), \theta) \prod_{s,a,r} \theta_{s|a,r}^{V_{t-1}(s,a,r) + V_0(s,a,r) - 1} d\theta.$$

Pro generátor akcí platí předpoklad 1.6. Díky tomu je nezávislý na θ a můžeme ho psát před integrál. Uvažujeme opět diskrétní případ a tedy pomocí vlastnosti (2.2) můžeme přepsat pravděpodobnost $P(s_t | a_t, d(t-1), \theta) = P(s_t | a_t, r_{t-1}, \theta) = \theta_{s_t|a_t, r_{t-1}}$ pomocí pravděpodobnostní tabulky.

$$\begin{aligned} P(s_t, a_t | d(t-1)) &\propto P(a_t | d(t-1)) \int_{\Theta} \prod_{s,a,r} \theta_{s|a,r}^{V_t(s,a,r)} \theta_{s|a,r}^{V_{t-1}(s,a,r) + V_0(s,a,r) - 1} d\theta = \\ &= P(a_t | d(t-1)) \int_{\Theta} \prod_{s,a,r} \theta_{s|a,r}^{V_t(s,a,r) + V_0(s,a,r) - 1} d\theta. \end{aligned}$$

Použitím Beta funkce (1.2) přepíšeme model na tvar:

$$P(s_t, a_t | d(t-1), \theta) = P(a_t | d(t-1)) \frac{B(V_t(s, a, r) + V_0(s, a, r))}{B(V_{t-1}(s, a, r) + V_0(s, a, r))}.$$

Dále použitím věty 1.3 o vlastnosti B a Γ funkce dostáváme:

$$P(a_t | d(t-1)) \prod_{a,r} \frac{\prod_{s \in S} \Gamma(V_t(s, a, r) + V_0(s, a, r)) \Gamma(\sum_{s \in S} V_{t-1}(s, a, r) + V_0(s, a, r))}{\Gamma(\sum_{s \in S} V_t(s, a, r) + V_0(s, a, r)) \prod_{s \in S} \Gamma(V_{t-1}(s, a, r) + V_0(s, a, r))}.$$

Z definice funkce rozepíšme $V_t = \delta(s, s_t) \delta(a, a_t) \delta(r, r_t) + V_{t-1}(a, s, r)$ a použitím vlastnosti Γ funkce (1.3) z věty 1.4 dostáváme:

$$P(a_t | d(t-1)) \frac{V_{t-1}(s_t, a_t, r_{t-1})}{\sum_{s \in S} V_{t-1}(s, a_t, r_{t-1})}. \quad (2.7)$$

Tím máme obecně odvozenou pravděpodobnost přechodu. V uvažovaném diskrétním případě v následujícím odhadu lze integrály nahradit sumami. Pak platí následující věta:

Věta 2.2: Pro systém s konečným počtem regresorů lze pravděpodobnost přechodu ke stavu $s_t = s$ za podmínky akce $a_t = a$ a regresního vektoru $r_{t-1} = r$ a nasbírané statistiky $V = V_{t-1} + V_0$ vypočítat jako:

$$P(s | a, r, V) = \frac{V_{s|a,r}}{\sum_s V_{s|a,r}}. \quad (2.8)$$

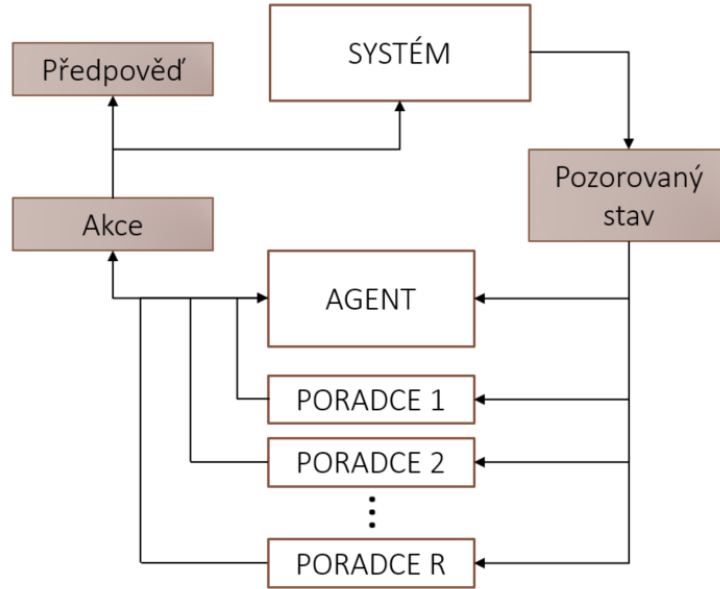
Podle této věty platí, že pravděpodobnost přechodů není nic jiného než počet posloupností dělený na pozorovanou četnost trojic (s, a, r) (opravenou apriorními představami) pro každý předpovídaný stav, což připomíná klasický výpočet pravděpodobnosti.

2.2 Kombinace prediktorů

Návrh metody spočívá v tom, že při odhadu pravděpodobnosti přechodu systému rozdělíme matematický model agentu na víc modelů, lišících se jejich regresory obsahujícími například různě zpožděné stavy či akce.

Předpovědácí model lze vytvořit ze dvou či více modelů: základní model agenta a další přídatné modely, pomocí kterých upravíme odhady parametrů základního modelu. Další prediktory vnímáme jako modely poradce podobně jako v sekci 1.2.

Při rozdělení předpokládáme, že modely se učí nezávisle na sobě, jinými slovy opravuji aposteriorní hustoty pravděpodobnosti podle těch regresorů, na kterých jsou závislé. Učení, což znamená upravení aposteriorní hustoty pravděpodobnosti v jistém čase t každého z prediktorů provedeme pomocí věty 2.1. Regresory lze rozdělit libovolně, modely můžou záviset na jednom (experimenty v kapitole 3) nebo i více regresorech. Na obrázku 2.1 je uveden algoritmus kombinování prediktorů pro $r \in \mathbb{N}$ poradců.



Obrázek 2.1: Algoritmus kombinaci prediktorů

Před výpočtem pravděpodobnosti přechodu chceme doplnit aposteriorní pravděpodobnost agenta pomocí dat poradců. Slučování dat provedeme pomocí věty 1.8 uvedené v sekci 1.2.:

$$P(\theta \mid d(t), M, v) \propto P(\theta \mid d(t)) \exp\left\{v \int_{\mathbf{s}} M(\mathbf{s} \mid \mathbf{z}_{t+1}) \ln(P(\mathbf{s} \mid \mathbf{a}_{t+1}, r_t, \theta)) d\mathbf{s}\right\}. \quad (2.9)$$

kde $M(\mathbf{s} \mid \mathbf{z})$ je dodatečný model poradce operující na svém regresoru \mathbf{z} , tj $M(\mathbf{s}_{t+1} \mid \mathbf{a}_{t+1}, d(t)) = M(\mathbf{s}_{t+1} \mid \mathbf{z}_{t+1})$, a v je váha důvěry k určitému poradci. Obecně \mathbf{z}_{t+1} odpovídá situaci, kdy agent užívá \mathbf{a}_{t+1}, r_t . Potřebujeme-li to zdůraznit píšeme místo \mathbf{z} symbol $\mathbf{z}(\mathbf{a}, r)$.

Slučování dat tak provedeme pro každý model poradce a díky tomu dostaneme novou aposteriorní pravděpodobnost parametrů modelu, který používá agent, která je upravená podle všech regresorů, na kterých je systém závislý.

Podle předpokladu 1.2 stav je diskretní veličina. Díky tomu integrál (2.9) se zredukuje na sumu přes předpovídaný stav. Pro přehlednost rozepíšeme vzorec podle předchozí teorie. Dosazením $P(\theta \mid d(t))$ ve tvaru jako (2.6) podle věty 2.1 a $P(s \mid a, r, \theta)$ z pravděpodobnostní tabulky analogicky jako (2.5) dostáváme:

$$P(\theta \mid d(t), M, v) \propto \prod_{s,a,r} \theta_{s|a,r}^{V_t(s,a,r)+V_0(s,a,r)-1} \exp \left\{ v \sum_{s \in \mathbf{S}} M(s \mid z_{t+1}(a, r)) \ln \left(\theta_{s|a_{t+1},r_t}^{\delta(a,a_{t+1})\delta(r,r_t)} \right) \right\}.$$

Pomocí základních vlastností logaritmu a vlastností mocnin spočítáme:

$$P(\theta \mid d(t), M, v) \propto \prod_{s,a,r} \theta_{s|a,r}^{V_t(s,a,r)+V_0(s,a,r)-1+vM(s|z(a,r))}.$$

Podle věty 2.2 model poradce v diskretním případě je

$$M(s \mid a, r) = \frac{W_{s|z}}{\sum_{s \in \mathbf{S}} W_{s|z}},$$

kde $W_{s|z}$ = počet pozorovaných hodnot $(s, z = z(a, r))$ vyskytujících se současně s výskyty (s, a, r) zvětšený o apriorní statistiku, kterou užil poradce. V podstatě je to stejný jako výpočet pravděpodobnosti v klasickém tvaru, kde pravděpodobnost je šance na uskutečnění jednoho nebo více jevů, vydělena počtem jevů možných.

A dosazením modelu $M(s \mid z(a, r))$ dostáváme výsledný tvar:

$$P(\theta \mid d(t), M, v) = \prod_{s,a,r} \theta_{s|a,r}^{V_t(s,a,r)+V_0(s,a,r)-1+v \frac{W_t(s,z(a,r))}{\sum_s W_t(s,z(a,r))}}.$$

Věta 2.3: Pro systém s konečným počtem stavu lze slučování dat M pro výpočet aposteriorní hustoty pravděpodobnosti $P(\theta \mid d(t), M, v)$ provést jako:

$$P(\theta \mid d(t), M, v) \propto \prod_{s,a,r} \theta_{s|a,r}^{V_t(s,a,r)+V_0(s,a,r)-1+v \frac{W_t(s,z(a,r))}{\sum_s W_t(s,z(a,r))}}.$$

Po výpočtu sdružené aposteriorní pravděpodobnosti, která obsahuje data každého poradce lze vypočítat společnou pravděpodobnost přechodu. Modely poradce a agenta jsou mnohem jednodušší díky tomu, že se nemusíme zabývat složitým propojením všech regresorů.

Pro model s konečným počtem regresorů s_i a a_j pro určité $i, j \in \mathbb{N}$ a s pravděpodobností přechodu definovanou jako $P(s_t, a_t \mid s_{t-1}, a_{t-1}, \dots)$ lze obecně rozměr N jedné tabulky spočítat jako:

$$N = \prod_{i,j} n_i \cdot m_j, \quad (2.10)$$

kde n_i je počet stavů s_i a m_j je počet akcí a_j .

Rozdělení modelů způsobí to, že nedostaneme jednu velkou pravděpodobnostní tabulku, ale více malých. Potom je zřejmé, že zavedeme-li více modelů rozměr tabulek se zmenší. Tím zmenšíme výpočetní náročnost složitých systémů.

Nevýhodou je, že předpověďací model nemůže zhodnotit míru vlivu poradce na kvalitu finální předpovědi. Proto do úpravy dat je zahrnutá váha důvěry v , což dovoluje nastavit kombinování modelu správně. Váhu můžeme nastavit tak, aby odpovídala vlivu regresorů na odhad předpovědi, díky čemu lze sestavit předpověďací model, aby nejlépe odpovídal skutečnosti.

V kapitole 3 jsou vyhodnocovány výsledky odhadu dalšího stavu pro různé hodnoty váhy. Potvrzují, že lze udělat také odhad nejlepší hodnoty váhy pro každý model poradce pomocí například bayesovského odhadování, avšak v této práci se tím zabývat nebudeme.

Kapitola 3

Experimenty

V dané kapitole jsou uvedeny simulace předpovídání pomocí metod popsaných v předchozích kapitolách. Cílem těchto simulací je ilustrovat použití předložené teorie a experimentálně otestovat navrženou metodu pro vyřešení problému s dimensionalitou.

Dále jsou uvedené tři experimenty provedené pro tři různé systémy. V sekci 3.1 jsou uvedeny základní předpoklady experimentů. V sekci 3.2 je popsána organizace výpočtu. V sekci 3.3 jsou uvedeny výsledky pro různé váhy poradce, na kterých je vidět, jak ovlivňují předpovídání dalšího stavu.

3.1 Základní předpoklady experimentů

V každém experimentu používáme metodu navrženou v kapitole 2. Experimenty jsou prováděny s délkou simulace $N = 10000$, aby se projevilo ustálené chování. Pro jednoduchost simulace budeme předpokládat, že reálné systémy jsou závislé pouze na dvou posledních stavech, tedy uvažujeme systémy s pamětí 2.

V každém experimentu předpokládáme, že se v systému vyskytují tři různé možné stavy a agent může volit tři různé možné akce. Dále se předpokládá rozdělení základního matematického modelu (1.4) na model agenta a modely poradců. Daný experiment ukazuje spolupráci agenta pouze s jedním poradcem.

Název	název v programu	hodnota
délka simulace	dur_simulation	10000
počet stavů	num_state	3
počet akcí	num_action	3
paměť systému	memory	2
počet poradců	num_adviser	1

Tabulka 3.1: Základní nastavení pro experimenty

Experiment je realizován pro různé váhy důvěry $v = 0.1 : 0.05 : 1$. Experiment

probíhal v prostředí Matlab, verze R2018b a pro opakovatelnost experimentu bylo použito společné počáteční $\text{seed} = 1$.

3.2 Organizace výpočtu

Experiment simulujeme cyklus učení a předpovídání agenta v systému. V této sekci bude popsáno využití teorie, která byla navrhována a popsána v kapitolách 1 a 2. Každý zkoumaný systém je zadán následující funkcí.

$$P(s_t | a_t, s_{t-1}, s_{t-2}) \propto \exp[-(s_t - m_t)^\delta \omega], \quad (3.1)$$

Regresní funkce m_t dovoluje zavést závislost na předchozích stavech. Parametry $\delta > 0$ a $\omega > 0$ nastavují míru koncentrace výsledné pravděpodobnosti kolem m_t . Složitější systémy potom můžou být nastavené se závislostí na více předchozích stavech. V našem případě regresní funkce m_t je lineární v pozorovaných datech a má paměť 2, tj. závisí na okamžité akci a dvou předchozích stavech a je zadána vztahem:

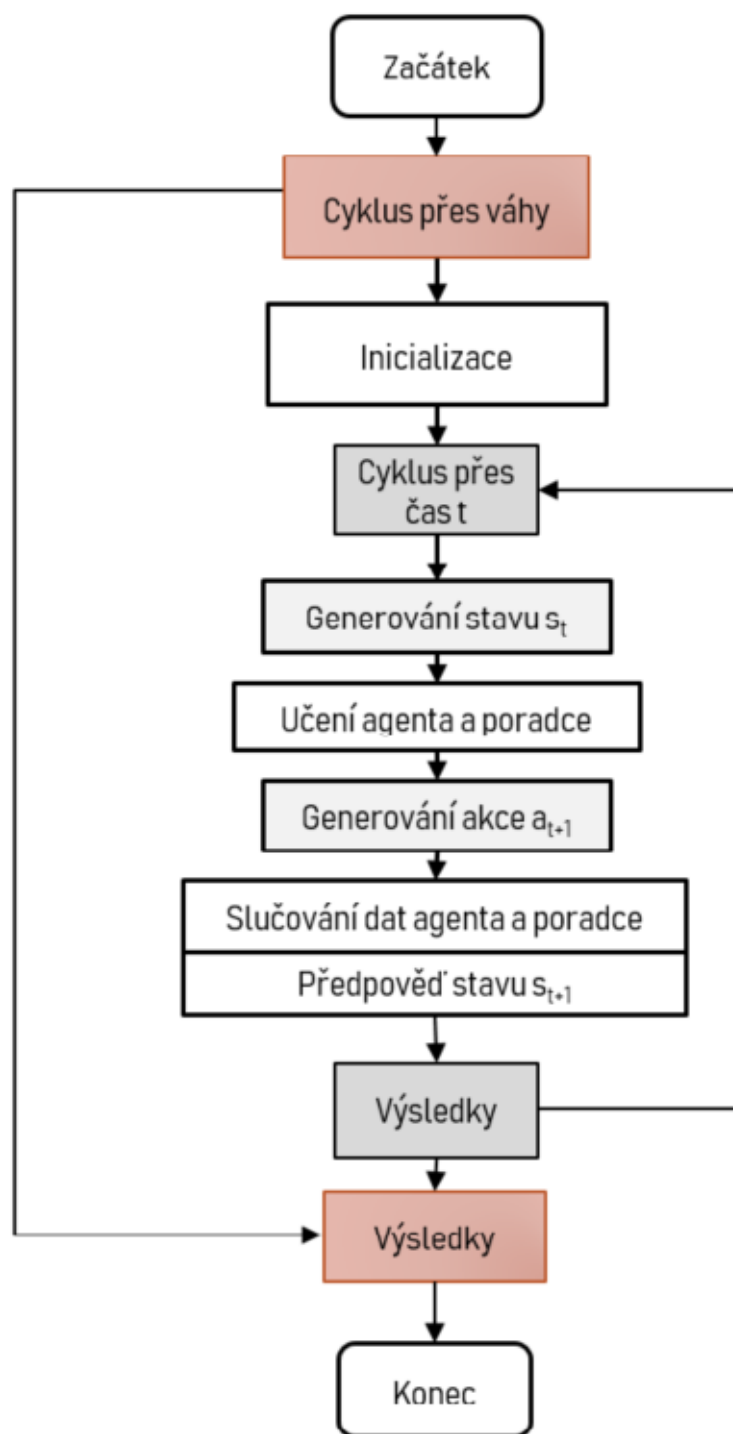
$$m_t = \alpha a_t + \beta s_{t-1} + \gamma s_{t-2} \quad (3.2)$$

Parametry α, β, γ stanovují míru závislosti systému na odpovídajícím regresoru. V tomto případě jsou pravděpodobnosti přechodu systému, které se snažíme odhadnout, ve tvaru 4-rozměrného pole, kde se pohybujeme po posloupnosti $s_{t-2} \rightarrow s_{t-1} \rightarrow a_t \rightarrow s_t$ v určitém čase t .

Normálně by předpověďovací model podle definice byl definován jako $P(s_t | a_t, s_{t-1}, s_{t-2})$. V takovém případě bychom museli odhadovat $3^4 = 81$ hodnot parametrů θ . Předpovídání rozdělíme na dva modely (model agenta + model poradce). Model agenta zavádíme ve tvaru $P(s_t | a_t, s_{t-1})$ a je závislý na okamžité akci a_t a regresoru s_{t-1} , model poradce ve tvaru $P(s_t | a_t, s_{t-2})$ je potom závislý na okamžité akci a_t a regresoru s_{t-2} . V takovém případě se počet odhadovaných hodnot rovná $2 \cdot 3^3 = 54$, což je o 27 prvků méně. V případě složitějších modelů tento rozdíl roste velmi prudce, jak je patrné z obecného vztahu (2.10).

Na obrázku 3.1. je uveden algoritmus simulace, podle které postupujeme v každém experimentu. Dále bude popsána každá jeho část.

- **Cyklus přes různé váhy** Dle teorie z sekce 2.2 lze slučování dat provádět s nastavením váhy důvěry vzhledem k modelu poradce. Díky tomu lze měnit její hodnotu a sledovat vliv poradce na agenta.
- **Inicializace** V této části proběhne inicializace všech struktur potřebných k simulaci. Pomocí funkcí `init_system`, `init_data` a `init_agent` inicializujeme 4 struktury. Funkce `init_system` zadává strukturu uvedenou v tabulce 3.2 popisující simulovaný systém:



Obrázek 3.1: Algoritmus simulace

system		
.num_state	počet stavů	3
.num_action	počet akcí	3
.dur_simulation	délka simulace	10000
.P_0	pravděpodobnost přechodu	4-rozměrné pole
.memory	paměť systému	2

Tabulka 3.2: Struktura popisující simulovaný systém

V jejím poli `P_0` jsou uloženy pravděpodobnosti přechodu v systému a jsou zadány pomocí vztahu (3.1). Jeho rozměr stanovuje paměť systému, resp. počet regresorů v systému.

Dále funkce `init_agent` inicializuje dvě struktury - strukturu agenta a poradce. Jsou to dvě stejné struktury s různým nastavením parametrů. Závislost na různých regresorech vyjadřuje proměnná `.dependence`.

agent nebo poradce		
.num_state	počet stavů	3
.num_action	počet akcí	3
.V_t	počet posloupností $s' \rightarrow a \rightarrow s$	pole 3x3x3
.V_0	počet posloupností $s' \rightarrow a \rightarrow s$ na začátku	pole 3x3x3
.model	prediktor	pole 3x3x3
.dependence	závislost na stavu	0 nebo 1
.des_rule	rozhodovací pravidlo	pole 3x3x3
.degrees_of_freedom	počet stupňů volnosti určující <code>V_0</code>	5

Tabulka 3.3: Struktura agent a adviser simulující chování agenta a poradce

`V_0` je pole pro ukládání počtu proběhlých posloupností $s' \rightarrow a \rightarrow s$ pozorovaných před začátkem simulace. Podle teorie to odpovídá datům potřebným k výpočtu apriorní hustoty pravděpodobnosti z věty 1.7. V programu je vygenerováno náhodně. `V_t` je pomocné pole pro ukládání počtu proběhlých posloupností během simulace, kde regresor $r = s'$ se mění podle toho, jestli mluvíme o agentovi nebo poradci. Pomocí těchto dat obnovujeme aposteriorní hustotu pravděpodobnosti v každé časové smyčce simulace podle věty 2.1.

Pole `model` odpovídá pravděpodobnostní tabulce předpovědicího modelu $P()$ agenta nebo poradce podobně jako (2.2). V této tabulce jsou vypočtené odpovídající aposteriorní hustoty pravděpodobnosti na θ pro bayesovský odhad reálných pravděpodobností přechodů systému dle věty 2.2. Pomocné pole `des_rule` obsahuje pravděpodobnosti výběru další akce agentem, které odpovídá generátoru akcí z předpokladu 1.6. V těchto experimentech je každá akce stejně pravděpodobná pro každý pozorovaný stav.

Poslední struktura data je inicializována funkcí `init_data` pro ukládání chování systému a předpovědi agenta.

data		
.state	proběhlé stavy	pole 1xdur_simulation
.action	proběhlé akce	pole 1xdur_simulation
.pred_state	stavy predikované agentem	pole 1 x dur_simulation
.pred_sstate	stavy predikované systémem	pole 1 x dur_simulation
.t	čas	3
.dur_simulation	délka simulace	10000

Tabulka 3.4: Struktura data pro ukládání chování systému a agenta

Simulace se ukládá od času $t = 3$, `state(1)`, `state(2)` a `action(1)` se vygenerují náhodně před simulací předpovídání.

- **Cyklus přes čas** Je to cyklus přes počet uzavřených smyček zadaných délkou simulace `dur_simulation` podle obrázku 1. Délku simulace lze nastavit. Dále budou ukázané výsledky pro délku simulace `dur_simulation = 10000`.
- **Generování stavu** probíhá pomocí funkce `dnoise`, která vybere další stav na základě pravděpodobnosti přechodu v závislosti na dvou předchozích stavech a akci. Vybraný stav se ukládá do pole `data.state`.
- **Učení agenta a poradce** Tady proběhne obnovení statistik V_t , kde V_t odpovídá funkci V_t (2.4). Podle věty 2.1 přepočítáme počet cest $s_{t-1} \rightarrow a_t \rightarrow s_t$ pro agenta a počet cest $s_{t-2} \rightarrow a_t \rightarrow s_t$ v čase t . Počet cest se obnovujeme dynamicky, a proto stačí přidat 1 na místo zpozorované nové cesty vzniklé novým stavem s_t , což odpovídá přepočtu statistiky V_t (2.4).

$$\begin{aligned} \text{agent.V}_t(s_t, a_t, s_{t-1}) &= \text{agent.V}_t(s_t, a_t, s_{t-1}) + 1; \\ \text{adviser.V}_t(s_t, a_t, s_{t-2}) &= \text{adviser.V}_t(s_t, a_t, s_{t-2}) + 1; \end{aligned}$$

- **Generování akce** Obecně akci vybírá agent podle toho, jaký další stav chce získat. Tomu odpovídá tzv. rozhodovací pravidlo. Náš případ nepředpokládá zkoumání preference agenta, proto jsou pravděpodobnosti výběru akcí umístěné v poli `des_rule` stejné pro každou akci. Tím zajistíme zcela náhodný výběr akcí. K tomu je použita funkce `dnoise`.
- **Slučování dat agenta a poradce** odpovídá teorii v sekci 2.2. Nejprve obnovujeme předpověď model poradce podle věty 2.2. $\text{adviser.V}_t + \text{adviser.V}_0$ odpovídá hodnotě W v (2.17) a hodnota modelu je potom:

$$\text{adviser.model}(:, a_t, s_{t-2}) = \frac{V_t(:, a_t, s_{t-2}) + V_0(:, a_t, s_{t-2})}{\sum_s V_t(:, a_t, s_{t-2}) + V_0(:, a_t, s_{t-2})} \quad (3.3)$$

Okamžitý model poradce použijeme podle věty 2.3. pro slučování dat. V našem případě se jedná o doplnění statistiky agent.V_t pomocí modelu poradce `adviser.model`:

$$\text{agent.V}_t(:, a_t, s_{t-1}) = \text{agent.V}_t(:, a_t, s_{t-1}) + v \cdot \text{adviser.model}(:, a_t, s_{t-2}) \quad (3.4)$$

To odpovídá výpočtu sdružené aposteriorní hustoty pravděpodobnosti z věty 2.1. Dále obnovíme model agenta podobně jako (3.3).

- **Předpověď agenta.** Na základě svého obnoveného modelu agent předpovídá další pozorovaný stav, který vygeneruje systém v dalším časovém cyklu. Jako predikovaný stav ukládám stav \hat{s}_{t+1} s největší pravděpodobností přechodu odhadnuté modelem.

$$\hat{s}_{t+1} = \max(\text{agent.model}(:, a_t, s_{t-1}))$$

Dále taky ukládám předpověď systému provedenou stejným způsobem na základě pravděpodobnosti P_0 .

- **Výsledky** V této sekci ukládám výsledky k hodnocení simulace (zpozorované stavy, akce, předpověď agenta a systému) pro zadanou váhu do speciálně připravených polí `save_states`, `save_actions`, `save_predstates`, `save_predsstates` ve stejném pořadí.

Dále vypočítávám počty chyb mezi predikovanými stavy a reálnými výskyty. Chybami nazýváme místa, kde předpovídaný stav neodpovídá reálnému stavu, který nastal v dalším kroku simulace. Taky počítám chyby mezi předpověďmi agenta a systému. Předpověď systému probíhá také podle maxima pravděpodobnosti přechodu a udává reálný počet chyb. Stane se to právě kvůli tomu, že systém nemusí generovat určitý stav, i když je nejvíce pravděpodobný. V následující sekci budou detailně popsány výsledky pro každý experiment.

3.3 Výsledky

V této sekci budou ukázány výsledky pro tři systémy EX1, EX2 a EX3 zadané pomocí různých parametrů. Zkoumáme tři různé systémy, kde pomocí regresního vektoru m_t (3.2) měníme parametry závislosti na akci a_t a stavech s_{t-1} , s_{t-2} . Vybrané parametry jsou uvedené v tabulce 3.5.

system	v programu	váha proměnné	EX1	EX2	EX3
alpha	α	a_t	0.3	0.3	0.3
beta	β	s_{t-1}	0.6	0.3	0
gamma	γ	s_{t-2}	0.2	0.3	0.5
delta	δ		2	2	2
omega	ω		1	1	1

Tabulka 3.5: Parametry systémů

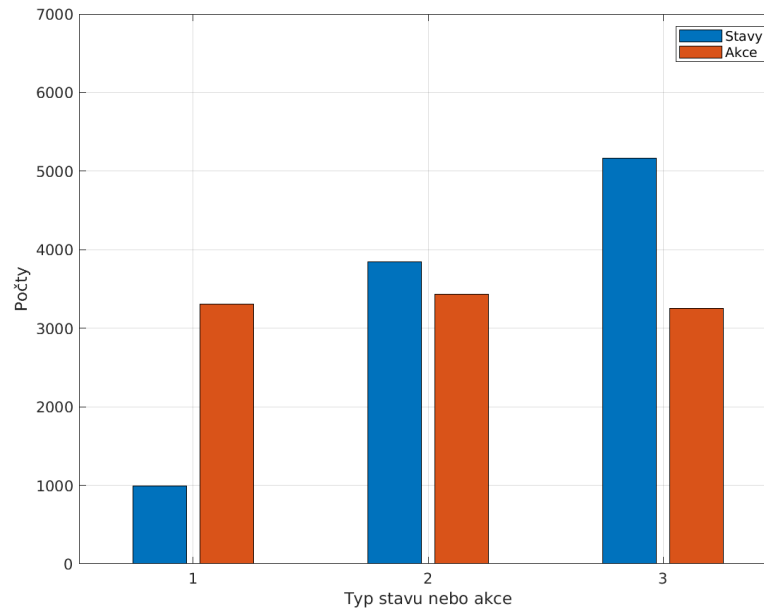
V prvním případě je systém nejvíce závislý na regresoru s_{t-1} , v druhém případě je závislost systému na regresorech s_{t-1} a s_{t-2} stejná. V třetím případě zkoumáme systém, který je nejvíce závislý na regresoru s_{t-2} . Systémy jsou vybrané tak, aby co nejlépe ukázaly chování algoritmu.

3.3.1 Systém EX1

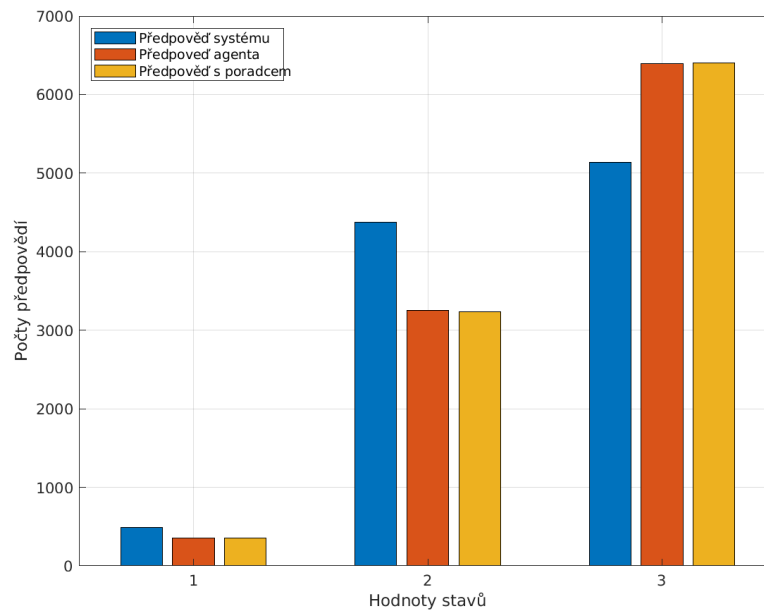
První systém EX1 je nejvíce závislý na stavu s_{t-1} . Obrázek 3.2 popisuje chování systému. Zobrazuje počty výskytu různých typů pozorovaných stavů a akcí, což

zadáva pravděpodobnost přechodu systému P_0 . V tomto případě se stav číslo 3 vyskytuje nejčastěji, avšak pravděpodobnost dosažení stavu číslo 2 je poměrně velká. Akci vybírá agent a je nastavena se stejnou pravděpodobností.

Podle sekce 3.2 agent odhaduje model, který je také závislý na stavu s_{t-1} . Díky tomu je předpovídání agenta samo o sobě poměrně spolehlivé. Na obrázku 3.3 sledujeme počty předpovědí každého typu stavu. Jako první je uvedena předpověď systému. Druhý sloupec ukazuje počet stavů s agentem bez poradce. Třetí sloupec udává předpověď agenta s maximální pomocí poradce, což znamená s vahou důvěry $v = 1$.



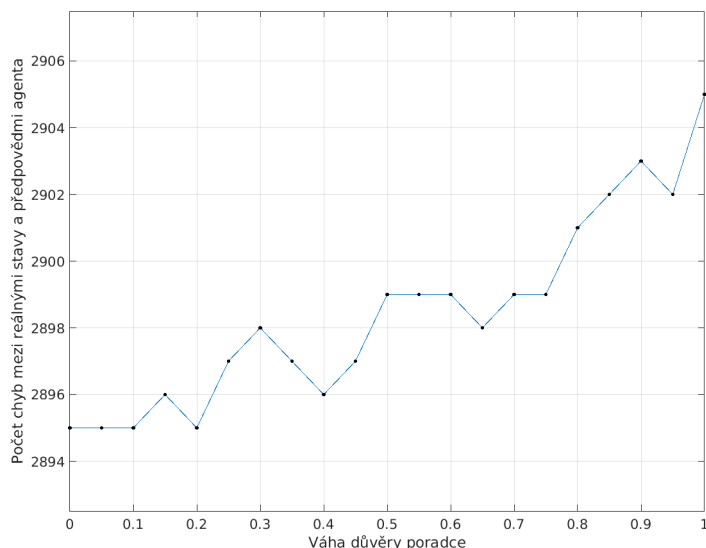
Obrázek 3.2: Počty stavů a akcí vyskytujících se v systému



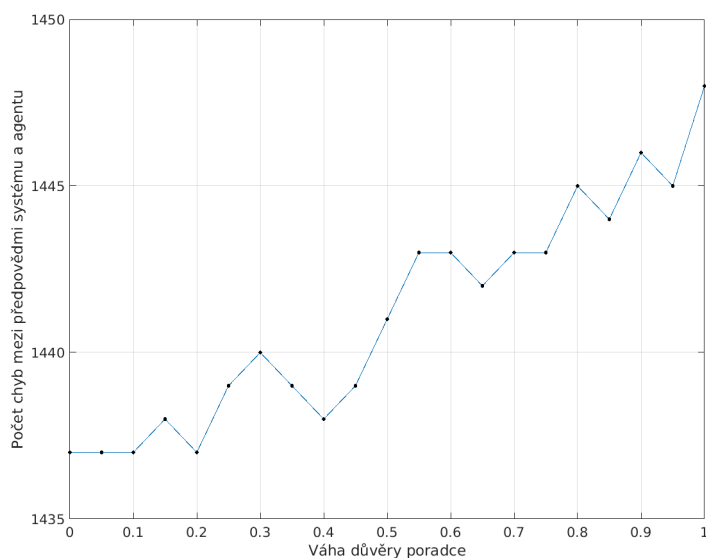
Obrázek 3.3: Počty předpovědí systému, agenta a agenta s poradcem

Dále je na obrázku 3.4. uveden graf závislosti počtu chyb na váze důvěry poradce. Chybou je nazývána každá předpověď, která neodpovídá stavu, který nastal reálně. Podle grafu se počet chyb zvětšuje s větší vahou poradce.

Na obrázku 3.5 je uveden graf závislosti počtu chyb mezi předpověďmi systému a agenta na váze důvěry poradce. Ten dává lepší přehled odhadu pravděpodobnosti přechodu agentem. I tady sledujeme podobnou tendenci, ovlivňování agenta poradcem vede k horším výsledkům.



Obrázek 3.4: Graf závislosti počtu chyb mezi pozorovanými stavy a předpověďmi agenta na váze důvěry k poradcovi



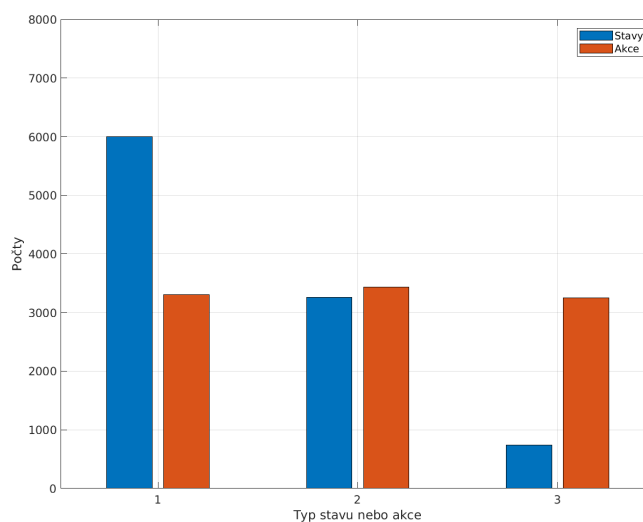
Obrázek 3.5: Graf závislosti počtu chyb mezi předpověďmi systému a agenta na váze důvěry k poradcovi

Diskuze

V tomto systému počet chyb roste s růstem váhy k poradcovi. Růst počtu chyb plyne z toho, že systém je málo závislý na regresoru, na kterém je závislý předpovídací model poradce. Není tedy potřeba zavádět model poradce. Je důležité stanovit závislost systému, zjistit vliv poradce na předpověď a správně nastavit váhu důvěry. Hlavním problémem je to, že tuto závislost neznáme předem. Tento problém je hlavní nevýhodou navržené metody. Avšak i v takových případech je vidět, že počet chyb roste jenom o maximálně 10-15 chyb.

3.3.2 Systém EX2

Systém EX2 závisí na každé proměnné stejně. V takových systémech očekáváme kladný vliv poradce na kvalitu předpovědi. Na obrázku 3.6 je uvedeno reálné chování systému podle nastavených pravděpodobností přechodů. V tomto systému se nejvíce vyskytuje stav číslo 3, potom číslo 2.

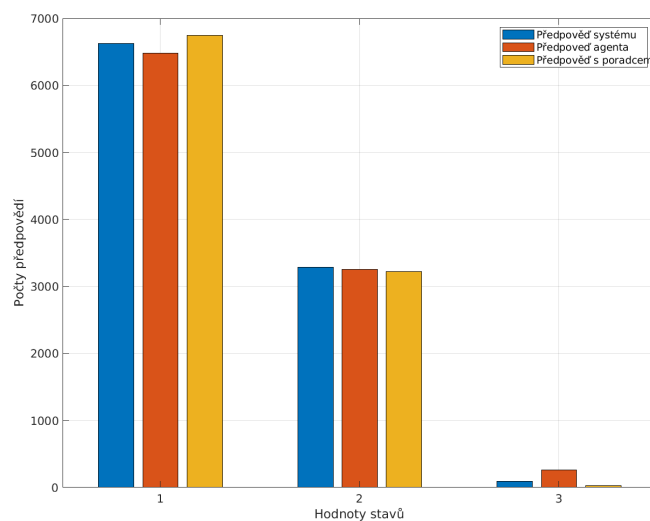


Obrázek 3.6: Počty stavů a akcí vyskytujících se v systému

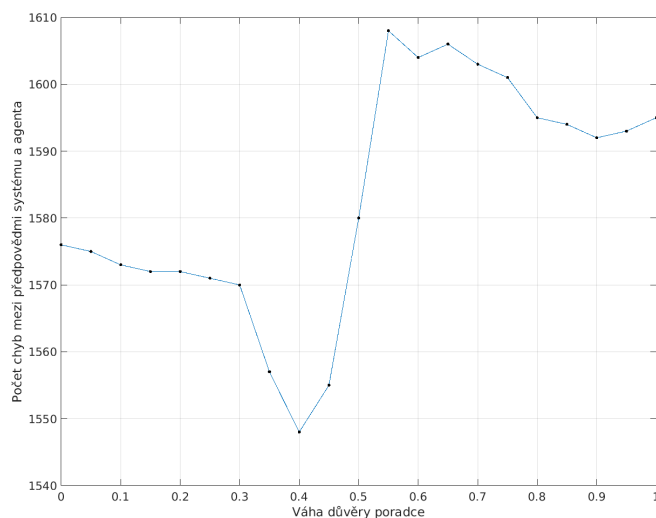
Obrázek 3.7 znázorňuje předpovědi systému, agenta a agenta s poradcem. Vidíme, že poradce snižuje hlavně počty předpovědi stavu číslo 3 a zvyšuje počty předpovědi stavu číslo 1, což je bližší k předpovědi systému.

Na obrázku 3.8 je uveden graf závislosti počtu chyb mezi předpověďmi agenta a reálnými pozorovanými stavy na váze důvěry poradce. Dále je na obrázku 3.9 uveden graf závislosti počtu chyb mezi předpověďmi agenta a systému na váze důvěry poradce. U prvního grafu opět sledujeme růst chyb, avšak druhý graf monotonní není.

Druhý graf ukazuje reálnou kvalitu odhadu pravděpodobnosti přechodu systému s pomocí poradce. Pro nějaké váhy důvěry je počet chyb menší, pro nějaké je větší. V bodě $v = 0.4$ je minimum.



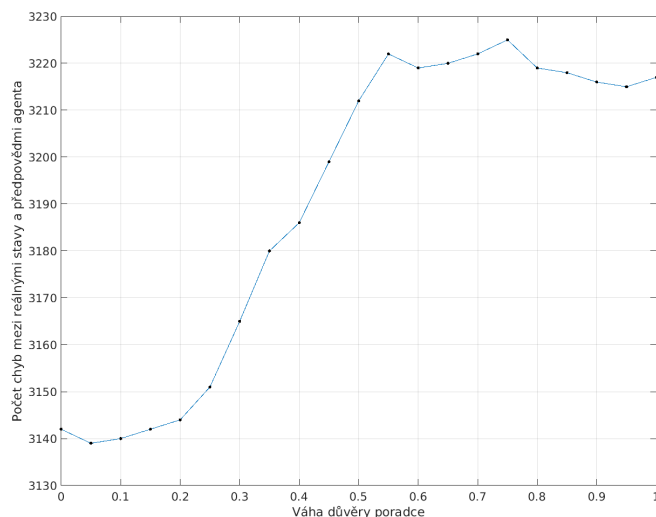
Obrázek 3.7: Počty předpovědí systému, agenta a agenta s poradcem



Obrázek 3.8: Graf závislosti počtu chyb mezi pozorovanými stavy a předpověďmi agenta na váze důvěry k poradcovi

Diskuze

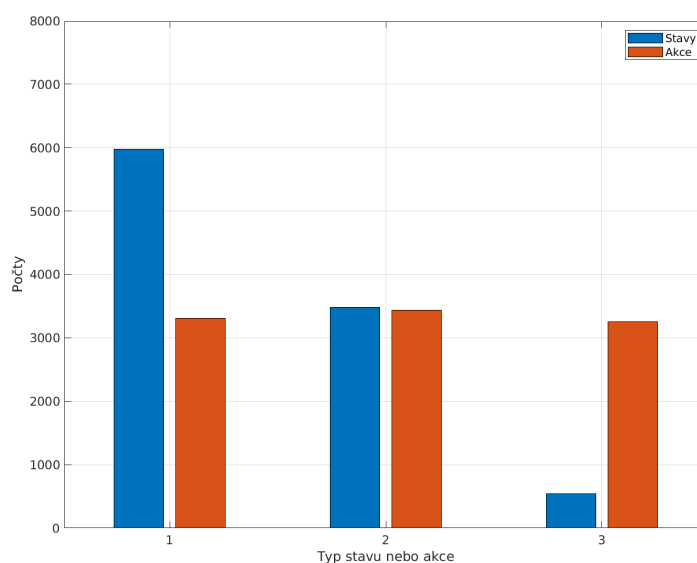
Takové chování vzniká kvůli podobné závislosti systému na obou regresorech s_{t-1} a s_{t-2} . Poradce zvyšuje pravděpodobnost přechodu k prvnímu stavu. V tomto případě poradce nejlépe pomáhá v odhadování pravděpodobnosti přechodu a opravuje nejvíce chyb pro váhu důvěry 0.4, větší váha potom zvyšuje počet chyb. Počet chyb v porovnání s reálnými výskyty roste kvůli výběru agentem nejpravděpodobnějšího stavu. Pro jiné systémy zadané podobným způsobem může být nejprínosnější váha jiná. V takových případech je vhodné odhadovat váhu důvěry pro maximální pomoc poradce. Odhad v takových případech je možné provést například pomocí bayesovského odhadování.



Obrázek 3.9: Graf závislosti počtu chyb mezi předpověďmi systému a agenta na váze důvěry k poradci

3.3.3 Systém EX3

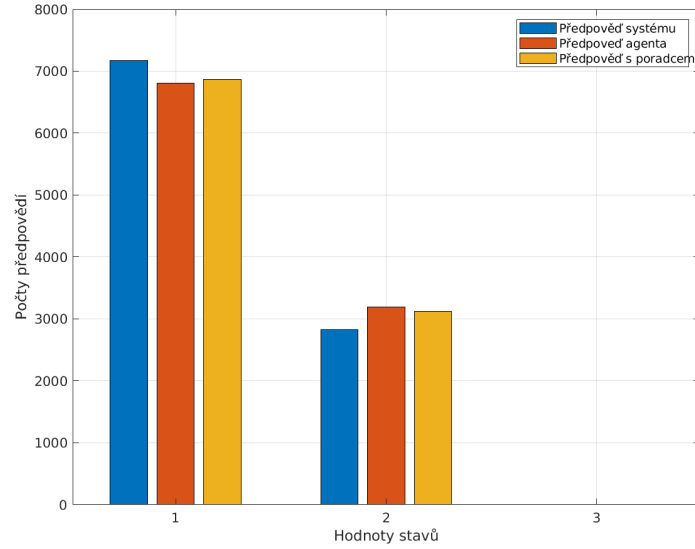
Systém EX3 je zadán pomocí parametrů tak, aby nejvíce závisel na stavu s_{t-2} . Díky tomu, že odhad poradce je zaměřený hlavně na odhadování tohoto stavu, je jeho model ve tvaru $P(s_t | a_t, s_{t-2})$, očekáváme větší vliv poradce na počet chyb vyskytujících se v simulaci.



Obrázek 3.10: Počty stavů a akcí vyskytujících se v systému

Na obrázku 3.10 sledujeme výskyt reálně pozorovaných stavů a akcí v simulaci systému. Výskyt stavu 3 je podle zadaných pravděpodobností přechodů málo pravděpodobný. V předpovědi systému na obrázku 3.11 se nevyskytuje vůbec, a proto se nevyskytuje ani v předpovědích agenta. Poradce ovlivňuje předpověď agenta a tím mu zvyšuje statistiky předpovědi stavu 3 a snižuje předpověď stavu 2.

V tabulce 3.4 jsou uvedené hodnoty pravděpodobností přechodu systému, následně modely agenta bez poradce a s maximální váhou poradce, $v = 1$ pro regresory $s_{t-2} = 2$ a $s_{t-1} = 2$. Sloupec odpovídá vybrané akci a_t a řádek předpovídanému stavu s_t . Například pro akci číslo 2 agent předpovídá stav číslo 1. Systém by naopak předpovídal stav číslo dva stejně jako poradce. Během simulace poradce zvyšuje pravděpodobnost druhého stavu a tím zlepšuje kvalitu odhadu.



Obrázek 3.11: Počty předpovědí systému, agenta a agenta s poradcem

system.P_0(:, :, 2, 2)

0.6622	0.3712	0.1510
0.2975	0.5538	0.7478
0.0403	0.0750	0.1012

adviser.model(:, :, 2)

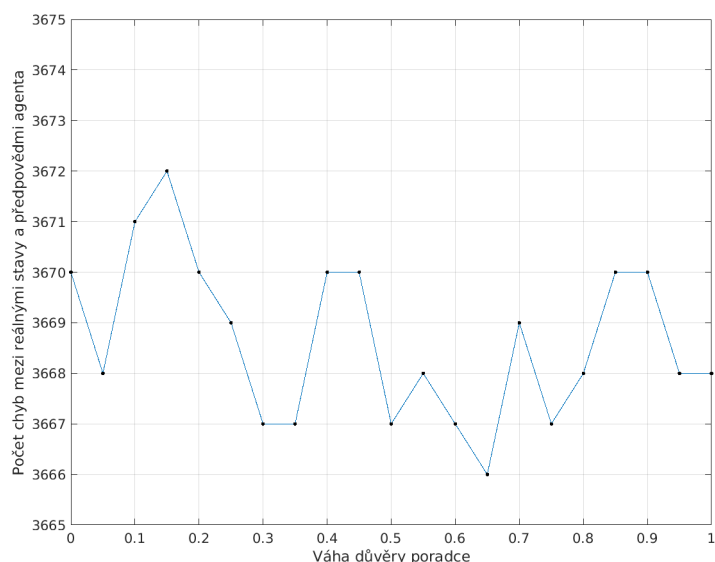
0.6730	0.3818	0.1425
0.2809	0.5382	0.7484
0.0461	0.0800	0.1092

agent.model(:, :, 2)

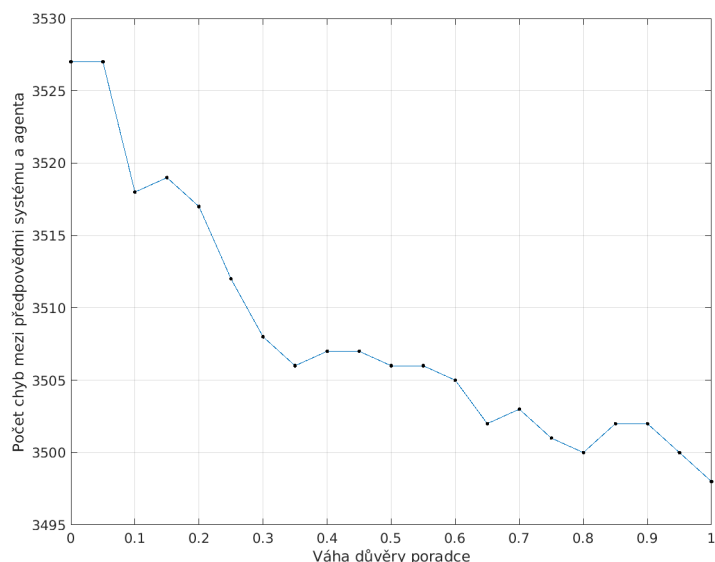
0.7480	0.6227	0.3985
0.2101	0.3212	0.5093
0.0419	0.0560	0.0922

Tabulka 3.6: Reálné pravděpodobnosti systému EX3 a odhady pravděpodobnosti agenta a poradce

Na obrázku 3.12 je uvedena závislost počtu chyb mezi předpověďmi agenta a reálnými stavy na váze důvěry. V tomto případě počet chyb opět roste. Na obrázku 3.13 je uveden graf závislosti počtu chyb mezi předpověďmi agenta a systému na váze důvěry poradce. Počet chyb s vyšší váhou důvěry poradce klesá, což znamená, že agent odhaduje pravděpodobnosti přechodu lépe.



Obrázek 3.12: Graf závislosti počtu chyb mezi pozorovanými stavy a předpověďmi agenta na váze důvěry k poradcovi



Obrázek 3.13: Graf závislosti počtu chyb mezi předpověďmi systému a agenta na váze důvěry k poradcovi

Diskuze

Výsledné grafy ukazují, že počet chyb v porovnání s předpověďmi agenta a reálnými výskyty se chová podobně pro každou váhu poradce. Pro některé váhy je ten počet větší, což se děje kvůli výskytu nepravděpodobných stavů. Tak například podle předpovědi systému stav číslo 3 nenastane, avšak ta pravděpodobnost nulová není, kvůli čemuž ho reálně pozorujeme.

Graf počtu chyb mezi předpověďmi systému a agenta ukazuje, že počet chyb pro maximální váhu klesá o 30 chyb, což v tomto případě není velkým rozdílem. Avšak podle tabulky 3.6 je odhad pravděpodobnosti přechodu poradcem dost přesný. Potom v takových případech musíme zaručit ještě větší vliv poradce na předpověď

nebo správně stanovit, který model odpovídá agentovi a který by měl být poradce.

3.3.4 Souhrnná diskuze

Zkoumala jsem tři různé případy, na kterých lze sledovat chování programu pro různé systémy. Každý systém byl závislý na dvou regresorech s_{t-1} a s_{t-2} a v každém případě jsem rozdělila předpověď model na agenta, který je závislý na regresoru s_{t-1} a poradce, se závislostí na regresoru s_{t-2} . V každém případě jsme zmenšili rozměrnost pravděpodobnostní tabulky z 81 hodnot na odhadování 54 hodnot.

Systém EX1 byl nastaven s malou závislostí na regresoru s_{t-2} . Výsledky potom ukazují pomalý růst počtu chyb s růstem váhy k modelu poradce. Takové chování je podmíněno slabou závislostí systému na parametru, který odhaduje model poradce. V takovém případě model agenta předpovídá další stav s dobrou kvalitou a zavedení modelu poradce je zbytečné.

Systém EX2 byl nastaven s podobnou závislostí na obou regresorech. V tomto případě se počet chyb nechová monotónně. Některé váhy důvěry k poradci vedou k lepším výsledkům, pro další váhy se výsledky zhoršují. Pro váhu 0.4 sledujeme nejlepší výsledek, proto pro každý systém musí být nalezená optimální hodnota váhy.

Systém EX3 byl nastaven s velkou závislostí na regresoru s_{t-2} . Poradce také odhaduje tento parametr. Avšak pomoc poradce nebyla tak význačná, i když s větší hodnotou váhy důvěry počet chyb v předpovědi prudce klesá. V takových případech musíme zaručit ještě větší vliv poradce na předpověď nebo správně stanovit, který model odpovídá agentovi a který poradci.

Ve většině systémů budeme muset správně stanovit závislost systému a najít takovou váhu, která poskytuje nejlepší pomoc poradce. Potom výpočet značně zjednodušíme bez velké ztráty kvality předpovídání.

Závěr

V této bakalářské práci jsem studovala teorii bayesovského odhadování parametrů a předpovídání. Dále jsem se zabývala návrhem prediktorů založených na odhadování parametrů pravděpodobnosti přechodu systému k dalšímu stavu na základě pozorované historie. Odvodila jsem vztahy pro diskrétní případ a pro systém s konečným počtem regresorů.

Dále jsem se zabývala metodou, která předpokládá užití více prediktorů vzniklých odhadováním jednoduchých parametrických modelů. Navržená metoda spočívá v rozdělení tohoto předpovídacího modelu na více jednodušších prediktorů, které vycházejí z odhadů jednoduchých parametrických modelů lišících se jejich regresory. Potom jsem využila vzorec převzatý z [4] pro kombinování takových prediktorů, kde jeden prediktor jsem používala jako předpovídací model agenta a další jako modely poradce.

Algoritmus byl zaměřen na vyřešení problému tzv. prokletí rozměrnosti vznikajícího během kvantitativního modelování složitých vzájemně propojených systémů. Pomocí této metody jsem zmenšila rozměrnost pravděpodobnostní tabulky vznikající při odhadování parametrů v diskrétních případech. Při správném stanovení závislosti systému a nalezení váhy, která poskytuje nejlepší pomoc poradce, jsem značně zjednodušila bez velké ztráty kvality předpovídání.

Nakonec jsem vybrala několik experimentů na různých systémech, které ilustrují danou teorii a simulovala jsem je pomocí programovacího prostředí Matlab. Zkoumané byly systémy s pamětí dva, což znamená, že jsou závislé na dvou regresorech. Systémy byly nastaveny s lineární závislostí na každém z regresorů. Předpovídací model jsem rozdělila do dvou menších prediktorů (agenta a poradce), které byly závislé na jednom z regresorů.

Každý systém se lišil svojí závislostí na regresorech. První zkoumaný systém byl málo závislý na regresoru, na kterém byl závislý poradce. Kvůli tomu data modelu poradce neovlivňovala předpověď, a tak počet chyb mezi předpověďmi agenta a reálnými stavy s růstem váhy důvěry k poradcovi rostl.

Další systém ukázal chování algoritmu při podobné závislosti na obou regresorech. V takovém případě je nutné najít optimální váhu důvěry k poradcovi. Potom poradce pozitivně pomáhá ovlivňovat předpověď.

Třetí systém byl nastaven s velkou závislostí na regresoru, na kterém byl také závislý poradce. V takovém případě pomoc modelu poradce zlepšuje výsledky předpovídání. Systémy 1 a 3 byly navrženy jako extrémní případy chování systému pro lepší po-

chopení teorie. Ve většině systémů musím správně stanovit závislost systému a najít takovou váhu, která poskytuje nejlepší pomoc poradce. Potom výpočet značně zjednodušíme bez velké ztráty kvality předpovídání.

To by mohlo být dalším krokem ke zlepšení algoritmu. Odhad optimální váhy důvěry pro konkrétní systém lze provést například na základě bayesovské statistiky. Do budoucna bych se chtěla více věnovat zlepšení metody a její implementaci na složitějších systémech, aplikovat teorii na komplexnější problémy. Chtěla bych se zaměřit na větší systémy a řešit pomocí této metody nějaký reálný případ.

Literatura

- [1] ALTMAN, N. a KRZYWINSKI, M. The curse(s) of dimensionality. *Nat Methods* 15, pp. 399–400, 2018.
- [2] ARTIN, E. The Gamma Function. HOLT, RINEHART AND WINSTON, 1931.
- [3] GAGNIUC, P. A. Markov chains: from Theory to Implementation and Experimentation. Hoboken: Wiley, 2017.
- [4] KRACÍK, J. a KÁRNÝ, M. Merging of data knowledge in Bayesian estimation. *Proceedings of the Second Int. Barcelona: Conference on Informatics in Control, Automation and Robotics, INSTICC, 2005*, pp. 229-232.
- [5] KIRKWOOD, J. R. Markov processes. Boca Raton, London, New York: CRC Press, Taylor and Francis Group, 2015.
- [6] LUO, L., WOOD, E.F. a PAN, M.. Bayesian merging of multiple climate model forecasts for seasonal hydrological predictions. *Journal of Geophysical Research, Atmospheres*, 2007.
- [7] ORLOFF, J. a BLOOM, J. Bayesian Updating with Continuous Priors. MIT OpenCourseWare, 2014.
- [8] PETERKA, V. Bayesian System Identification. In: *Trends and Progress in System Identification*. Oxford: Pergamon Press, 1981, pp. 239-304.
- [9] QUINN, A., KÁRNÝ, M. and GUY, T.V. Optimal design of priors constrained by external predictors. In: *Journal of Approximate Reasoning*. 2017, 84: pp. 150-158.
- [10] RISH, I. An empirical study of the naive Bayes classifier, T.J. Watson Research Center, 2001.
- [11] YE, H. a SUGIHARA, G. Information leverage in interconnected ecosystems: Overcoming the curse of dimensionality. *Science*, 2016, pp. 922-925.

Příloha A

Obsah přiloženého paměťového média

- Podlesna.pdf text bakalářské práce
- program adresář obsahující simulační program
 - main.m hlavní skript pro spouštění běhu programu
 - initialization.m funkce pro inicializace proměnných
 - init_system.m funkce pro inicializace systému
 - init_agent.m funkce pro inicializace agenta a poradce
 - init_data.m funkce pro inicializace struktury data
 - generate_state.m funkce pro generování stavu
 - learning.m funkce pro učení agenta nebo poradce
 - generate_action.m funkce pro generování akce
 - dnoise.m funkce pro generování stavu nebo akce na základě pravděpodobnosti
 - merging.m funkce pro slučování dat agenta a poradce
 - prediction.m funkce pro předpověď dalšího stavu