

Pokročilé statistické metody a jejich aplikace

(přednášky pro doktorandy)

Ivan Nagy

FD ČVUT, Praha

Obsah

1 Popis aplikace	5
2 Model a jeho odhad	10
2.1 Statické modely	10
2.2 Dynamické modely	22
2.3 Dodatky k odhadu	26
3 Predikce	32
3.1 Jednokroková predikce	32
3.2 Vícekroková predikce	34
3.3 Aplikace	40
4 Filtrace	42
4.1 Lineární model	42
4.2 Nelineární model	47
4.3 Aplikace	53
5 Testy hypotéz	68
5.1 Hypotézy o platnosti modelu	68
5.2 Aplikace	72

6	Logistická regrese	76
6.1	Aplikace	80
7	Odhad statické směsi	85
7.1	Odhad modelu ve tvaru směsi komponent	85
7.2	Aplikace	94
8	Odhad dynamické směsi	104
8.1	Odhad směsi s dynamickým ukazovátkem	104
8.2	Predikce aktivní komponenty	113
8.3	Aplikace	119
9	Přílohy	127
9.1	Speciální funkce	127
9.2	Doplnění na čtverec	129
9.3	Přirozené podmínky řízení	130
9.4	Bayesův vzorec	130
9.5	Multinomiální rozdělení	132
9.6	Dirichletovo rozdělení	133
9.7	Normální rozdělení	133
9.8	Inverzní Gauss-Wishartovo (GiW) rozdělení	134
9.9	Bodový odhad podle kvadratického kritéria	135
9.10	Bodové odhady parametrů spojitého modelu	136
9.11	Bodové odhady parametrů diskrétního modelu	137
9.12	Bodový odhad likelihoodu	138
9.13	Predikce s normálním modelem	140
9.14	Logistická regrese	141
9.15	Generování hodnot posunutého exponenciálního rozdělení	142
10	Elementární úlohy	144
10.1	Odhad parametrů normálního regresního modelu	144
10.2	Odhad parametrů regresního modelu s rovnoměrným šumem	145
10.3	Odhad modelu s exponenciálním rozdělením šumu	146

10.4	Odhad diskrétního modelu s kategorickým rozdělením	147
10.5	Off-line odhad parametrů modelu	148
10.6	Exponenciální zapomínání v odhadu parametrů modelu	150
10.7	Odhad struktury regresního modelu	151
11	Knihovna funkcí	153
11.1	c2c.m	153
11.2	catKL.m	154
11.3	drift.m	154
11.4	fnorm.m	155
11.5	GaussN.m	155
11.6	genps.m	156
11.7	logReg.m	156
11.8	lrLik.m	157
11.9	normKL.m	158
11.10	scal.m	158
11.11	seed.m	159
11.12	th2v.m	159
11.13	unscal.m	160
11.14	uut.m	160
11.15	v2thN.m	161
11.16	wflt.m	161

Úvod

Tento předmět těsně navazuje na předmět Stochastické systémy, přednášený na FD ČVUT, Praha v magisterském studiu. Zabývá se modelováním dynamických systémů za neurčitosti a na základě vytvořeného a odhadnutého modelu řeší cílové úlohy jako odhad hodnot neměřené veličiny, předpověď výstupu nebo klasifikace pracovních módů sledovaného systému. Teoretické základy pro řešení těchto úloh byly již shrnuty v předmětu Stochastické systémy. V tomto předmětu budou teoretické základy stručně zopakovány a rozšířeny. Hlavní důraz ale bude kladen na algoritmizaci a programování výsledků tak, aby posluchač byl schopen svůj problém správně posoudit a s pomocí uvedených programů i řešit. Při tom se ukáže, že samotné algoritmy pro sledované úlohy zdaleka nejsou to nejpodstatnější - ty jsou prostě odvozeny a připraveny k použití. Celá úloha má ale více částí, které vyžadují ani ne tolik matematiku jako spíše expertní (dopravní) znalost a zkušenosti. Celá úloha se tedy skládá z:

1. Formulace problému - vymezení systému co do veličin a jejich povahy (spojité \times diskrétní, měřené \times neměřené atd.) a odhadu jeho struktury (linearita, řád atd.)
2. Výběr vhodného modelu - pro začátek z těch, které budeme uvádět.
3. Odhad modelu
 - (a) inicializace odhadu na základě apriorních dat a expertní znalosti,
 - (b) vlastní odhad z měřených dat,
 - (c) validace odhadnutého modelu.
4. Řešení daného problému na základě matematického algoritmu.
5. Simulační validace celého řešení
 - (a) validace programového kódu,
 - (b) validace výsledků ve smyslu řešené úlohy.
6. Pokusné spuštění úlohy v reálném prostředí a její doladění.

Cílem tohoto předmětu bude právě příprava pro komplexní řešení takové úlohy. Důraz bude proto kladen nejen na teorii ale i na algoritmizaci a implementaci algoritmů v softwarovém prostředí jazyka Matlab a jejich správnou inicializaci.

Na úvod našeho výkladu o pokročilých statistických metodách a jejich praktickém využití připomeneme hlavní pojmy předmětu Stochastické systémy, který je přednášen v magisterském studiu a jehož znalost se zde předpokládá. Pro ty, kdo předmětem Stochastické systémy neprošli, je nutné, aby si texty k tomuto předmětu sami prostudovali. K dispozici jsou na webu na adrese

<http://www.fd.cvut.cz/personal/nagyivan/SSA/StSysTexty.pdf>

1 Popis aplikace

Hodiny: 1

Úloha levného a bezpečného řízení

Použitý přístup k řešení zvolené problematiky je datově závislý. Pro popis našeho systému řídič-automobil je použit regresní model, odhadovaný z měřených dat. Možnosti měření dat a jejich výběr pro model jsou proto prvořadým úkolem.

Data, která lze pro modelování využít, jsme získali od firmy Škoda Auto a.s. Na speciálně vybaveném vozidle Škoda Octavia je možno za jízdy měřit celou řadu veličin. Pro účely modelování je třeba vybrat:

1. modelované (výstupní) veličiny - tj. ty veličiny, které chceme ovlivňovat,
2. řídicí veličiny - tj. veličiny kterými systém ovlivňujeme,
3. ostatní veličiny - které nemůžeme přímo volit, ale které mají na modelování veličiny vliv.

Skupina 1. je jasná. Jsou to veličiny “spotřeba” a “rychlost”.

Do skupiny 2. zřejmě patří veličiny “plyn”, “brzda” a “rychlostní stupeň”, protože těmito veličinami je automobil přímo řízen. Nicméně, při rozumné (šetřivé) jízdě se brzda prakticky průběžně nepoužívá a když se použije, tak na úplné dobrzdění. To ale není předmětem optimalizace ale určitá deterministická akce, vynucená okolnostmi nebí koncem jízdy. V této skupině tedy zůstává pouze “plyn” a “rychlostní stupeň”.

Z ostatních veličin, patřících do skupiny 3. zvolíme “moment motoru”, vyvolaný plynem a mající přímý vliv na spotřebu, “otáčky motoru”, které těsně souvisí se zvoleným rychlostním stupněm a “ujetá dráha”, která je velmi těsně svázána s rychlostí automobilu (jinak dosti obtížně modelovanou).

Cílem úlohy je provést syntézu optimálního řízení realizovaného veličinami “plyn” a “rychlostní stupeň”, které minimalizuje odchylky modelovaných veličin od žádaných hodnot - nulové nebo zadané minimální spotřeby a doporučené rychlosti. Vzhledem k potřebné stabilitě řešení je pro syntézu řízení použita metoda dynamického programování na konečném horizontu řízení a žádaná hodnota pro řízené veličiny je uvažována dopředu známá a její budoucí hodnoty jsou využity pro okamžité řízení - tzv. řízení s předprogramováním.

Abychom mohli využít metodu dynamického programování, potřebujeme nejprve model řízené soustavy.

Model systému řidič-automobil

Pro účely modelování uvažujeme veličiny jako náhodné posloupnosti v diskretním čase. Značíme: y_t - výstup soustavy v čase t (tvořen spotřebou a rychlostí), u_t - řízení v čase t (plyn a rychlostním stupněm) a v_t - externí veličiny v čase t (moment a otáčky motoru, ujetá dráha). Symbolem $d_t = \{y_t, u_t, v_t\}$ značíme všechna data v čase t , tj. a symbolem $d(t) = \{d_0, d_1, \dots, d_t\}$ všechna data od začátku modelování, včetně apriorní informace označené d_0 .

Jako model využijeme normální lineární regresní model

$$y_t = b_0 u_t + c_0 v_t + \sum_{i=1}^n (a_i y_{t-i} + b_i u_{t-i} + c_i v_{t-i}) + k + e_t$$

kde n je řád modelu, e_t je i.i.d. náhodná posloupnost s normálním rozdělením, nulovou střední hodnotou a konstantním rozptylem r . k je konstanta modelu.

Parametry a_i, b_i, c_i a r jsou odhadovány z měřených dat. Nejprve je však třeba určit vhodný řád modelu (tzv. odhad struktury modelu). Ten odhadneme z apriorních (předem změřených) dat poměrně snadno tak, že všechny veličiny normujeme tak aby měly nulovou střední hodnotu a jednotkový rozptyl, provedeme odhad modelu (např. metodou nejmenších čtverců) a velikosti odhadnutých koeficientů porovnáme s předem stanovenou hladinou (např. 10% z největšího koeficientu). Do modelu vezmeme jen ty veličiny, které odpovídají parametrům s větší velikostí než je stanovená reference, případně vezmeme model takového řádu, aby byly pokryty všechny významné zpožděné hodnoty výstupu.

Levné a bezpečné řízení

S navrženým, z apriorních dat přednastaveným a dále průběžně odhadovaným modelem můžeme přejít k syntéze řízení. Tato syntéza je navrhována tak, aby minimalizovala střední hodnotu kvadratického kritéria

$$Q = \min_{U \in U^*} E \left[\sum_{t=1}^N Q_t | d_0 \right]$$

$$Q_t = (y_t - s_t)' \omega (y_t - s_t) + (u_t - u_{t-1})' \lambda (u_t - u_{t-1})$$

kde N je horizont řízení, $U = \{u_1, u_2, \dots, u_N\}$ je řízení na intervalu $t = 1, 2, \dots, N$, s_t je žádaný průběh výstupní veličiny y_t a penalizace řídicí veličiny je uvažována v přírůstkovém tvaru, aby se zabránilo trvalému off-setu řízené veličiny.

Řízení na konečném horizontu se uvažuje proto, že jsou dobře známy potíže se stabilitou tzv. jednokrokového řízení, které plánuje řídicí veličinu vždy jen pro budoucí časový okamžik. Řízení je pak navrženo tak, aby v jednom kroku udělalo co nejvíce a může soustavu excitovat natolik, že ji v dalších krocích už nelze uřídit. Dalším důvodem pro řízení na konečném horizontu je skutečnost, že budoucí cesta automobilu je nám známá

a proto můžeme i žádané hodnoty modelovaných veličin nastavit dopředu a jejich znalost v syntéze řízení využít. Provádíme pak syntézu řízení s předprogramováním. Při ní se např. řídicí algoritmus dozví o zatáčce již předem a může postupně brzdit tak, aby jeho jízda byla šetřivá. To je v našem případě velice žádoucí.

Celý princip optimalizace je postaven na vytváření kompromisu mezi levnou jízdou a dobou jízdy. Kdybychom požadovali jen levnou jízdu, bylo by řešení triviální - nikam nejezdit. Protože cestu chceme absolvovat, stanovíme určitou doporučenou rychlost, kterou se má automobil na cestě pohybovat. V našich dosavadních úvahách byla tato doporučená rychlost konstruována expertně z mapy plánované trasy. V dalším výzkumu by tato doporučená rychlost měla být modelována a navrhována na základě údajů z navigace. Řídicí algoritmus minimalizující zvolené kritérium pak optimalizuje řízení tak, aby se automobil pohyboval pokud možno blízko doporučené rychlosti a všude tam, kde je to výhodné šetřil palivo i za cenu snížení rychlosti. Hlavními úsporami při řízení může být (i) postupné zpomalování před zatáčkou, (ii) postupné rozjíždění za zatáčkou, (iii) rozjetí vozidla před kopcem, (iv) jízda bez plynu z kopce. To jsou hlavní principy šetřivé jízdy, které řidiči většinou znají. My však doufáme, že takových situací, ve kterých je možno šetřit, je daleko více a že budou při optimálním řízení objeveny a využity.

Jak jsme se již zmínili, model požitý pro syntézu řízení musí být apriori přednastaven. Nicméně nic nebrání tomu, aby se jeho vlastnosti nezlepšovaly další průběžnou identifikací. Tady by se ale v optimálním případě jednalo o současný odhad a řízení, což je úloha duálního programování, která není rozumně řešitelná. Proto je využita suboptimální metoda tzv. ustupujícího horizontu. Při ní se postupuje takto: V daném časovém okamžiku se s existujícími odhady parametrů provede syntéza na celém řídicím intervalu. Parametry jsou konstantní. Výpočet řídicí veličiny se počítá od konce intervalu směrem dopředu. Z celého řízení se ale použije jen jeho aktuální hodnota (tedy naposledy vypočtená) a ta se aplikuje. Změří se nový výstup, a s ním se provede výpočet nového odhadu parametrů. Řídicí interval se posune o krok dopředu. Tato procedura se pak stále opakuje.

Pro vlastní realizaci algoritmu adaptivního řízení se využívá metoda iterací rozprostřených v čase. Při této metodě se na začátku startuje výpočet optimálního řízení na konečném horizontu tak, jak plyne z teorie - tedy z nulových hodnot. Poté, co se posune horizont o krok a řízení se počítá znova se však nezačíná z nuly, ale z hodnoty (tedy matice), ke které se došlo v kroku minulém. Je známo, že výpočet řízení na konečném horizontu se postupně ustaluje a správně by se měl horizont volit tak, aby došlo k úplnému ustálení. Při této metodě však lze horizont volit poměrně krátký a k ustalování dochází postupně při řízení v čase - odtud název metody. Použití metody iterací rozprostřených v čase velmi zrychlí výpočty a dovolí adaptivní řízení realizovat ve reálném čase.

Výpočet optimálního řízení na konečném horizontu se provádí podle následujícího algoritmu:

```

 $\varphi_{N+1}^* = 0$ 
for  $t = N, N - 1, \dots, 1$  do

     $\varphi_t = E [\varphi_{t+1}^* + Q_t | u_t, d(t-1)]$ 
     $\varphi_t^* = \min_{u_t} \varphi_t, \quad u_t^* = \arg \min_{u_t} \varphi_t$ 

end

```

kde φ_t je tzv Bellmanova funkce, která představuje hodnotu optimalizovaného kritéria v čase t po předchozí optimalizaci, a φ_t^* je hodnota kritéria po minimalizaci v čase t (zbytek kritéria, který přechází do dalších kroků minimalizace). Jak je naznačeno, výpočty probíhají od konce intervalu řízení. V každém kroku se spočte optimální řízení, které ale nelze okamžitě realizovat - data, která jsou zapotřebí jsou ještě neznámá. Teprve když dojdeme na začátek intervalu, můžeme realizovat první řízení, aplikovat, změřit výstup atd. Jádrem algoritmu je výpočet střední hodnoty za pomocí modelu řízení soustavy v prvním kroku a minimalizace vystředovaného kritéria v kroku druhém. Tyto dva kroky středování a minimalizace se stále opakují.

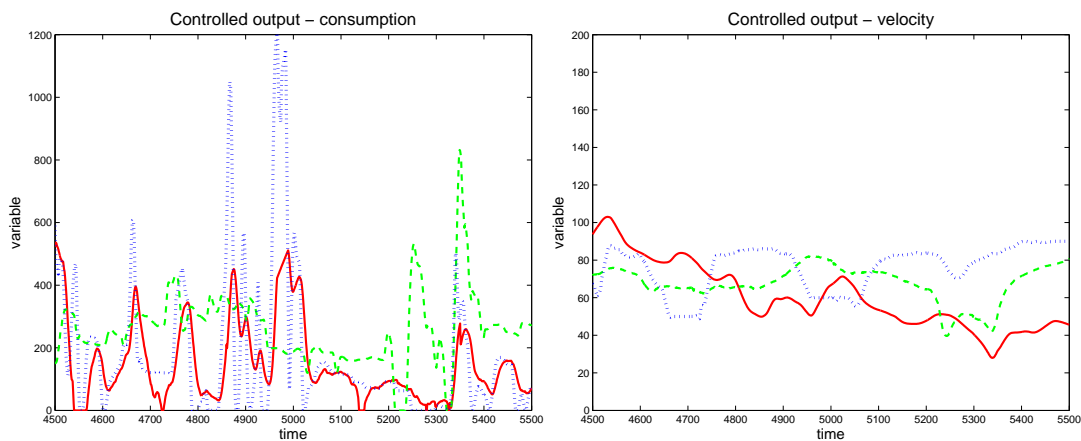
Logický blok řízení

Optimální řízení je jistě žádoucí, ale jestliže optimální rychlost je 55 km/hod a maximální povolená je 50 km/hod, budeme těžko policejní kontrole něco vysvětlovat o optimalitě a podobně. Proto je třeba automatické řízení zastřešit blokem logických kontrol, případně některých deterministických zásahů do řízení. Kromě kontroly maximální povolené rychlosti sem patří například také kontrola, zda je při jízdě z kopce zařazen neutrální, nebo signalizace přechodu pro chodce či řízené křižovatky. Tento blok je nadřazen automatickému řízení a má přední prioritou.

Výsledky simulačních experimentů

Experimenty s navrženým řídicím algoritmem byly prováděny pro data naměřená na zvoleném testovacím okruhu, dlouhém přibližně 40 km a nacházejícím se poblíž Mladé Boleslavi. Tato cesta vede nejprve po dálnici, dále pokračuje po silnici I. třídy a její poslední část vede po silnici II. třídy a prochází také několika obcemi. Tak měřená data zahrnují všechny hlavní typy jízdy a zaručují tak reprezentativnost měřeného vzorku. K dispozici jsou data z osmi jízd po tomto okruhu a to počínaje jízdou klidnou (šetřivou) až po jízdu živou (sportovní). Tato data slouží jednak jako apriorní data a jednak pro porovnání výsledků jízdy s automatickým řízením. Podle těchto dat a podle expertní zkušenosti byla také zkonstruována doporučená rychlost pro celý testovací okruh.

Pro účely testování řídicího algoritmu byl zkonstruován softwarový simulátor spotřeby automobilu. Tento simulátor je postaven na základě základních fyzikálních vztahů mezi veličinami, které mají vliv na spotřebu automobilu. Simulátor byl testován na reálných datech a jeho kvalita byla více než dobrá. Samozřejmě, že konečné výsledky je možno

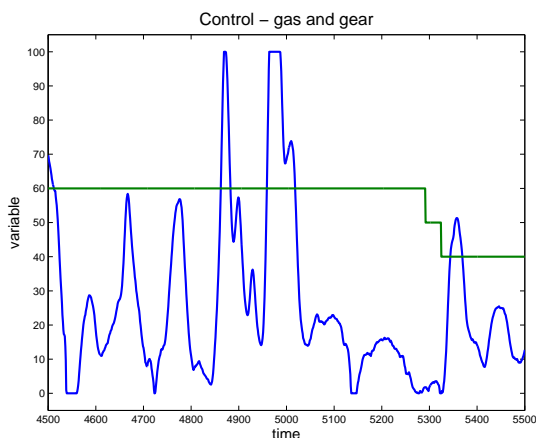


Obrázek 1.1: Řízený výstup - spotřeba (vlevo), rychlost (vpravo)

potvrdit až při testech na reálném automobilu. Na tomto ověření se pracuje, zatím však provedeno nebylo.

Pro experimenty byl použit regresní model 2. řádu. Tento model byl off-line přednastaven na vzorku apriorních dat o velikosti 1000 měření. Perioda snímání dat je 5 měření za vteřinu. Apriorní data tedy zahrnují dobu přibližně 3.3 minuty. Důležitou charakteristikou řízení je použitý koeficient penalizace v kritériu řízení. Penalizovány jsou odchylky spotřeby a rychlosti od jejich požadovaných průběhů a změny řídicích veličin plynu a rychlostního stupně. Penalizace výstupů byly stejné, rovny jedné. Penalizace změn řízení byla zvolena 1000, Tato velká penalizace řízení má za cíl zabránit prudkým změnám v hodnotách řízení, protože se ale jedná o změny, nevnaší do řízení trvalý offset. Pro výpočet řízení byla zvolena délka horizontu pět, což dovoluje použití metody iterací rozprostřených v čase. Doba výpočtu řízení pro 75000 vzorků je 8 vteřin.

Na obrázku 1.1, v jeho levé části, je zobrazena řízená veličina “spotřeba” (plná červená čára) spolu se spotřebou, která byla za stejných podmínek dosažena při jízdě, při které se prováděla měření - jedná se o jednu ze šetřivých jízd (modrá tečkovaná čára). Pro informaci je ještě zobrazen průběh žádané spotřeby, která se rovná 85% spotřeby z průměrných hodnot z apriorních měření (zelená čárkovaná čára). V pravé části obrázku 1.1 je stejnými typy čar zobrazena veličina “rychlost” - tedy rychlost řízená (modrá), dosažená při měření (modrá) a referenční (zelená). Obrázky ukazují část jízdy obsahující 1000 vzorků (tedy opět asi 3.3 minuty) po silnici I. třídy. Delší úsek jízdy dává nepřehledné grafy a jízda po dálnici je poměrně nezajímavá. Vybraný úsek ukazuje podstatné vlastnosti řízení. Z grafů je patrné, že hlavní úsporou při jízdě je omezení špiček ve spotřebě, které se dosahuje šetrným zacházením s plynem a celkově o něco málo pomalejší jízdou.



Obrázek 1.2: Řízení - plyn a rychlostní stupeň

Pro zajímavost, na obrázku 1.2 jsou zobrazeny řídicí veličiny na stejném úseku cesty kterou ukazuje předchozí graf. Modrá čára je veličina “plyn” (procento stlačení plynového pedálu) a “rychlostní stupeň” (desetinásobek zařazeného rychlostního stupně - pro viditelnost grafu spolu s plynem). Vidíme, že převážně se jede na šestý stupeň, jen na konci zobrazované periody se řadilo níže. Vzhledem ke značné síle motoru vozu Škoda Octavia se ale většinu cesty jede na šestku.

2 Model a jeho odhad

Hodiny: 2, 3; 4, 5, 6

Program: simulace a odhad (základní modely), vývoj aposteriorní, zapomínání, odhad struktury.

Mezi úlohy odhadu patří odhad parametrů modelu, klasifikace a predikce. Při klasifikaci přidělujeme změřený datový vzorek do určité třídy vzorků (například zda řidič jede normálně nebo divoce), při predikci odhadujeme hodnotu určité veličiny v budoucím okamžiku (například jízda “bezpečná”, “riskantní”, “havárie”). Pro klasifikaci většinou používáme statické modely (modelovaný výstup nezávisí na svých minulých hodnotách), úloha predikce vyžaduje model dynamický.

2.1 Statické modely

Statický model má obecný tvar

$$f(y_t|\Theta), \text{ nebo s řízením } f(y_t|u_t, \Theta)$$

a lze jej vyjádřit pomocí rovnice

$$y_t = bu_t + k + e_t$$

kde y_t je modelovaná veličina, u_t je řízení, e_t je šum s rozptylem r a b , k jsou regresní koeficienty.

Nejvyužívanější je normální regresní model, kde šum má normální rozdělení. Dále předvedeme i modely pro některá další rozdělení šumu. Důležitou úlohu zde hraje tzv. exponenciální třída rozdělení, pro kterou lze jednoduše provést odhad parametrů modelu.

MODEL Z EXPONENCIÁLNÍ TŘÍDY ROZDĚLENÍ

Řekneme, že rozdělení náhodné veličiny patří do exponenciální třídy, jestliže jej lze vyjádřit ve tvaru

$$f(y|\theta) = \exp\{Q(\theta)T(y) + G(\theta) + H(y)\}$$

a množina y , pro kterou je $f(y|\theta) \neq 0$ je nezávislý na θ .

Příklady exponenciálních rozdělení

- exponenciální rozdělení

$$f(y|a) = a \exp\{-ay\} = \exp\{-ay + \ln(a)\} \quad (2.1)$$

- alternativní rozdělení

$$\begin{aligned} f(y|p) &= p^y (1-p)^{1-y} = \exp\{y \ln(p) + (1-y) \ln(1-p)\} = \\ &= \exp\left\{\ln\left(\frac{p}{1-p}\right) y + \ln(1-p)\right\} \end{aligned}$$

- Poissonovo rozdělení

$$\begin{aligned} f(y|\lambda) &= \exp\{-\lambda\} \frac{\lambda^y}{y!} = \exp\{\ln(\lambda^y) - \lambda - \ln(y!)\} = \\ &= \exp\{y \ln(\lambda) - \lambda - \ln(y!)\} \end{aligned}$$

- normální rozdělení

$$\begin{aligned} f(y|\mu, r) &= \frac{1}{\sqrt{2\pi r}} \exp\left\{-\frac{1}{2r}(y-\mu)^2\right\} = \exp\left\{-\frac{1}{2r}(y^2 - 2\mu y + \mu^2) - \frac{1}{2} \ln(2\pi r)\right\} = \\ &= \exp\left\{-\frac{1}{2r}y^2 + \frac{\mu}{r}y + \frac{\mu^2}{2r} - \frac{1}{2} \ln(2\pi) - \frac{1}{2} \ln(r)\right\} = \\ &= \exp\left\{\begin{bmatrix} -\frac{1}{2r}, & \frac{\mu}{r}, & \frac{\mu^2}{2r} \end{bmatrix} \begin{bmatrix} y^2 \\ y \\ 1 \end{bmatrix} - \frac{1}{2} \ln(r) - \frac{1}{2} \ln(2\pi)\right\}. \end{aligned}$$

Likelihood

Pro modely z exponenciální třídy je snadné spočítat likelihood. Ten je definován jako součin modelů s dosazenými měřenými daty

$$L_N(\theta) = \prod_{t=1}^N f(y_t|\theta).$$

Pro exponenciální třídu to bude

$$\begin{aligned} L_N(\theta) &= \prod_{t=1}^N \exp\{Q(\theta)T(y_t) + G(\theta) + H(y_t)\} = \\ &= \exp\left\{Q(\theta)\sum_{t=1}^N T(y_t) + NG(\theta) + \sum_{t=1}^N H(y_t)\right\} = \\ &= \exp\{Q(\theta)S_N + NG(\theta) + V_N\}. \end{aligned}$$

Odhad parametru

Ukážeme odhad metodou ML (maximum likelihood), kdy bodový odhad parametru leží v maximu likelihoodu. Maximum hledáme pomocí derivace $L'_N(\theta) = 0$:

$$L'_N(\theta) = \exp\{Q(\theta)S_N + NG(\theta) + V_N\} (Q'(\theta)S_N + NG'(\theta)) = 0,$$

kde

$$S_N = \sum_{t=1}^N T(y_t), \quad V_N = \sum_{t=1}^N H(y_t) \quad (2.2)$$

a S_N a N jsou statistiky odhadu.

Odtud plyne podmínka pro maximum

$$Q'(\theta)S_N + NG'(\theta) = 0.$$

Např. pro exponenciální rozdělení $f(y_t|a) = a \exp\{ay_t\}$ je $\hat{a} = \frac{1}{\bar{y}}$, kde \bar{y} je průměr z naměřených hodnot výstupu.

Příklad odhadu parametru exponenciálního rozdělení

Odhad parametru rozdělení z exponenciální třídy ukážeme na příkladě:

Uvažujme rozdělení

$$f(y_t|a) = ay_t^{a-1}, \quad \text{pro } y_t \in (0, 1), \quad \text{a } a > 0.$$

Toto rozdělení patří do exponenciální třídy s parametry: $Q = a$, $T = \ln y_t$, $G = \ln a$ a $H = -\ln y_t$.

Spočteme: $Q' = 1$, $S_N = \sum_{t=1}^N \ln y_t$ a $G' = \frac{1}{a}$.

Dosadíme do rovnice a dostaneme

$$1 \cdot \sum_{t=1}^N \ln y_t + N \frac{1}{a} = 0.$$

Odtud

$$\hat{a} = \frac{N}{\sum_{t=1}^N \ln y_t} = \frac{1}{\overline{\ln y_t}}.$$

což je převrácená hodnota průměru logaritmů dat.

MODEL S POISSONOVÝM ROZDĚLENÍM

Jak jsme již dříve ukázali, Poissonovo rozdělení patří do exponenciální třídy. Ukážeme zde, jak se provádí jeho odhad.

Model pro výstup y_t s parametrem λ je dán následující hp

$$f(y_t|\lambda) = \exp\{-\lambda\} \frac{\lambda^{y_t}}{y_t!}.$$

Likelihood je dán součinem modelů

$$L_N = \prod_{t=1}^N \exp\{-\lambda\} \frac{\lambda^{y_t}}{y_t!} = \exp\{-N\lambda\} \frac{\lambda^{\sum_{t=1}^N y_t}}{\prod_{t=1}^N y_t!}.$$

Odtud snadno určíme jak a posteriori hp, tak i přepoččet statistik.

A posteriori hp

$$f(\lambda|d(t)) \propto \exp\{-\kappa_t \lambda\} \lambda^{S_t},$$

kde statistiky κ_t a S_t se počítají podle následujících vztahů

$$\begin{aligned} \kappa_t &= \kappa_{t-1} + 1, \\ S_t &= S_{t-1} + y_t, \end{aligned}$$

s počátečními hodnotami κ_0 a S_0 .

Bodový odhad parametru λ je

$$\hat{\lambda} = \frac{S_N}{\kappa_N} = \bar{y},$$

což je výběrový průměr modelované veličiny y_t ve změřených hodnotách. Vzorec dostaneme jako argument maxima aposteriorní hp. Derivace je

$$-\kappa_N \exp\{-\kappa_N \lambda\} \lambda^{S_N} + \exp\{-\kappa_N \lambda\} S_N \lambda^{S_N-1} = 0 \quad \text{a odtud}$$

$$\kappa_N \lambda = S_N.$$

MODEL S EXPONENCIÁLNÍM ROZDĚLENÍM

Model má tvar

$$f(y_t|a) = a \exp\{-ay_t\}.$$

Likelihood

$$L_N = a^N \exp\left\{-a \sum_{t=1}^N y_t\right\}.$$

Podle tvaru likelihoodu volíme tvar apriorní hp odkud dostaneme aposteriorní hp ve tvaru

$$f(a|d(t)) = a^{\kappa_t} \exp\{-aS_t\},$$

kde statistiky se přepočítávají následujícím způsobem

$$\kappa_t = \kappa_{t-1} + 1,$$

$$S_t = S_{t-1} + y_t.$$

Bodový odhad parametru a dostaneme opět jako argument maxima aposteriorní hp ve tvaru

$$\hat{a} = \frac{\kappa_N}{S_N} = \frac{1}{\bar{y}}$$

jako převrácenou hodnotu výběrového průměru.

KATEGORICKÝ (MULTINOMIÁLNÍ, DISKRÉTNÍ) MODEL

Je model diskrétní modelované veličiny s kategoričným (někdy též nazývaným multinomiálním) rozdělením

$$\frac{y_t}{f(y_t)} \left| \begin{array}{cccc} 1 & 2 & \cdots & m \\ p_1 & p_2 & & p_m \end{array} \right.$$

kde p_1, p_2, \dots, p_m jsou pravděpodobnosti, a tedy platí

$$p_i \geq 0, i = 1, 2, \dots, m, \quad \text{a} \quad \sum_{i=1}^m p_i = 1.$$

Model ve tvaru hp je

$$f(y_t|p) = p_{y_t}$$

a tuto hp vyjádříme tabulkou, která přímo odpovídá uvedenému rozdělení

$$\frac{y_t = 1 \quad y_t = 2 \quad \cdots \quad y_t = m}{p_1 \quad p_2 \quad \cdots \quad p_m}$$

Statistika ν_t pro odhad parametru p má stejný tvar, jako tabulka modelu

$$\frac{y_t = 1 \quad y_t = 2 \quad \cdots \quad y_t = m}{\nu_{1;t} \quad \nu_{2;t} \quad \cdots \quad \nu_{m;t}}$$

a její přepočítání se provádí podle vzorce

$$\nu_{i;t} = \nu_{i;t-1} + \delta(i, y_t),$$

tedy k prvku ν_{y_t} se přidá jednička (prvky počítají, kolikrát byla která hodnota výstupu).

Bodový odhad parametru p dostaneme normalizací statistiky tak, aby součet jejích prvků byl roven jedné, tj.

$$\hat{p} = \frac{\nu_t}{\sum_{i=1}^m \nu_{i;t}}.$$

STATICKÝ NORMÁLNÍ REGRESNÍ MODEL

Rovnice, kterou je tento model zadán je

$$y_t = k + e_t, \tag{2.3}$$

kde

$$e_t \sim N(0, r) = \frac{1}{\sqrt{2\pi r}} \exp \left\{ -\frac{1}{2r} e_t^2 \right\}.$$

Rovnice (2.3) je transformační rovnicí pro náhodnou veličinu y_t . Příslušnou transformací dostaneme rozdělení závislé na parametrech $\Theta = \{k, r\}$

$$\begin{aligned} f(y_t|\Theta) &= \frac{1}{\sqrt{2\pi r}} \exp\left\{-\frac{1}{2r}(y_t - k)^2\right\} = \\ &= \frac{1}{\sqrt{2\pi r}} \exp\left\{-\frac{1}{2r}[-1, k] \begin{pmatrix} y_t \\ 1 \end{pmatrix} [y_t, 1] \begin{pmatrix} -1 \\ k \end{pmatrix}\right\} = \\ &= \frac{1}{\sqrt{2\pi r}} \exp\left\{-\frac{1}{2r}[-1, k] D_t \begin{pmatrix} -1 \\ k \end{pmatrix}\right\}, \end{aligned}$$

kde

$$D_t = \begin{pmatrix} y_t \\ 1 \end{pmatrix} [y_t, 1]$$

je datová matice v kroku t .

Podle tvaru modelu zvolíme aposteriorní hp

$$f(\Theta|d(t)) \propto r^{\kappa_t} \exp\left\{-\frac{1}{2r}[-1, k] V_t \begin{pmatrix} -1 \\ k \end{pmatrix}\right\},$$

kde κ_t a V_t jsou statistiky odhadu parametrů.

Přepočítání statistik se děje podle vzorců

$$V_t = V_{t-1} + D_t, \quad \kappa_t = \kappa_{t-1} + 1$$

a tento přepočítání startuje s apriorními statistikami κ_0 a V_0 .

Poznámky

1. V případě modelu řízené soustavy bude rovnice modelu

$$y_t = bu_t + k + e_t \rightarrow e_t = -[-1, b, k] [y_t, u_t, 1]'$$

a podle toho vypadá i aposteriorní hp a přepočítání statistik.

2. V případě mnohorozměrného modelu, kdy např. $y_t = [y_{1;t}, y_{2;t}]'$ bude

$$\begin{bmatrix} y_{1;t} \\ y_{2;t} \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} + \begin{bmatrix} e_{1;t} \\ e_{2;t} \end{bmatrix} \quad \begin{bmatrix} e_{1;t} \\ e_{2;t} \end{bmatrix} = - \begin{bmatrix} -1 & 0 & k_1 \\ 0 & -1 & k_2 \end{bmatrix} \begin{bmatrix} y_{1;t} \\ y_{2;t} \\ 1 \end{bmatrix}$$

a přepočítání informační matice

$$V_t = V_{t-1} + \begin{bmatrix} y_{1;t} \\ y_{2;t} \\ 1 \end{bmatrix} [y_{1;t}, y_{2;t}, 1].$$

Odtud také plyne, že po rozdělení této matice bude mít submatice V_y rozměr 2×2 . Ostatní submatice budou s ní kompatibilní.

ROVNOMĚRNÉ ROZDĚLENÍ

Rovnoměrné rozdělení na intervalu (b_L, b_U) s hp $1/(b_U - b_L)$ pro $b_L < y_t < b_U$ a jinde nula je asi nejjednodušším rozdělením. Jeho charakteristika je: (1) všechny hodnoty z intervalu (b_L, b_U) jsou stejně pravděpodobné (nemáme žádné preference), (2) mimo interval (b_L, b_U) jsou hodnoty nemožné (pro hodnoty existují ostré hranice).

Přes jednoduchost rovnoměrného rozdělení není práce s ním příliš jednoduchá. Důvodem je to, že nepatří do exponenciální třídy rozdělení, protože hranice jeho oboru nenulovosti závisí na jeho parametrech (jsou přímo jeho parametry). V následujícím textu ukážeme, jak se s ním pracuje. Metoda odhadu je jednoduchá pro statický model, kterým se budeme zabývat. V případě dynamického modelu vede úloha odhadu na metodu lineárního programování.

Model

Uvažovaný model má tvar

$$f(y_t | b_L, b_U) = \frac{1}{b_U - b_L} \chi(y_t > b_L) \chi(y_t < b_U),$$

kde χ je indikátorová funkce podmínky (rovná se jedné, kde je podmínka splněna, a nule jinde).

Likelihood

Likelihood dostaneme jako součin modelů s postupně dosazenými daty y_t , $t = 1, 2, \dots, N$.

$$\begin{aligned} L_N &= \prod_{t=1}^N \frac{1}{b_U - b_L} \chi(y_t > b_L) \chi(y_t < b_U) = \\ &= \left(\frac{1}{b_U - b_L} \right)^N \prod_{t=1}^N \chi(y_t > b_L) \prod_{t=1}^N \chi(y_t < b_U) = \\ &= \left(\frac{1}{b_U - b_L} \right)^N \chi \left(b_U > \max \{y_t\}_{t=0}^N \right) \chi \left(b_L < \min \{y_t\}_{t=0}^N \right). \end{aligned}$$

Apriorní hp

Apriorní hp určíme na základě tvaru likelihoodu

$$f(b_L, b_U | d(0)) \propto \left(\frac{1}{b_U - b_L} \right)^{\kappa_0} \chi(b_U > U_0) \chi(b_L < L_0),$$

kde κ_0 , L_0 a U_0 jsou apriorní statistiky s významem: počet apriorních dat a dolní a horní hranice intervalu rovnoměrného rozdělení.

Aposteriorní hp

Aposteriorní hp je součin likelihoodu a apriorní hp

$$\mathcal{P} = f(b_L, b_U | d(N)) \propto \left(\frac{1}{b_U - b_L} \right)^{\kappa_N} \chi(b_U > U_N) \chi(b_L < L_N). \quad (2.4)$$

Přepočítání statistik

Pro postupný (on-line) výpočet statistik je možno použít následující rekurzivní vztahy

$$\kappa_t = \kappa_{t-1} + 1, \quad L_t = \min \{L_{t-1}, y_t\}, \quad U_t = \max \{U_{t-1}, y_t\}.$$

Bodové odhady (MP)

Vzhledem k tomu, že rovnoměrné rozdělení nepatří do exponenciální třídy, bodové odhady parametrů b_L a b_U se počítají poněkud uměle.

Hledáme maximum \mathcal{P} . Přitom to bude tím větší, čím jsou b_L a b_U blíže (viz (2.4)). Snažíme se tedy posunout b_L co nejvíce doprava a b_U zase doleva. Přitom narazíme na podmínky $y_t > L_N$ a $y_t < U_N$. Pro bodové odhady tedy platí

$$\hat{b}_L = L_N = \min \{y_t\}_{t=0}^N \quad \text{a} \quad \hat{b}_U = U_N = \max \{y_t\}_{t=0}^N.$$

POSUNUTÉ EXPONENCIÁLNÍ ROZDĚLENÍ

Exponenciální rozdělení s hp $a \exp\{-ay_t\}$ pro $y_t \geq 0$ je ukázkovým příkladem exponenciální třídy rozdělení (*ref*). Nicméně, chceme-li zobecnit na posunuté rozdělení $a \exp\{-a(y_t - A)\}$ pro $y_t \geq A$ dostaneme se mimo exponenciální třídu rozdělení. Důvodem je závislost oboru nenulovosti této hp na parametru A . Okamžitě přestávají platit všechny hezké a jednoduché vztahy, které obecně platí právě pro exponenciální třídu. Naštěstí lze i v tomto případě postupovat dosti rozumně. Dále budeme tento postup demonstrovat.

Model

Jak jsme již uvedli, uvažovaný model má tvar

$$f(y_t|a, A) = a \exp\{-a(y_t - A)\}, \quad y_t \geq A.$$

Likelihood

Likelihood dostaneme jako součin modelů s postupně dosazenými daty y_t , $t = 1, 2, \dots, N$. Protože obor nenulovosti modelu je závislý na parametru, vstupuje také do likelihoodu

$$L_N = \prod_{t=1}^N a \exp\{-a(y_t - A)\}, \quad y_1, y_2, \dots, y_N \geq A,$$

$$L_N = a^N \exp\left\{-a\left(\sum_{t=1}^N y_t - NA\right)\right\}, \quad \min_{t=1:N} y_t \geq A.$$

Prior hp

Apriorní hp určíme na základě tvaru likelihoodu

$$f(a, A|d(0)) \propto a^{\kappa_0} \exp\{-a(S_0 - NA)\},$$

kde κ_0 a S_0 jsou apriorní statistiky s významem: počet apriorních dat a součet apriorních dat.

Posterior hp

Aposteriorní hp je součin likelihoodu a apriorní hp

$$\mathcal{P} = f(a, A|d(N)) \propto a^{\kappa_N} \exp\{-a(S_N - NA)\}.$$

Přepočítání statistik

Pro postupný (on-line) výpočet statistik je možno použít následující rekurzivní vztahy

$$S_t = S_{t-1} + y_t, \quad \kappa_t = \kappa_{t-1} + 1.$$

Bodové odhady (MP)

Vzhledem k tomu, že posunuté exponenciální rozdělení nepatří do exponenciální třídy, bodové odhady parametrů a a A se počítají trochu uměle.

Maximum \mathcal{P} podle A spočteme následující úvahou:

$$\mathcal{P} = a^{\kappa_N} \exp\{-a(S_N - NA)\} = a^{\kappa_N} \exp\{-aS_N\} \exp\{aNA\}$$

Máme-li zvolit A tak, aby \mathcal{P} bylo co největší, musíme volit $A \rightarrow \infty$. Jsme ale omezeni podmínkou $\min\{y\} > A$. Proto volíme $A = \min\{y\}$.

Maximum \mathcal{P} podle a počítáme pomocí derivace, tj. tak, jak je to obvyklé v exponenciální třídě

$$\partial_a \mathcal{P} = \kappa_N a^{\kappa_N - 1} \exp\{-a(S_N - NA)\} - a^{\kappa_N} (S_N - NA) \exp\{-a(S_N - NA)\} = 0,$$

$$\begin{aligned} \kappa_N &= a(S_N - NA), \\ a &= \frac{\kappa_N}{S_N - NA} = \frac{1}{\bar{y} - \frac{N}{\kappa_N} A} \doteq \frac{1}{\bar{y} - A} = \frac{1}{\bar{y} - \min\{y\}}. \end{aligned}$$

Odtud dostáváme bodové odhady

$$\hat{a}_N \doteq \frac{1}{\bar{y} - \min\{y\}} \quad \text{a} \quad \hat{A}_N = \min\{y\}.$$

LOGISTICKÝ MODEL

Speciální forma modelu, který nepatří do exponenciální třídy modelů, dokonce není ani lineární, je model logistický. Nelze pro něho použít běžné bayesovské postupy a jeho odhad se provádí numericky metodou ML (maximum likelihood).

Význam logistického modelu spočívá v tom, že popisuje závislost diskrétní veličiny na veličinách spojitých (a případně i diskrétních). Tuto závislost nelze vyjádřit ani modelem čistě spojitým, ani čistě diskrétním.

V předmětu Stochastické systémy jsme se již touto problematikou zabývali pro dvouhodnotovou modelovanou veličinu s alternativním rozdělením

$$\frac{y_t}{P(y_t)} \quad \left| \quad \begin{array}{cc} 0 & 1 \\ p & 1-p \end{array} \right.$$

s modelem

$$f(y_t|p) = p^{y_t} (1-p)^{1-y_t}, \quad p \in (0, 1), \quad y_t = 0, 1,$$

kde jsme předpokládali, že p není konstanta a modelovali jsme ho v závislosti na regresním vektoru nezávislých veličin $\psi_t = [x_{1;t}, x_{2;t}, \dots, x_{n;t}]$ tak, že $g(p) = \psi_t' \theta$ kde θ je vektor regresních koeficientů a funkce g musí mít definiční obor v intervalu $(0, 1)$ a obor hodnot na celé reálné ose - transformuje pravděpodobnost na reálnou osu. Tuto funkci volíme většinou jako logistickou funkci *logit* (odtud název regrese)

$$\text{logit}(p) = \ln \frac{p}{1-p}.$$

Model (bez šumu) má potom tvar

$$\text{logit}(p) = \psi_t' \theta,$$

kde

$$p = P(y_t = 1|x_t).$$

Invertováním modelu dostaneme

$$p = \frac{\exp\{\psi_t' \theta\}}{1 + \exp\{\psi_t' \theta\}} \quad (2.5)$$

a

$$1-p = \frac{1}{1 + \exp\{\psi_t' \theta\}}. \quad (2.6)$$

Odhad parametru p provedeme metodou maximum likelihood. Budeme počítat log-likelihood

$$\begin{aligned} \ln L &= \ln \prod_{t=1}^N f(y_t|p) = \ln \left(p^{\sum y_t} (1-p)^{N-\sum y_t} \right) = \\ &= \ln \left(\left(\frac{\exp\{\psi_t' \theta\}}{1 + \exp\{\psi_t' \theta\}} \right)^{\sum y_t} \left(\frac{1}{1 + \exp\{\psi_t' \theta\}} \right)^{N-\sum y_t} \right) = \\ &= \sum y_t \ln \left(\frac{\exp\{\psi_t' \theta\}}{1 + \exp\{\psi_t' \theta\}} \right) + \left(N - \sum y_t \right) \ln \left(\frac{1}{1 + \exp\{\psi_t' \theta\}} \right). \end{aligned}$$

Tento výraz je možno ještě trochu zjednodušit, nicméně to nic nemění na věci, že pro jeho maximalizaci je nutno použít numerickou metodu. Protože je možno analyticky spočítat derivace tohoto výrazu, je výhodné použití jednoduché Newtonovy metody.

V maximu likelihoodu leží optimální parametry, které dosadíme do modelu

$$\psi_t' \theta = \text{logit}(p)$$

a ze změřeného regresního vektoru a odhadnutých parametrů spočteme $\text{logit}(p) = \text{logit}(P(y_t = 1|\psi_t))$. Inverzí funkce *logit* podle (2.5) určíme $P(y_t = 1|\psi_t)$. Je-li tato pravděpodobnost větší než 0.5, určíme hodnotu $y_t=1$. V opačném případě bude $y_t = 0$.

Poznámka

Všimněme si, že závěr předchozího odstavce vyznívá jako klasifikace. Pro odhadnuté parametry logistické regrese můžeme všechny případné regresní vektory přiřadit do dvou skupin. V první skupině jsou ty, kterým regrese přiřadí $y_t = 1$ a v druhé ty, kterým odpovídá $y_t = 0$.

Žádoucí proto bude rozšířit logistickou regresi in na případy, kdy výstup y_t může nabývat více než dvě hodnoty. Touto problematikou se budeme zabývat později.

2.2 Dynamické modely

Dynamický model popisuje modelovanou veličinu nejen v závislosti na jiných veličinách ze stejného časového okamžiku, ale také na svých zpožděných hodnotách. Nejznámějším pokusem s dynamickým systémem je sledování rychlosti určitého tělesa, na které působí síla. Změna rychlosti nemůže být skoková - rychlost se působením síly zvyšuje postupně v závislosti na již dosažené rychlosti.

Protože dynamický systém “žije svým určitým životem” i když se veličiny které na něj působí mění, má u dynamických systémů smysl zajímat se o predikci jejich chování.

Chování dynamického systému je složitější než statického. Proto i úlohy s ním spojené jsou složitější.

REGRESNÍ DYNAMICKÝ MODEL S NORMÁLNÍM ŠUMEM

Model je zadán podmíněnou hp modelované veličiny v závislosti na jejích zpožděných hodnotách a dalších veličinách mezi kterými bývá také řízení - tyto veličiny jsou shrnuty v regresním vektoru. Obecný tvar modelu je

$$f(y_t | \psi_t, \Theta),$$

kde y_t je modelovaná veličina, ψ_t je regresní vektor a $\Theta = \{\theta, r\}$ jsou parametry modelu, θ jsou regresní koeficienty a r je rozptyl šumu.

V případě lineárního regresního modelu s normálním rozležením je uvedená hp definována rovnicí

$$\begin{aligned} y_t &= \theta' \psi_t + e_t = \\ &= b_0 u_t + a_1 y_{t-1} + b_1 u_{t-1} + \dots + a_n y_{t-n} + b_n u_{t-n} + k + e_t, \end{aligned}$$

kde $\theta' = [b_0, a_1, b_1, \dots, a_n, b_n, k]$ jsou regresní koeficienty, u_i je řízení a e_t je normální bílý šum s nulovou střední hodnotou a konstantním rozptylem r .

Statistiky pro odhad parametrů tohoto modelu jsou

- V_t - informační matice (uchovává střední hodnoty, rozptyly a kovariance veličin)
- κ_t - počítadlo (počítá datové vzorky použité pro odhad)

Přepoččet statistik je

$$\begin{aligned} V_t &= V_{t-1} + \begin{bmatrix} y_t \\ \psi_t \end{bmatrix} \begin{bmatrix} y_t & \psi_t' \end{bmatrix}, \\ \kappa_t &= \kappa_{t-1} + 1, \end{aligned}$$

který startuje z hodnot V_0 a κ_0 , což jsou statistiky apriorního rozdělení.

Aposteriorní hp je

$$f(\Theta | d(N)) \propto r^{-.5\kappa_t} \exp \left\{ -\frac{1}{2r} [-1, \theta'] V_t \begin{bmatrix} -1 \\ \theta \end{bmatrix} \right\}$$

a bodové odhady parametrů

$$\hat{\theta} = V_{\psi}^{-1} V_{y\psi}, \quad \hat{r} = \frac{V_y - V_{y\psi}' V_{\psi}^{-1} V_{y\psi}}{\kappa_t},$$

kde

$$V_t = \begin{bmatrix} V_y & V_{y\psi}' \\ V_{y\psi} & V_{\psi} \end{bmatrix}$$

s dimenzemi y a ψ .

KATEGORICKÝ DYNAMICKÝ MODEL

Dynamický kategorický model je model, ve kterém vystupují jen diskrétní veličiny a modelovaná veličiny závisí také na svých minulých hodnotách. Tento model generuje podmíněnou hp ve tvaru

$$f(y_t | \psi_t, \Theta) = \Theta_{y_t | \psi_t}.$$

Tento model lze vyjádřit ve formě tabulky (např. pro $\psi_t = [u_t, y_{t-1}]'$ a $y_t, u_t \in \{1, 2\}$ dostaneme)

$$f(y_t | [u_t, y_{t-1}]', \Theta)$$

	$y_t = 1$	$y_t = 2$
$[u_t, y_{t-1}] = [1, 1]$	$\Theta_{1 11}$	$\Theta_{2 11}$
$[u_t, y_{t-1}] = [1, 2]$	$\Theta_{1 12}$	$\Theta_{2 12}$
$[u_t, y_{t-1}] = [2, 1]$	$\Theta_{1 21}$	$\Theta_{2 21}$
$[u_t, y_{t-1}] = [2, 2]$	$\Theta_{1 22}$	$\Theta_{2 22}$

kde platí $\Theta_{i|jk} \geq 0$, $\sum_{c=1}^2 \Theta_{c|jk}$, $i, j, k \in \{1, 2\}$, tedy parametry jsou pravděpodobnosti - jsou nezáporné a v proměnné před podmínkou se sčítají na jedničku.

Na tento model lze také pohlížet jako na množinu nezávislých modelů pro výstupní veličinu y_t , indexovanou hodnotami veličin v podmínce. Tedy: podle konfigurace hodnot v regresním vektoru vybereme model a z něj generujeme hodnotu výstupu.

Statistika modelu pro odhad parametru Θ má stejný tvar jako parametr

$$\nu_{y_t|[u_t, y_{t-1}]}$$

	$y_t = 1$	$y_t = 2$
$[u_t, y_{t-1}] = [1, 1]$	$\nu_{1 11}$	$\nu_{2 11}$
$[u_t, y_{t-1}] = [1, 2]$	$\nu_{1 12}$	$\nu_{2 12}$
$[u_t, y_{t-1}] = [2, 1]$	$\nu_{1 21}$	$\nu_{2 21}$
$[u_t, y_{t-1}] = [2, 2]$	$\nu_{1 22}$	$\nu_{2 22}$

a její přepočtení se děje podle pravidla

$$\nu_{i|jk;t} = \nu_{i|jk;t-1} + \delta(i|j, k; y_t|u_t, y_{t-1}),$$

kde $\delta(i|j, k; y_t|u_t, y_{t-1}) = 1$ pro $y_t = i$, $u_t = j$, $y_{t-1} = k$ jinak je to nula.

Aposteriorní hp je

$$f(\Theta|d(t)) = \prod_{i,j,k} \Theta_{i|jk}^{\nu_{i|jk;t}}.$$

Bodové odhady

$$\Theta_{i|jk} = \frac{\nu_{i|jk}}{\sum_c \nu_{c|jk}}. \quad (2.7)$$

Tyto bodové odhady dostaneme prostě tak, že vezmeme tabulku statistik a její řádky normalizujeme tak, aby jejich součet byl roven jedné.

STAVOVÝ MODEL

Popisuje neměřenou veličinu x_t - stav. Model má dvě části (i) stavovou a (ii) výstupní

$$(i) \quad f(x_{t+1}|x_t, u_t), \quad (2.8)$$

$$(ii) \quad f(y_t|x_t, u_t). \quad (2.9)$$

Normální, lineární model má rovnice

$$x_{t+1} = Mx_t + Nu_t + w_t, \quad (2.10)$$

$$y_t = Ax_t + Bu_t + v_t, \quad (2.11)$$

kde M , N , A , B jsou známé parametry modelu a w_t a v_t jsou stavový a výstupní šum s kovariančními maticemi r_w a r_v .

Odhad stavu provádí Kalmanův filtr, který zapsaný v hp má tvar

$$\begin{aligned} f(y_t|d(t-1)) &= \int_{x_1^*} f(y_t|x_t, u_t) f(x_t|d(t-1)) dx_t, \\ f(x_t|d(t)) &= \frac{f(y_t|x_t, u_t) f(x_t|d(t-1))}{f(y_t|d(t-1))}, \\ f(x_{t+1}|d(t)) &= \int_{x_1^*} f(x_{t+1}|x_t, u_t) f(x_t|d(t-1)) dx_t. \end{aligned}$$

Zavedeme značení

$$\begin{aligned} f(y_t|d(t-1)) &= N_{y_t}(\hat{y}_t, r_y), \\ f(x_t|d(t-1)) &= N_{x_t}(x_{t|t-1}, r_{t|t-1}), \\ f(x_t|d(t)) &= N_{x_t}(x_{t|t}, r_{t|t}), \\ f(x_{t+1}|d(t)) &= N_{x_{t+1}}(x_{t+1|t}, r_{t+1|t}). \end{aligned}$$

Vývoj charakteristik těchto hp dává **Kalmanův filtr**.

$$\begin{aligned} \hat{y} &= A\hat{x} + Bu_t \\ r_y &= r_v + Ar_x A' \\ r_x &= r_x - r_x A' r_y^{-1} A r_x \\ e_p &= y_t - \hat{y} \\ K_g &= r_x A' r_y^{-1} \\ \hat{x} &= \hat{x} + K_g e_p \\ \hat{x} &= M\hat{x} + Nu_t \\ r_x &= r_w + Mr_x M' \end{aligned}$$

MODEL SMĚSI KOMPONENT

Model směsi se uplatní jestliže modelovaný systém

- se přepíná mezi několika pracovními módy (jako je tomu například u dopravy v závislosti na denní době, během týdne nebo i v různých ročních obdobích),
- je nelineární a my se snažíme jeho hp aproximovat směsí rozdělání.

Navíc, popis systému pomocí směsi komponent lze využít rovněž pro úlohu klasifikace modelované veličiny. Třídy klasifikace odpovídají komponentám směšového modelu.

Model směsi komponent má tvar

$$f(y_t|\psi_t, \Theta) = \sum_{c=1}^{n_c} \alpha_c f(y_t|c, \psi_t, \Theta_c),$$

kde v argumentu sumy jsou komponenty - jednoduché modely popisující stejnou modelovanou veličinu, pracující se stejným regresním vektorem (což obecně není nutno) a mající každý své parametry. Typickým příkladem komponent mohou být regresní modely z nichž každý má svou pozici (střední hodnotu) a šířku (kovarianční matici).

Odhad modelu se provádí podle Bayesova vzorce, který ovšem vede na výpočetně nerealizovatelný postup a proto je třeba určité aproximace. Tato aproximace spočívá v tom, že deterministické ukazovátka na aktivní komponentu, které je reprezentováno funkcí delta

$$\delta(c, c_t) = \begin{cases} 1 & \text{pro } c = c_t \\ 0 & \text{pro ostatní } c \end{cases},$$

kde $c = 1, 2, \dots, n_c$ a c_t označuje aktivní komponentu v čase t , nahradíme jeho střední hodnotou. Dostaneme tak pravděpodobnostní ukazovátka, které je reprezentováno vektorem pravděpodobností $w_t = [w_{1;t}, w_{2;t}, \dots, w_{n_c;t}]$ toho, že příslušná komponenta je aktivní $w_{c;t} = Pr(c = c_t|d(t))$.

Odtud pak dostáváme vztahy pro přepočítání jednotlivých komponent

$$\begin{aligned} \kappa_{c;t} &= \kappa_{c;t-1} + w_{c;t}, \\ V_{c;t} &= V_{c;t-1} + w_{c;t} \Psi_t \Psi_t', \end{aligned}$$

pro $c = 1, 2, \dots, n_c$.

Bodové odhady parametrů pro jednotlivé komponenty pak spočteme stejně, jako pro jednoduché modely které komponenty reprezentují.

2.3 Dodatky k odhadu

V této kapitole uvedeme některá rozšíření a speciality v problému odhadu modelu. Zaměříme se na nejčastěji využívaný dynamický regresní model s normálním šumem. Většinu z uvedeného lze ale aplikovat i na ostatní modely.

Bayesův vzorec

Bayesův vzorec má základní tvar

$$f(\Theta|d(t)) \propto f(y_t|\psi_t, \Theta) f(\Theta|d(t-1)), \quad (2.12)$$

kde hp (zleva) jsou (i) aposteriorní, (ii) model, (iii) apriorní. Rekurze běží pro $t = 0, 1, \dots$, kde $t = 0$ označuje apriorní informaci. Předpokládá se platnost přirozených podmínek řízení (viz Dodatky 9.3).

Postupným dosazováním Bayesova vzorce lze aposteriorní hp vyjádřit v jednorázovém tvaru

$$f(\Theta|d(t)) \propto \left[\prod_{\tau=1}^t f(y_\tau|\psi_\tau, \Theta) \right] f(\Theta|d(0)) = L_t(\Theta) f(\Theta|d(0)),$$

kde $L_t(\Theta) = [\prod_{\tau=1}^t f(y_\tau|\psi_\tau, \Theta)]$ je likelihood.

Statistiky a jejich přepoččet

Pro normální regresní jednorozměrný model volíme apriorní hp ve tvaru inverzního Gauss-Wishartova rozdělení jehož tvar se zachová i pro aposteriorní hp. Statistiky tohoto rozdělení jsou κ_t - počítadlo kroků a V_t - informační matice.

Přepoččet statistik

$$\begin{aligned} \kappa_t &= \kappa_{t-1} + 1, \\ V_t &= V_{t-1} + \Psi_t \Psi_t', \end{aligned}$$

kde $\Psi_t = [y_t, \psi_t']' = [y_t, u_t, y_{t-1}, u_{t-1}, \dots, y_{t-n}, u_{t-n}]'$ je rozšířený regresní vektor a n je řád modelu.

Bodové odhady jsou

$$\hat{\theta} = V_\psi^{-1} V_{y\psi}, \quad \hat{r} = \frac{V_y - V_{y\psi}' V_\psi^{-1} V_{y\psi}}{\kappa_t},$$

kde V_y , $V_{y\psi}$ a V_ψ jsou submatice matice V_t o rozměrech daných vektorem Ψ .

Jednorázový výpočet bodových odhadů pro N změřených dat provedeme takto. Sestavíme matice

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix} \quad \text{a} \quad X = \begin{bmatrix} \psi_1' \\ \psi_2' \\ \dots \\ \psi_N' \end{bmatrix}.$$

Bodové odhady regresních koeficientů a rozptylu šumu jsou

$$\hat{\theta} = (X'X)^{-1} X'Y \quad (2.13)$$

$$\hat{r} = \frac{Y'Y - Y'X(X'X)^{-1} X'Y}{N} \quad (2.14)$$

Bodové odhady určují hodnoty parametrů bez uvažování jejich nejistoty. Pokud chceme do dalších výpočtu, využívajících odhadnutý model, zahrnout i neurčitost, musíme použít celou aposteriorní hp do které dosadíme spočtené statistiky V_N a κ_N

$$f(\Theta|d(N)) \propto r^{-0.5\kappa_N} \exp \left\{ -\frac{1}{2r} [-1, \theta'] V_N \begin{bmatrix} -1 \\ \theta \end{bmatrix} \right\}.$$

Odhad struktury

Při návrhu modelu je třeba zvolit modelovanou veličinu a vybrat další veličiny, které modelovanou veličinu ovlivňují. Při návrhu dynamického modelu je další důležitou otázkou zvolit řád modelu, tj. maximální zpoždění modelované veličiny, jejíž hodnota má ještě vliv na její okamžitou hodnotu. Tomuto kroku se říká odhad struktury modelu.

Tento odhad lze jednoduše provést na základě apriorních dat takto

1. Apriorní data normujeme tak, aby jednotlivé veličiny měly nulový průměr a jednotkový rozptyl. Tj. normovanou veličinu \tilde{x} z veličiny x s průměrnou hodnotou m a směrodatnou odchylkou s dostaneme takto

$$\tilde{x} = \frac{x - m}{s}.$$

2. Provedeme regresní analýzu pro modelovanou veličinu a všechny dostupné nezávislé veličiny. Výsledkem jsou regresní koeficienty. Velmi malé koeficienty říkají že když se příslušná nezávislá veličina mění, její změna se změny modelované veličiny promítne velmi málo. Tedy na ní téměř nezávisí.
3. Určíme hranici (třeba 5% největšího koeficientu) a do modelu vezmeme jen ty veličiny, kterým odpovídá koeficient v absolutní hodnotě větší než tato hranice.

Poznámka

Tento odhad můžeme udělat pro každou veličinu zvlášť, nebo jen pro modelovanou veličinu a ostatní brát se stejnou hloubkou zpoždění.

Příklad

Odhad struktury budeme demonstrovat na příkladu se simulovanými daty. Pro simulaci uvažujeme regresní model

$$y_t = ay_{t-1} + b_0u_t + b_1u_{t-1} + cv_t + e_t,$$

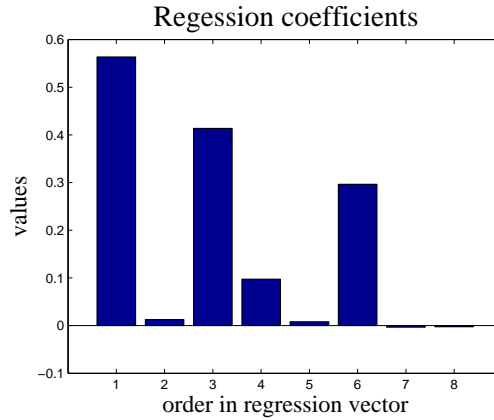
kde v_t je externí porucha a e_t je normální šum. Simulací získáme data, která považujeme za apriorní data a na jejich základě budeme odhadovat strukturu modelu pro další použití.

Pro odhad struktury nejprve normalizujeme data: např. pro veličinu x spočteme

$$\frac{x - E[x]}{\sqrt{D[x]}}.$$

Pro normované veličiny provedeme regresní analýzu s hodně širokým regresním vektorem. V našem případě do něj zahrneme dva zpožděné výstupy, vstup, dva zpožděné vstupy, externí porucha, další (v simulaci nepoužitá) externí porucha a konstanta. Tedy v regresním vektoru jsou prvky 2, 5, 7 a 8 navíc - nebyly použity v simulaci.

Po odhadu dostaneme vektor regresních koeficientů na následujícím obrázku



Vidíme, že přesně ty veličiny, které do modelu nepatří, mají velmi malé regresní koeficienty. Do modelu tedy vybereme jen ty veličiny, jejichž koeficienty jsou větší, než téměř nulové. Zde jsou to veličiny 1 (zpožděný výstup), 3 (vstup), 4 (zpožděný vstup) a 6 (první externí porucha).

Program k tomuto příkladu je uveden v [10.7](#).

Zapomínání

Odhad parametrů modelu se provádí za předpokladu, že parametry jsou konstantní, v čase neproměnné.

POZNÁMKA

Pozor, nepleťte si časový vývoj bodových odhadů parametrů s proměnností parametrů. Podle předpokladu jsou parametry konstanty a bodové odhady hledají hodnotu těchto konstant. Startují ze zadaných počátečních hodnot a postupně se k hodnotám skutečných parametrů přibližují. Odtud jejich vývoj.

Ve skutečnosti je situace ještě složitější. O skutečných parametrech můžeme mluvit je tehdy, je-li systém simulován a odhadovaný model má stejnou strukturu (stejně veličiny a stejný řád). Reálný systém žádné parametry nemá (reálný systém může být křížovatka, nehody v určité komunikaci, retardéry nebo řidič v jedoucím automobilu - tady můžeme sledovat a měříme veličiny, ale žádné parametry tu nejsou). Skutečnými parametry máme na mysli parametry modelu, který co nejlépe danou soustavu popisuje. Pokud by takový model chvíli soustavu dobře popisoval a za chvíli ne, je zřejmé, že vlastnosti soustavy jsou takové, že by vyžadovaly model s proměnnými parametry. Pokud se změny parametrů dějí velmi pomalu, mluvíme o pomalu se měnících parametrech. Pokud se parametry mění tak, že nabývají jen konečný počet hodnot, lze jako model využít směs komponent. Jestliže se parametry mění rychle, jedná se o nestacionární systém, se kterým je třeba zacházet jinak. Nestacionárními systémy se zde zabývat nebudeme.

Pokud se parametry systému (ve smyslu předešle poznámky) pomalu mění, lze použít standardní metody bayesovského odhadování, kombinované s technikou zapomínání. Ta

spočívá v tom, že čím jsou data použitá k odhadu starší, tím se jim přiřadí menší váha. Tak největší vliv na odhady mají nová data a vliv starších dat pomalu ustává. Při metodě exponenciálního zapomínání se jako váha dat volí hodnota λ^t , kde $\lambda \in (0, 1)$ a t je čas. Hodnota koeficientu zapomínání λ se volí podle toho, jakou délku historie dat si v odhadování přejeme. Běžná hodnota je $\lambda = 0.985$. Při ní dostáváme následující hodnoty datových vah

zpoždění	1	2	3	4	5	10	50	100	500
váha	0.985	0.970	0.956	0.941	0.927	0.86	0.55	0.26	0.0005

Váha v podstatě říká, jaký podíl dat se při odhadu uplatní. Při zpoždění 50 tedy hodnota jednoho datového údaje svou významností odpovídá dvěma aktuálním datovým údajům.

Váhy se do odhadování zavedou velice jednoduše. Stačí modifikovat Bayesův vzorec následujícím způsobem

$$f(\Theta|d(t)) \propto f(y_t|\psi_t, \Theta) f(\Theta|d(t-1))^\lambda.$$

Umocnění apriorní hp na λ znamená její “rozšlápnutí”, což je zvýšení neurčitosti, které má právě popsany efekt (menší důvěryhodnost starších dat). Protože Bayesův vzorec je rekurzivní, čím starší je apriorní hp, tím menší je její mocnina (čím starší, tím méně důvěryhodné).

Pro regresní model s normální rozdělením dostaneme přepočtení statistik v následujícím tvaru

$$\begin{aligned} \kappa_t &= \kappa_{t-1} + \lambda. \\ V_t &= V_{t-1} + \lambda \Psi_t \Psi_t'. \end{aligned}$$

Tento přepočtení plyne přímo z Bayesova vzorce po dosazení normálního modelu s rozšířeným regresním vektorem $\Psi_t = [y_t, \psi_t']'$ a hp pro parametry ve tvar inverzního Gauss-Wishartova rozdělení se statistikami κ_t a V_t .

Bodové odhady

Optimálním bodovým odhadem podle kvadratického kriteriá optimality je střední hodnota podmíněná vším, co je v daném okamžiku k dispozici (většinou jsou to změřená data). Tento fakt je ukázán ve skriptu [Nagy: \[2012\]](#). Chceme-li tedy získat bodový odhad $\hat{\Theta}_t$ parametru modelu Θ , počítáme

$$\hat{\Theta}_t = E[\Theta|d(t)] = \int_{\Theta^*} \Theta f(\Theta|d(t)) d\Theta. \quad (2.15)$$

Uvedený výpočet bodového odhadu lze vyjádřit jako aproximaci, při které veškerou pravděpodobnost hodnot Θ rozdělenou aposteriorní hp $f(\Theta|d(t))$ soustředíme do jediného

bodů - střední hodnoty (nebo případně do maxima aposteriorní hp). Formálně lze tuto aproximaci popsat takto

$$f(\Theta|d(t)) \rightarrow \delta(\Theta, \hat{\Theta}_t),$$

kde $\delta(x, a)$ je Diracův impulz¹. Dosadíme-li nyní do (2.15), dostaneme totéž

$$\int_{\Theta^*} \Theta f(\Theta|d(t)) d\Theta = \int_{\Theta^*} \Theta \delta(\Theta, \hat{\Theta}_t) d\Theta = \hat{\Theta}_t.$$

Lze proto udělat následující závěr: Optimálním odhadem parametru Θ je aposteriorní hp, která poskytuje nejen hodnotu odhadu, ale také vypovídá o její přesnosti. Výpočet podle (2.15) potom probíhá tak, že bereme všechny možné hodnoty parametru, vážíme je jejich pravděpodobností a “sčítáme” - tj. děláme vážený průměr.

Pokud se spokojíme s bodovým odhadem, vezmeme jen jednu hodnotu (střední hodnotu, případně modus) a dosadíme ji za parametr.

Tuto akci můžeme zobecnit: Rádi bychom použili model $f(y_t|d(t-1))$, ale máme k dispozici jen model parametrizovaný $f(y_t|\psi_t, \Theta)$. Musíme proto zavést odhad parametrů. Postup je následující

$$\begin{aligned} f(y_t|d(t-1)) &= \int_{\Theta^*} f(y_t, \Theta|d(t-1)) d\Theta = \\ &= \int_{\Theta^*} f(y_t|\psi_t, \Theta) f(\Theta|d(t-1)) d\Theta. \end{aligned}$$

Tedy, abychom získali model pro neznámé parametry, musíme mít k dispozici odhad parametrů ve formě aposteriorní hp. Potom, podobně jako v (2.15) bereme modely se všemi možnými parametry, vážíme pravděpodobnostmi těchto parametrů a “sčítáme” - děláme vážený průměr. Pokud se opět spokojíme s bodovým odhadem, lze aposteriorní hp aproximovat Diracovým impulzem $\delta(\Theta, \hat{\Theta}_t)$ a dostaneme

$$\begin{aligned} &\int_{\Theta^*} f(y_t|\psi_t, \Theta) f(\Theta|d(t-1)) d\Theta = \\ &= \int_{\Theta^*} f(y_t|\psi_t, \Theta) \delta(\Theta, \hat{\Theta}_t) d\Theta = f(y_t|\psi_t, \hat{\Theta}_t). \end{aligned} \quad (2.16)$$

Výsledkem je tedy parametrický model, kde za parametry je dosazen bodový odhad.

¹Diracův impulz je funkce, pro kterou platí: $\delta(x, a) \neq 0$ jen pro $x = a$ a přitom $\int_{-\infty}^{\infty} \delta(x, a) dx = 1$.

Pro tuto funkci platí (a někdy se bere jako definice), že pro všechny rozumné funkce $g(x)$, tj. takové, pro které má následující výraz smysl, je

$$\int_{-\infty}^{\infty} g(x) \delta(x, a) dx = g(a)$$

Obecný závěr je tedy: Optimální práce s objektem, který obsahuje neznámé parametry, je počítat jeho vážený průměr pro všechny možné hodnoty parametrů s vahami, které se rovnají pravděpodobnostem těchto hodnot. Bodový odhad objektu dostaneme tak, že spočteme bodový odhad parametru a do objektu jej za parametr dosadíme. Tedy ve vztahu (2.16) vynecháme integrál, vynecháme aposteriorní hp a zůstane jen objekt (model), kde místo parametru figuruje jeho bodový odhad.

3 Predikce

Hodiny: 7

Aplikace: předpověď intenzity s denním průběhem.

Předpovědi máme na mysli odhad hodnoty budoucího výstupu soustavy, počítaného na základě současných a minulých dat. V obecném případě se uvažuje model s neznámými parametry. Tato úloha je však většinou analyticky neřešitelná, proto se spíše zabýváme predikcí s modelem se známými parametry. Za parametry často dosazujeme bodové odhady spočtené v úloze odhadu, která běží paralelně s úlohou predikce.

Podle počtu kroků lze predikci dělit na jednokrokovou (předpověď do budoucího časového okamžiku) a vícekrokovou (předpověď na více kroků dopředu).

Poznámka

V úloze predikce je výhodné posunout časový index o krok dopředu. Předpokládáme následující uspořádání času. Jsme v čase t a známe data až do t , včetně. Predikci děláme na čas $t + n_p$, tedy n_p kroků dopředu. Přitom známá data jsou $d(t)$.

3.1 Jednokroková predikce

Znamé parametry

Jednokrokovou predikci výstupu pro model se známými parametry provádí přímo sám model

$$f(y_{t+1}|u_{t+1}, d(t)) = f(y_{t+1}|\psi_{t+1}, \Theta).$$

Pro známá data $\psi_{t+1} = [u_{t+1}, y_t, u_t, \dots, y_{t-n+1}, u_{t-n+1}]' \subset \{u_{t+1}, d(t)\}^2$ a známé pevné parametry Θ dává přímo rozdělení budoucího výstupu y_{t+1} .

Bodová predikce \hat{y}_{t+1} se spočte jako střední hodnota výstupu

$$\hat{y}_{t+1} = \int_{y^*} y_{t+1} f(y_{t+1}|\psi_{t+1}, \Theta) dy_{t+1}.$$

²Rízení předpokládáme pevné a předem dané na celém intervalu predikce. Pokud tomu tak není, je třeba jej modelovat. To už je ale jiná písnička.

Pro normální model $y_{t+1} = \psi'_{t+1}\theta + e_{t+1}$ je to

$$\hat{y}_{t+1} = \psi_{t+1}\theta.$$

Odvození je v Nagy: [2012].

Neznámé parametry

Jednokrokovou předpověď pro neznáme parametry ve formě prediktivní hp lze vyjádřit takto

$$f(y_{t+1}|u_{t+1}, d(t)) = \int_{\Theta^*} f(y_{t+1}|\psi_{t+1}, \Theta) f(\Theta|d(t)) d\Theta, \quad (3.1)$$

kde $f(y_{t+1}|\psi_{t+1}, \Theta)$ je předpověď se známými parametry a $f(\Theta|d(t))$ je aposteriorní hp, tedy odhad parametrů.

Pro bodový odhad $\hat{\Theta}_t$ parametru Θ dostaneme

$$f(y_{t+1}|u_{t+1}, d(t)) = f(y_{t+1}|\psi_{t+1}, \hat{\Theta}_t),$$

viz str. 30 vzoreček (31).

Bodový odhad budoucího výstupu \hat{y}_{t+1} určíme jako střední hodnotu

s použitím bodového odhadu parametru Θ , určíme jako střední hodnotu z předchozího modelu s odhadnutým parametrem

$$\begin{aligned} \hat{y}_{t+1} &= E[y_{t+1}|d(t)] = \int_{y^*} y_{t+1} f(y_{t+1}|u_{t+1}, d(t)) dy_{t+1} = \\ &= \int_{y^*} y_{t+1} \int_{\Theta^*} f(y_{t+1}|\psi_{t+1}, \Theta) f(\Theta|d(t)) d\Theta dy_{t+1}. \end{aligned}$$

Použijeme-li pro odhad Θ také bodový odhad, dostaneme

$$\hat{y}_{t+1} = \int_{y^*} y_{t+1} f(y_{t+1}|\psi_{t+1}, \hat{\Theta}_t) dy_{t+1}.$$

Pro regresní model $y_{t+1} = \psi'_{t+1}\theta + e_{t+1}$ s bodovým odhadem parametru $\hat{\theta}_t$ je tato bodová predikce

$$\hat{y}_{t+1} = \psi'_{t+1}\hat{\theta}_t.$$

Jednokroková predikce se hodí pro ověření např. nových algoritmů odhadu a podobně. Zde se většinou vyhodnocuje chyba odhadu, například součtem kvadrátů chyby predikce, což jsou rozdíly mezi skutečnou hodnotou výstupu a jeho jednokrokovou predikcí.

3.2 Vícekroková predikce

Prakticky zajímavější, ale také algoritmicky a výpočetně náročnější je vícekroková predikce, tj. odhad hodnoty výstupu na několik period dopředu. Tato predikce může být použita např. pro předpověď intenzity dopravy a využita pro varování před blížící se dopravní kalamitou nebo pro to, aby se udělala příslušná opatření pro zachování plynulosti dopravy.

Předpověď má nyní několik variant podle toho zda:

1. Parametry modelu

- (a) známe,
- (b) neznáme a použijeme jejich bodové odhady,
- (c) neznáme a použijeme celou aposteriorní hp.

2. Pro predikci použijeme

- (a) celou prediktivní hp,
- (b) bodový odhad průběžné predikce,
- (c) bodový odhad průběžné i konečné predikce³.

V následujících odstavcích ukážeme jednotlivé případy predikce. Protože se jedná o komplikovanou úlohu s mnoha variantami, budeme uvažovat co nejjednodušší případ (autoregresní model 1. řádu bez řízení $f(y_t|y_{t-1}, \Theta)$ s rovnicí $y_t = ay_{t-1} + e_t$) a dvou-krokovou predikci. Zobecnění této situace je přímočaré. Opět použijeme posunuté indexování času, kdy známá data jsou $d(t)$ a predikujeme y_{t+2} .

Obecný případ

Nejsložitější případ je výpočet prediktivní hp pro model s neznámými parametry. Konstrukce je následující

$$\begin{aligned}
 f(y_{t+2}|y(t)) &= \int_{\Theta^*} \int_{y_{t+1}^*} f(y_{t+2}, y_{t+1}, \Theta|y(t)) dy_{t+1} d\Theta = \\
 &= \int_{\Theta^*} \int_{y_{t+1}^*} f(y_{t+2}|y_{t+1}, \Theta) f(y_{t+1}|y_{t+1}, \Theta) f(\Theta|y(t)) dy_{t+1} d\Theta = \\
 &= \int_{\Theta^*} \underbrace{\int_{y_{t+1}^*} f(y_{t+2}|y_{t+1}, \Theta) f(y_{t+1}|y_{t+1}, \Theta) dy_{t+1}}_{\text{prediktor pro známé parametry}} \underbrace{f(\Theta|y(t))}_{\text{odhad parametrů}} d\Theta.
 \end{aligned}$$

Jak je ve výsledku naznačeno, úloha se skládá ze dvou podúloh:

³Co je průběžná a konečná predikce bude patrné z dalšího.

1. Výpočet dvou krokového prediktoru pro známé parametry

$$f(y_{t+2}|y(t), \Theta) = \int_{y_{t+1}^*} f(y_{t+2}|\psi_{t+2}, \Theta) f(y_{t+1}|\psi_{t+1}, \Theta) dy_{t+1}.$$

2. Odhad parametru Θ

$$f(y_{t+2}|y(t)) = \int_{\Theta^*} f(y_{t+2}|y(t), \Theta) f(\Theta|y(t)) d\Theta.$$

Přítom obě podúlohy mají formálně stejný tvar. První hp za integrálem je objekt (srv odst. Bodový odhad na str. 2.3), závislý na neznámé veličině - první je y_{t+1} a druhá Θ . Dále následuje hp, popisující rozdělení neznámé veličiny. Objekt s neznámou veličinou se tedy počítá jako vážený průměr s vahami rovnými pravděpodobnostem hodnot neznámé veličiny.

Poznámka

Všimněme si, že pro výpočet konečného odhadu y_{t+2} potřebujeme určit i průběžný odhad veličiny y_{t+1} . V tomto smyslu jsme v úvodu kapitoly hovořili o průběžné a konečné predikci.

Bodové odhady v predikci

Víme již, že výpočet obecného odhadu pro hp odhadovaných veličin je složité a často dokonce analyticky nespočitatelné. Proto se budeme zajímat o zjednodušení, které dostaneme při požití bodových odhadů. Jak jsme již naznačili, bude se jednat o bodové odhady při odhadu parametrů, při výpočtu průběžné predikce a při určení konečné predikce.

Nejdříve budeme uvažovat bodové odhady parametrů a bodové průběžné predikce. Tedy budeme aproximovat $f(\Theta|y(t)) \rightarrow \delta(\Theta, \hat{\Theta}_t)$ a $f(y_{t+1}|y(t), \Theta) \rightarrow \delta(y_{t+1}, \hat{y}_{t+1}(\Theta))$, kde $\hat{\Theta}_t$ je bodový odhad parametru a $\hat{y}_{t+1}(\Theta)$ je bodová průběžná predikce, která je ovšem závislá na parametru Θ . Tak dostaneme

1. Bodový odhad prediktoru

$$\begin{aligned} f(y_{t+2}|y(t), \Theta) &= \int_{y_{t+1}^*} f(y_{t+2}|y_{t+1}, \Theta) \delta(y_{t+1}, \hat{y}_{t+1}(\Theta)) dy_{t+1} = \\ &= f(y_{t+2}|\hat{y}_{t+1}, \Theta). \end{aligned}$$

2. Bodový odhad parametru

$$\begin{aligned}
 f(y_{t+2}|y(t)) &= \int_{\Theta^*} f(y_{t+2}|y(t), \Theta) \delta(\Theta, \hat{\Theta}_t) d\Theta = \\
 &\underbrace{=}_{\text{bod. odhad } y_{t+1}} \int_{\Theta^*} f(y_{t+2}|\hat{y}_{t+1}, \Theta) \delta(\Theta, \hat{\Theta}_t) d\Theta = \\
 &\underbrace{=}_{\text{bod. odhad } \Theta} f(y_{t+2}|\hat{y}_{t+1}, \hat{\Theta}_t).
 \end{aligned}$$

Jako výsledek tedy dostáváme

$$f(y_{t+2}|y(t)) = f(y_{t+2}|\hat{y}_{t+1}, \hat{\Theta}_t).$$

To jest, vezmeme model a za neznámé veličiny dosadíme jejich bodové odhady.

Bodová predikce

Budeme-li kromě bodových odhadů v predikci požadovat také výslednou predikci bodovou $\hat{y}_{t+2} = E[y_{t+2}|y(t)]$ a normální regresní model s rovnicí $y_t = ay_{t-1} + e_t$ dostaneme

$$\begin{aligned}
 \hat{y}_{t+2} &= \int_{y^*} y_{t+2} f(y_{t+2}|y(t)) dy_{t+2} = \\
 &= \int_{y^*} y_{t+2} f(y_{t+2}|\hat{y}_{t+1}, \hat{\Theta}_t) dy_{t+2} = \\
 &= E[\hat{a}_t \hat{y}_{t+1} + e_{t+1}|y(t)] = \hat{a}_t \hat{y}_{t+1}.
 \end{aligned}$$

Výsledek je tedy takový: vezmeme model predikované veličiny a za všechno neznámé dosadíme bodové odhady.

V obecném případě (dvoukroková predikce s obecným regresním modelem a předem daným řízením) dostaneme

$$\begin{aligned}
 \hat{y}_{t+2} &= \hat{\psi}'_{t+2} \hat{\Theta}_t = \\
 &= \hat{b}_0 u_{t+2} + \hat{a}_1 \hat{y}_{t+1} + \hat{b}_1 u_{t+1} + \dots + \hat{a}_n y_{t-n+1} + \hat{b}_n u_{t-n+1},
 \end{aligned}$$

kde vystupují bodové odhady regresních koeficientů a bodový odhad výstupu y_{t+1} .

Praktické výpočty při bodové predikci

Počítáme velice jednoduše tak, že vezmeme střední hodnotu modelu (tj. model bez šumu) a dosazujeme jej na horizontu predikce - ukážeme pro model 1. řádu $y_t = b_0u_t + a_1y_{t-1} + e_t$:

$$\hat{y}_{t+1} = b_0u_{t+1} + a_1y_t$$

$$\hat{y}_{t+2} = b_0u_{t+2} + a_1y_{t+1} \doteq b_0u_{t+2} + a_1\hat{y}_{t+1}.$$

Lze provádět postupně tak, jak je naznačeno, nebo dosazovat a odvodit výsledný prediktor

$$\begin{aligned}\hat{y}_{t+2} &= b_0u_{t+2} + a_1\hat{y}_{t+1} = b_0u_{t+2} + a_1(b_0u_{t+1} + a_1y_t) = \\ &= a_1^2y_t + b_0u_{t+2} + b_0a_1u_{t+1}.\end{aligned}$$

Obecný vzorec pro k -krokovou predikci s modelem 1. řádu je

$$\hat{y}_{t+k} = a_1^k y_t + b_0 \sum_{i=1}^k a_1^{k-i} u_{t+i}.$$

Prediktivní hp s normálním modelem

V případě normálního regresního modelu lze předchozí postup použít i v případě, kdy požadujeme jako výsledek celou prediktivní hp. Využijeme skutečnosti, že konvoluce dvou normálních rozdělání je opět normální rozdělání a to je plně určeno střední hodnotou a rozptylem. Uvažujeme tedy model (pro jednoduchost opět prvního řádu)

$$y_{t+1} = b_0u_{t+1} + a_1y_t + e_{t+1},$$

kde šum e_t má nulovou střední hodnotu a rozptyl r .

Postupným dosazováním pro dvou-krokovou predikci dostaneme

$$\begin{aligned}y_{t+2} &= b_0u_{t+2} + a_1y_{t+1} + e_{t+2} = \\ &= b_0u_{t+2} + a_1(b_0u_{t+1} + a_1y_t + e_{t+1}) + e_{t+2} = \\ &= b_0u_{t+2} + a_1b_0u_{t+1} + a_1^2y_t + a_1e_{t+1} + e_{t+2}.\end{aligned}$$

Kromě členu s y_t a u_{t+1} , u_{t+2} dostáváme ještě další člen, který popisuje neurčitost prediktoru. Vzorec také vyjadřuje celou náhodnou veličinu y_{t+2} , nikoli jen její střední hodnotu. Proto z ní můžeme spočítat jak střední hodnotu ale i rozptyl, které určují celou prediktivní hp

$$E[y_{t+2}|d(t)] = a_1^2 y_t + b_0 u_{t+2} + a_1 b_0 u_{t+1},$$

$$D[y_{t+2}|d(t)] = D[a_1 e_{t+1} + e_{t+2}] = a_1^2 r + r = [a_1 + 1] r.$$

Prediktivní hp tedy bude

$$f(y_{t+2}|d(t)) = N_{y_{t+2}}(a_1^2 y_t + b_0 u_{t+2} + a_1 b_0 u_{t+1}, [a_1 + 1] r).$$

Obecný vzorec pro k -krokovou predikci je:

Stření hodnota je stejná jako pro bodovou predikci.

Rozptyl je

$$r \sum_{i=1}^k a_1^{2(i-1)}.$$

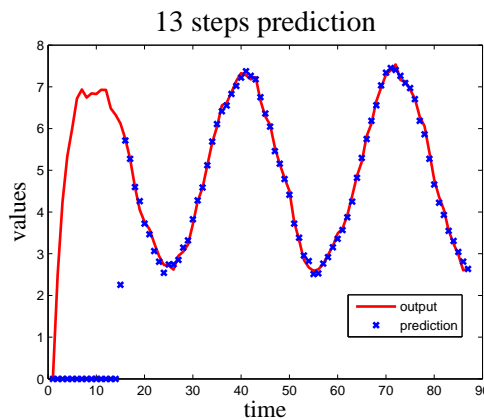
Vzorový příklad

Algoritmus predikce budeme demonstrovat na příkladě. Budeme uvažovat regresní model prvního řádu

$$y_t = a y_{t-1} + b u_t + e_t,$$

kde řídicí veličina (nebo spíše externí porucha) u_t je generována jako zašuměná sinusovka (proto, aby model stále “něco dělal”). Pro simulaci jsou zvoleny parametry $a = 0.6$ a $b = 1$. Model je průběžně identifikován a pro bodové odhady parametrů je počítána predikce výstupu pro n_p kroků.

Výsledek se 13 kroky predikce je na obrázku



Program

Program, realizující příklad predikce je následující

```

clc , clear all , close all
% Prediction with first order regression model
addpath 'funct '

nd=100;           % length of simulation
np=13;           % length of prediction
ord=2;           % model order
a=.6; b=1;       % model parameters

% generation of control variable
u=2+sin(20*(1:nd)/nd)+.1*randn(1,nd);
y=zeros(1,nd);
V=1e-8*eye(3);

% TIME LOOP
for t=2:(nd-np)
    % simulation
    e=.1*randn;           % mnoise
    y(t)=a*y(t-1)+b*u(t)+e; % model

    % estimation
    Ps=[y(t) y(t-1) u(t)]'; % regression vector
    V=V+Ps*Ps';           % statistics update
    [th,s2]=v2thN(V/t,1); % point estimates

% prediction
yy=y(1:t);           % auxiliary data
for i=1:np
    j=t+i;           % time during prediction
    yy(j)=th(1)*yy(j-1)+th(2)*u(j); % running prediction
end
yp(t+np)=yy(t+np); % final prediction at the time
end

% RESULTS
vPred_R

```

3.3 Aplikace

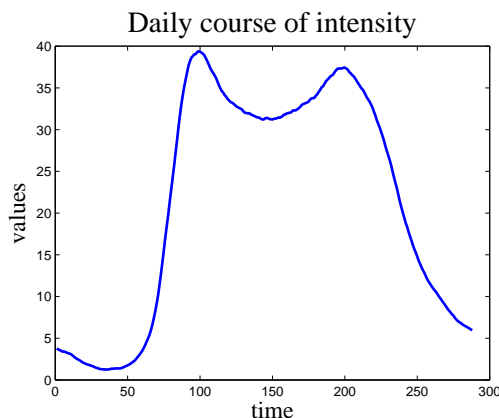
Predikce denního průběhu intenzity

Cílem naší úlohy je předpověď dopravní intenzity v určeném místě městské dopravní komunikace.

Potíž, která je s předpovědí dopravní intenzity spojená je, že tato intenzita je nezáporná a během dne sleduje určitý průběh - tzv. denní průběh intenzity. Tento průběh je podobný tomu, který vytvoří hroznýš, který zažívá slona⁴. Regresní model, jako diferenční rovnice (bez konstanty) popisuje signály, které se po přechodové fázi jako odezvě na počáteční podmínku pohybují kolem nuly nebo (s konstantou) kolem určité konstanty. V žádném případě ale nemohou sami od sebe bez řízení sledovat daný průběh. Proto modelujeme odchylky intenzity od daného (známého) denního průběhu. Prvořadým úkolem je tedy určit denní průběh intenzity pro dané místo.

Denní průběh intenzity

Pokud máme k dispozici větší množství apriori změřených dat (a to můžeme prakticky vždy získat), je situace poměrně jednoduchá. Nejprve je třeba data “nakrájet” po jednotlivých dnech. Potom je třeba dny roztřídit tak, aby v jednotlivých skupinách byly podobné průběhy. Typicky třídíme na všední dny a víkendy. V jednotlivých skupinách průběhy zprůměrujeme a případně vyhladíme, např. průměrováním na určitém okně. Takto získané průběhy intenzit lze považovat za denní průběhy pro jednotlivé typy dní. Typický denní průběh je na následujícím obrázku



Na ose x je vynesena čas v periodě 5 min.

⁴Antoine de Saint-Exupéry: Malý princ

Model intenzity

Označíme-li dopravní intenzitu I_t , její denní průběh \bar{I}_t a odchylku jako $y_t = I_t - \bar{I}_t$, můžeme pro odchylky psát regresní model ve tvaru

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \dots + a_n y_{t-n} + e_t.$$

Tento model lze odhadnout běžným způsobem z měřených dat I_t a známých hodnot denního průběhu \bar{I}_t .

Spočteme k -krokovou bodovou predikci odchylky y_{t+k} a predikovanou intenzitu dopočteme pomocí denního průběhu

$$I_{t+k} = \bar{I}_{t+k} + y_{t+k}.$$

Program

```

clc , clear all , close all
addpath 'funct '
% Prediction of traffic intensity with daily course

load df1 df1
load dt34k12IOV dtI dtO dtV
dI=dtI(1:288);

ord=2;
np=3;

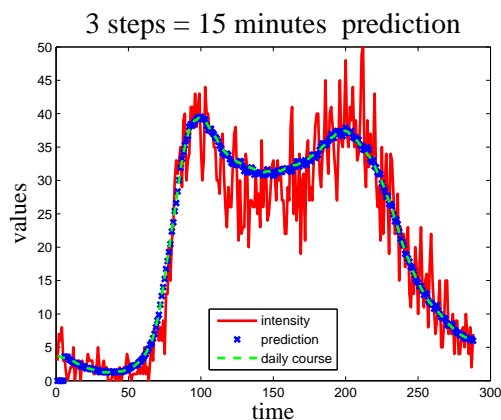
y=dI-df1;
V=zeros(ord+1);
for t=3:288
    Psi=y(t-1:(t-ord));
    V=V+Psi'*Psi;
end
th=v2thN(V/288,1);

dy=dtI((1:288)+288);
for t=3:(288-np)
    yy=dy(1:t)-df1(1:t);
    for i=1:np
        yy(t+i)=yy(t+i-(1:ord))*th';
    end
    yp(t+np)=yy(t+np)+df1(t+np);
end

```

```
% RESULTS
aPredInt_R
```

Výsledky



Z obrázku je patrné, že predikce, navzdory velkému šumu, je velmi dobrá.

4 Filtrace

Hodiny: 8, 9

Aplikace: odhad délky kolony.

Filtraci máme na mysli odhad stavu systému, popsaného stavovým modelem, z průběžně měřených dat - vstupu u_t a výstupu y_t .

4.1 Lineární model

V případě, kdy sledovaný systém obsahuje neměřitelné veličiny⁵, které nás ale zajímají, použijeme pro popis stavový model. Tento model je v lineárním případě dán rovnicemi 2.10 a 2.11. Odhad stavu je realizován Kalmanovým filtrem, uvedeným na straně 25.

Odhad stavu z měřených vstupních a výstupních veličin nelze provést vždy. Může se stát, že některé složky stavu se v těchto veličinách vůbec neprojeví. Podobně je to i s řízením, kdy se snažíme vhodnou volbou vstupů dosáhnout předepsané hodnoty stavu. Podmínkami, kdy stav lze pozorovat a volbou vstupní veličiny daným způsobem ovlivnit se budeme zabývat v následujícím odstavci.

⁵Tyto veličiny ovšem musí splňovat podmínky kladené na stav, tj. v jejich okamžité hodnotě musí být obsažena informace o celém jejich dosavadním vývoji - jejich dynamický popis je prvního řádu.

Pozorovatelnost a říditelnost dynamického systému

Uvažujeme systém, popsaný stavovým modelem (2.10) a (2.11)

$$\begin{aligned}x_{t+1} &= Mx_t + Nu_t + w_t, \\y_t &= Ax_t + Bu_t + v_t.\end{aligned}$$

Pozorovatelnost

Tento systém nazveme **pozorovatelným**, jestliže matice pozorovatelnosti P

$$P = \begin{bmatrix} A \\ AM \\ AM^2 \\ \dots \\ AM^{n-1} \end{bmatrix}$$

má hodnost n (tj. má n lineárně nezávislých řádků), kde n je dimenze stavu.

Pro pozorovatelný systém platí, že z n po sobě naměřených datových dvojic vstup-výstup můžeme spočítat stav z počátku měření.

Tuto vlastnost lze jednoduše demonstrovat. Pro jednoduchost položíme začátek pozorování do času $t = 0$ a nebudeme uvažovat řízení, které na pozorovatelnost nemá vliv. Potom platí

$$\begin{aligned}y_0 &= Ax_0, \\y_1 &= Ax_1 = AMx_0, \\y_2 &= AM^2x_0, \\&\dots \\y_{n-1} &= AM^{n-1}x_0.\end{aligned}$$

Tuto soustavu lze zapsat maticově

$$Y = Px_0.$$

Aby tato soustava byla řešitelná, tj. abychom mohli z měřených y_t vypočítat x_0 , musí mít matice soustavy plnou hodnost n .

Je-li systém pozorovatelný, pak stav x_0 vypočteme pomocí pseudoinverze

$$x_0 = (P'P)^{-1} P'Y.$$

Dosažitelnost

Systém nazveme **dosažitelným**, jestliže matice dosažitelnosti R

$$R = [N, MN, M^2N, \dots, M^{n-1}N]$$

má hodnotu n (což je dimenze stavu).

Pro dosažitelný systém platí, že existuje řízení, které systém v konečném čase převede ze stavu $x_0 = 0$ do libovolného stavu x_t ($t < \infty$).

Demonstrace je opět jednoduchá. V prvním kroku s použitím stavového modelu $x_{t+1} = Mx_t + Nu_t$ (bez šumu) dostaneme

$$x_1 = Mx_0 + Nu_0 = Nu_0,$$

protože $x_0 = 0$. A dále platí

$$\begin{aligned} x_2 &= Mx_1 + Nu_1 = MNu_0 + Nu_1 \\ x_3 &= Mx_2 + Nu_2 = M^2Nu_0 + MNu_1 + Nu_2 \\ &\dots \\ x_k &= Mx_{k-1} + Nu_{k-1} = M^{k-1}Nu_0 + M^{k-2}Nu_1 + \dots + Nu_{k-1} \end{aligned}$$

Pokud má být vektor x_k dimenze n generován pomocí řízení u_0, u_1, \dots, u_{k-1} , musí být mezi koeficienty $M^{k-1}N, M^{k-2}N, \dots, N$ alespoň n nezávislých prvků.

Poznámka

Pojmy pozorovatelnost a dosažitelnost jsou důležité pro to, abychom se dozvěděli, co vlastně od našeho modelu můžeme očekávat. Tyto pojmy jsme jen stručně zmínili. Kromě nich existují ještě pojmy rekonstruovatelnost a říditelnost, které jsou velice podobné zmíněným. Pro bližší informace doporučujeme např. skripta ([Štecha and Havlena](#)).

Převod regresního modelu do stavového tvaru

Často je vhodné dynamický regresní model realizovat ve stavové podobě. V dalším odstavci uvádíme možnosti, jak takový převod realizovat tak, aby dimenze stavu byla nejmenší možná.

1. způsob

Uvažujeme regresní model, který zapíšeme ve tvaru diferenční rovnice a pro jednoduchost bez šumu a bez přímé vazby vstupu na výstup

$$y_t = a_1y_{t-1} + \dots + a_ny_{t-n} + b_1u_{t-1} + b_2u_{t-2} + \dots + b_nu_{t-1},$$

což odpovídá tvaru diferenční rovnice s pravou stranou

$$y_t - a_1 y_{t-1} - \dots - a_n y_{t-n} = b_1 u_{t-1} + b_2 u_{t-2} + \dots + b_n u_{t-n}.$$

Ten lze vyjádřit operátorově

$$Ay_t = Bu_t,$$

kde $A = 1 - a_1 z^{-1} - a_2 z^{-2} - \dots - a_n z^{-n}$, $B = b_1 z^{-1} + \dots + b_n z^{-n}$ a z^{-i} je operátor i -krokového zpoždění $z^{-i} y_t = y_{t-i}$.

S využitím operátorového značení zavedeme novou proměnnou ξ_t takto

$$y_t = \frac{B}{A} u_t = B\xi_t \quad \text{kde} \quad \xi_t = \frac{1}{A} u_t.$$

Odtud dostaneme

$$A\xi_t = u_t \quad \text{a} \quad y_t = B\xi_t,$$

tedy zapsáno ve formě diferenčních rovnic máme

$$\xi_t - a_1 \xi_{t-1} - \dots - a_n \xi_{t-n} = u_t, \quad (4.1)$$

$$y_t = b_1 \xi_{t-1} + b_2 \xi_{t-2} + \dots + b_n \xi_{t-n}. \quad (4.2)$$

Stav x zavedeme jako vektor

$$x_{t-1} = [\xi_{t-1}, \xi_{t-2}, \dots, \xi_{t-n}]'.$$

Rovnici (4.1) pak můžeme psát ve tvaru

$$\begin{aligned} \xi_t &= a_1 x_{1;t-1} + a_2 x_{2;t-1} + \dots + a_n x_{n;t-1} + u_t = \\ &= [a_1, a_2, \dots, a_n] x_{t-1} + u_t. \end{aligned}$$

Podle definice veličiny x platí

$$x_t = \begin{bmatrix} a_1 & a_2 & \dots & a_{n-1} & a_n \\ 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} x_{t-1} + \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \\ 1 \end{bmatrix} u_t.$$

Rovnice (4.2) má tvar

$$y_t = [b_1, b_2, \dots, b_n] x_{t-1}.$$

Tak dostáváme ekvivalentní stavový model ve tvaru

$$\begin{aligned}x_t &= Mx_{t-1} + Nu_t, \\y_t &= Ax_{t-1},\end{aligned}$$

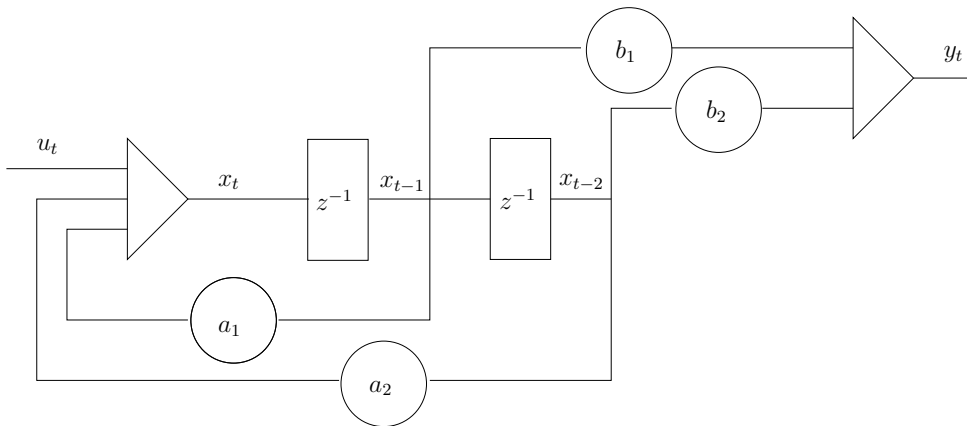
kde

$$M = \begin{bmatrix} a_1 & a_2 & \cdots & a_{n-1} & a_n \\ 1 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad N = \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \\ 1 \end{bmatrix}$$

a

$$A = [b_1, b_2, \dots, b_n].$$

Schéma pro tento převod je pro druhý řád modelu nakresleno na následujícím obrázku



2. způsob

Uvažujeme diferenční rovnici ve tvaru

$$y_t = a_1 y_{t-1} + b_1 u_{t-1} + a_2 y_{t-2} + b_2 u_{t-2} + \cdots + a_n y_{t-n} + b_n u_{t-n}.$$

Stav zavedeme takto

$$\begin{aligned}x_{n;t} &= a_n y_t + b_n u_t, \\x_{n-1;t} &= a_{n-1} y_t + b_{n-1} u_t + x_{n;t-1}, \\x_{n-2;t} &= a_{n-2} y_t + b_{n-2} u_t + x_{n-1;t-1}, \\&\dots \\x_{1;t} &= a_1 y_t + b_1 u_t + x_{2;t-1}\end{aligned}$$

a

$$y_t = x_{1;t-1}.$$

Za y_t dosadíme nahoru a obrátíme pořadí rovnic. Dostaneme

$$\begin{aligned} x_{1;t} &= a_1 x_{1;t-1} + x_{2;t-1} + b_1 u_t, \\ x_{2;t} &= a_2 x_{1;t-1} + x_{3;t-1} + b_2 u_t, \\ &\dots \\ x_{n-1;t} &= a_{n-1} x_{1;t-1} + x_{n;t-1} + b_{n-1} u_t, \\ x_{n;t} &= a_n x_{1;t-1} + b_n u_t. \end{aligned}$$

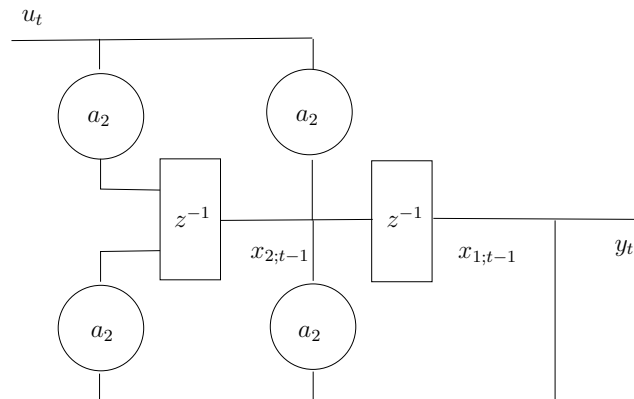
Maticový zápis je

$$\begin{aligned} x_t &= Mx_{t-1} + Nu_t, \\ y_t &= Ax_{t-1}, \end{aligned}$$

kde

$$M = \begin{bmatrix} a_1 & 1 & 0 & 0 & 0 \\ a_2 & 0 & 1 & 0 & 0 \\ \dots & 0 & 0 & 0 & 0 \\ a_{n-1} & 0 & 0 & 0 & 1 \\ a_n & 0 & 0 & 0 & 0 \end{bmatrix} \quad N = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_{n-1} \\ b_n \end{bmatrix} \quad A = [1, 0, \dots, 0].$$

Blokové schéma pro tuto stavovou realizaci je nakreslena v následujícím schématu



4.2 Nelineární model

V případě, kdy je pro popis systému třeba použít nelineární model, je situace podstatně složitější a výsledky odhadu stavu jsou zdaleka ne tak dobré, jako v lineárním případě.

Model je nyní zadán obecně podmíněnými hp (24) a (24), definovaný rovnicemi

$$\begin{aligned}x_{t+1} &= g(x_t, u_t) + w_t, \\y_t &= h(x_t, u_t) + v_t.\end{aligned}$$

Nelineární funkce g a h linearizujeme tak, že je aproximujeme Taylorovým rozvojem 1. řádu v bodovém odhadu z minulého kroku

$$\begin{aligned}g(x_t, u_t) &= g(\hat{x}_t, u_t) + \frac{\partial}{\partial x_t} g(\hat{x}_t, u_t) (x_t - \hat{x}_t), \\h(x_t, u_t) &= h(\hat{x}_t, u_t) + \frac{\partial}{\partial x_t} h(\hat{x}_t, u_t) (x_t - \hat{x}_t).\end{aligned}$$

Označíme: $g' = \frac{\partial}{\partial x_t} g(\hat{x}_t, u_t)$ a $h' = \frac{\partial}{\partial x_t} h(\hat{x}_t, u_t)$ linearizovaný model má tvar

$$x_{t+1} = g'x_t + [g - g'\hat{x}_t] + w_t, \quad (4.3)$$

$$y_t = h'x_t + [h - h'\hat{x}_t] + v_t \quad (4.4)$$

a tedy když označíme $M = g'$, $F = g - g'\hat{x}_t$, $A = h'$, $G = h - h'\hat{x}_t$, dostaneme stavový model ve tvaru

$$\begin{aligned}x_{t+1} &= Mx_t + F + w_t, \\y_t &= Ax_t + G + v_t,\end{aligned}$$

kde ovšem M , F , A , G jsou závislé na u_t .

Pro tento model již lze použít Kalmanův filtr. Program, realizující Kalmanův filtr je uveden na straně 2.2.

Vzorový příklad

Odhad stavu s nelineárním stavovým modelem budeme demonstrovat pro simulovaná data z modelu

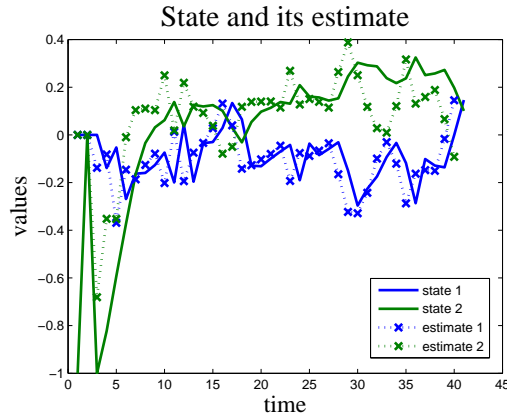
$$\begin{aligned}x_{1;t} &= x_{1;t-1}x_{2;t-1} + 0.1x_{2;t-1} - u_t + 0.01e_{1;t}, \\x_{2;t} &= 0.5x_{1;t-1} + 0.8x_{2;t-1} + u_t + 0.01e_{2;t}.\end{aligned}$$

Pro odhad byly použity kovarianční matice

$$R_w = 0.1E, \quad R_v = 0.1, \quad R_x = 1000E,$$

kde E je jednotková matice stupně 2.

Výsledky pokusu jsou na obrázku



Program

Program je následující

```

clc , clear all
addpath('funct ')
%% State estimation (nonlinear model)

nd=40; % length of data

ut=.1*(1-randn(1,nd)); % control
yt=zeros(1,nd);
x1=0; x2=-1; % initial state
xs=zeros(2,nd); xs(:,1)=[x1;x2];
% SIMULATION
for t=2:nd
    yt(t)=x1; % output equation
    xs(:,t+1)=[x1;x2]; % store the actual state variable
    x1=x1*x2+.1*x2-ut(t)+.01*randn; % state
    x2=.5*x1+.8*x2+ut(t)+.01*randn; % equations
end

xx=[0;0]; % initial state estimate
Rw=.1*eye(2); Rv=.1; Rx=1e3*eye(2); % model and initial covariances
xt=zeros(2,nd); rr=zeros(2,nd); rr(:,1)=diag(Rx);
% ESTIMATION
for t=2:nd
    % parameters of the linearized model
    M=[xx(2) xx(1)+.1; .5 .8];
    N=[-1;1];
    F=[-.1*xx(1)+(.1-xx(2))*xx(2);0];

```

```

A=[1 0];
B=0;
G=0;
% Kalman filter
[xx,xf,Rx,yp]=Kalman(xx,yt(t),ut(t),M,N,F,A,B,G,Rw,Rv,Rx);
xt(:,t)=xf; % stor actual state estimate
rr(:,t)=diag(Rx); % stor variance of noise estimate
end

% RESULTS
vStEstNL_R

```

Model s neznámými parametry

Kalmanův filtr funguje pro stavový model se známými parametry. Jestliže některé parametry jsou neznámé, zacházíme s nimi jako s neznámou veličinou, tj. se stavem. Do existujícího stavu přidáme všechny neznámé parametry. Tím se ale model, i když byl původně lineární, stane nelineárním - stavy, označující neznámé parametry se objeví jak v matici koeficientů a dostanou se tak do součinu se samotným stavem.

Ukážeme jednoduchý příklad:

Máme dynamický systém, popsaný modelem

$$\begin{aligned}x_{t+1} &= 0.8x_t + u_t + w_t, \\y_t &= \theta x_t + v_t.\end{aligned}$$

Zavedeme nový stav

$$\xi_t = \begin{bmatrix} x_t \\ \theta \end{bmatrix}$$

a nový model je

$$\begin{aligned}\begin{bmatrix} \xi_{1;t+1} \\ \xi_{2;t+1} \end{bmatrix} &= \begin{bmatrix} 0.8 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_{1;t} \\ \xi_{2;t} \end{bmatrix} + \begin{bmatrix} w_{1;t} \\ w_{2;t} \end{bmatrix}, \\y_t &= [\xi_{2;t}, 0] \begin{bmatrix} \xi_{1;t} \\ \xi_{2;t} \end{bmatrix} + v_t.\end{aligned}$$

Stavový model je lineární, výstupní model je

$$y_t = \xi_{1;t}\xi_{2;t} + v_t$$

a nelineární funkci $h = \xi_{1;t}\xi_{2;t}$ linearizujeme

$$\frac{\partial h}{\partial \xi_{1;t}} = \xi_{2;t}, \quad \frac{\partial h}{\partial \xi_{2;t}} = \xi_{1;t}.$$

Podle (48) pak je

$$y_t = \underbrace{[\xi_{2;t}, \xi_{1;t}]}_{\partial h / \partial \xi_1} \begin{bmatrix} \xi_{1;t} \\ \xi_{2;t} \end{bmatrix} + \left\{ \underbrace{\begin{bmatrix} \hat{\xi}_{2;t} \\ 0 \end{bmatrix}}_h - \underbrace{[\xi_{2;t}, \xi_{1;t}]}_{\partial h / \partial \xi_1} \underbrace{\begin{bmatrix} \hat{\xi}_{1;t} \\ \hat{\xi}_{2;t} \end{bmatrix}}_{\hat{\xi}_t} \right\} + v_t$$

Na stavovou i linearizovanou výstupní rovnici použijeme Kalmanův filtr, uvedený na straně 2.2.

Vzorový příklad

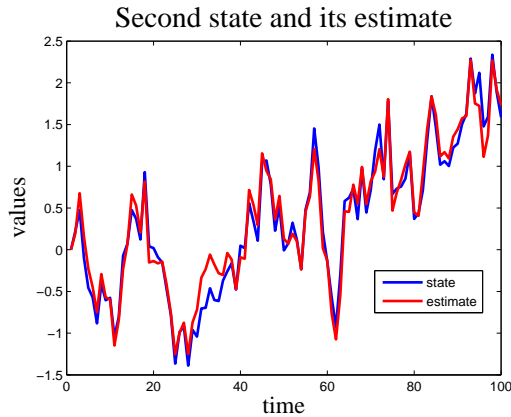
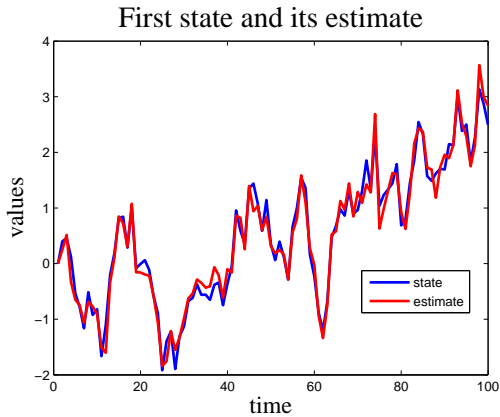
Neznámé parametry modelu řešíme společným odhadem stavu i parametrů. Cena za to je nelineární model. Situaci ukážeme na jednoduchém příkladě.

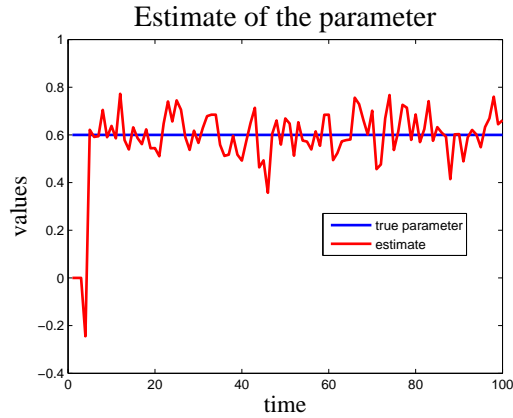
Budeme uvažovat stavový model s parametry

$$M = \begin{bmatrix} a & 0.5 \\ 0.1 & 0.8 \end{bmatrix}, \quad N = \begin{bmatrix} 0.5 \\ 0.4 \end{bmatrix}, \quad A = [1, 0], \quad r_v = 0.01, \quad r_w = 0.01 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Pro simulaci položíme $a = 0.6$ a v odhadu budeme tento parametr považovat za neznámý.

Výsledky experimentu jsou na obrázcích





Program

Detaily, jako je např. inicializace odhadu a celkové řešení příkladu je možno vysledovat z následujícího programu

```

clc, clear all, close all
addpath('funct')
%% State estimation (model with unknown parameter)

nd=100; % length of data
a=.6; % unknown model parameter
sM=[a .5 % simulation parametrs
    .1 .8];
sN=[.5; .4];
sA=[1 0];
rv=.01; rw=.01*eye(2); % noise variances

ut=randn(1,nd);
yt=zeros(1,nd);
xt=zeros(2,nd);
x=zeros(2,1); % initial state
% simulation
for t=1:nd
    xt(:,t)=x;
    yt(:,t)=sA*x+sqrt(rv)*randn;
    x=sM*x+sN*ut(:,t)+sqrt(rw)*randn(2,1);
end
xt(3,:)=a; % true parameter

Rv=.01; Rw=.01; % model covariance matrices
Rx=1e6*eye(3); % covariance matrix of state estimate

```

```

N=[.5; .4; 0]; % parameters of model extended by unknown
A=[1 0 0]; % parameter
G=0;
B=0;
x=zeros(3,1);
xe=zeros(3,nd);
% ESTIMATION
for t=1:nd
    M=[x(3) .5 0 % parameters of thr linearized model
        .1 .8 0
        0 0 1];
    dM=[x(3) .5 x(1)
        .1 .8 0
        0 0 1];
    F=(M-dM)*x;
    xe(:,t)=x;
    % Kalman filter
    [x,xf,Rx,yp]=Kalman(x,yt(t),ut(:,t),dM,N,F,A,B,G,Rw,Rv,Rx);
end

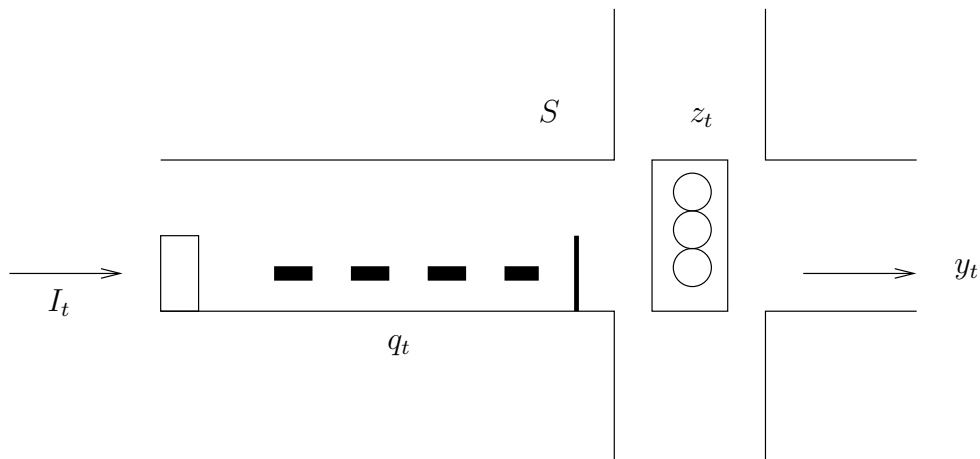
% RESULTS
vStEstPar_R

```

4.3 Aplikace

Úloha odhadu délky kolony v rameni křižovatky

Budeme se zabývat situací, kdy sledovaným systémem je jedno rameno řízené křižovatky tak, jak je to naznačeno na následujícím obrázku



Zleva přijíždí automobily s intenzitou I_t a jsou registrovány na magnetickém detektoru. V pěti-minutové periodě je zaznamenáván jejich počet. Vzhledem k řízení křižovatky se vytváří kolona o délce q_t , která stojí na stop čáře na vjezdu do prostoru křižovatky. Řízení křižovatky je pevné a pracuje s poměrem zelené z_t (vliv délky cyklu neuvažujeme). U ramene křižovatky předpokládáme pevný saturovaný tok S (počet aut, která projedou za periodu vzorkování při stálém zeleném signálu). Počet aut, která projedou za periodu vzorkování při poměru zelené z_t je potom Sz_t .

Délku kolony v rameni křižovatky považujeme za neměřitelnou a chceme ji odhadovat na základě známých parametrů křižovatky a měřených dat - vstupní a výstupní intenzity. Výstupní intenzitu měříme ihned za stop čarou, nebo můžeme uvažovat případ, kdy oba směry v křižovatce jsou jednosměrné a my výstupní intenzitu měříme v pravém rameni. Ve sledovaném rameni lze pozorovat dvě odlišné situace, které v čase t označíme indikátorem δ_t :

- kolona je - $\delta_t = 1$; to nastane, jestliže na konci zelené neprojde celá kolona a její konec je zachycen červenou, tj. kolona z minula + to, co přijelo je více než to, co projelo na zelenou

$$q_t + I_t > Sz_t,$$

- kolona není - $\delta_t = 0$; je situace, kdy na konci zelené žádná kolona nezůstane (tvoří se až na další červené), tedy

$$q_t + I_t \leq Sz_t.$$

Podle toho, ve které situaci se nacházíme, platí následující modely

Kolona je

– stavová rovnice říká, že to, co během periody vzorkování (5 min) přijelo muselo zase odjet, nebo zůstalo v koloně

$$q_{t+1} = q_t + I_t - Sz_t,$$

– výstupní rovnice

$$y_t = Sz_t.$$

Kolona není

– stavová rovnice vyjadřuje skutečnost, že na konci červené se vytvoří kolona aut, která během červené přijela

$$q_{t+1} = I_t(1 - z_t),$$

– výstupní rovnice je určuje počet automobilů, která během periody vyjedou jako počet těch, co byly v koloně na červenou a počet těch, co přijely a projely na zelenou (na konci zelené žádná kolona není)

$$y_t = q_t + I_t z_t.$$

Celkový model kolony v rameni křižovatky lze vyjádřit pomocí indikátoru δ_t spojením obou předchozích modelů. Celkový model, upravený tak, aby odpovídal stavovému modelu, je následující

$$q_{t+1} = Mq_t + Nz_t + F, \quad (4.5)$$

$$y_t = Aq_t + Bz_t, \quad (4.6)$$

kde se zavedením $\bar{\delta}_t = 1 - \delta_t$ je

$$M = \delta_t, \quad N = -\delta_t S - \bar{\delta}_t I_t, \quad F = I_t,$$

$$A = \bar{\delta}_t, \quad B = \delta_t S + \bar{\delta}_t I_t.$$

Poznámka

Všimněme si, že v případě, kdy kolona je, tedy $\delta_t = 1$ výstup y_t nezávisí na koloně ($A = 0$). To je průšvih, kterého si všimneme později. Teď jenom zkonstatujeme, že toto pozorování odpovídá skutečnosti. Jestliže je kolona, pak výjezd je vždy Sz_t ať je kolona dlouhá jakkoli.

Chceme-li se touto úlohou dále zabývat, musíme v první řadě vyřešit simulaci dat v tomto rameni.

Simulace dat

Pro simulaci dat se budeme opírat o reálná data. Jsou to intenzity provozu v nově otevřeném strahovském tunelu. Jsou měřeny v intervalu pěti minut a zahrnují období přibližně čtyř týdnů. Z těchto dat získáme tzv. denní průběh intenzity tak, že zprůměrujeme a vyhladíme intenzity ze všech všedních dnů (nebo víkendů). Vyhlazení provedeme tak že místo naměřené hodnoty bereme průměr z deseti hodnot vlevo a vpravo (průměrování na okně). Pro všední dny tak dostaneme denní průběh, který jsme již uvedli na straně 3.3. Tento denní průběh intenzity je jakousi průměrnou křivkou, která popisuje, jak se ve všední den chová doprava. Vlastní intenzity pro jednotlivé dny budeme simulovat jako náhodnou posloupnost s exponenciálně rozdělenými časovými mezerami. Jedná se o Poissonův proces, známý zejména z teorie hromadné obsluhy. Posloupnost časových mezer mezi automobily označíme ϑ_t a platí pro ni

$$f(\vartheta_t | a, A) = a \exp\{-(a - A)\vartheta_t\}, \quad A \geq a, \quad \vartheta_t \geq A.$$

Distribuční funkce je

$$F(\vartheta_t) = 1 - \exp\{-(a - A)\vartheta_t\}, \quad \vartheta_t \geq A.$$

Hodnoty z tohoto rozdělení jsou ϑ_t , když hodnoty distribuční funkce generujeme jako rovnoměrné rozdělení na intervalu $U \in (0, 1)$, tj.

$$U = 1 - \exp \{-(a - A) \vartheta_t\},$$

odkud je

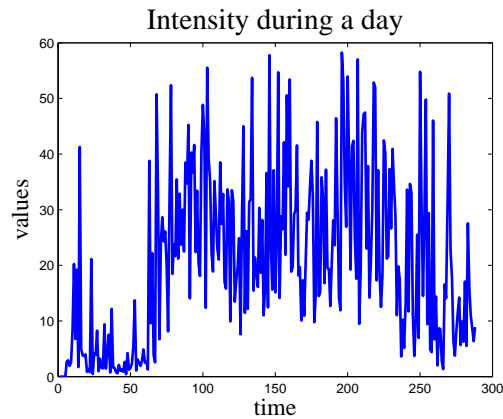
$$\vartheta_t = A - \ln \{U\} / a,$$

kde a je intenzita provozu (počet aut za periodu) a A je minimální interval (v jednotce jedna perioda) dvou následujících aut. Dále jsme využili toho, že $1 - U$ generuje hodnoty ze stejného rozdělení jako U .

Intenzita a se během dne mění a v každém řasovém okamžiku ji nastavíme na příslušnou hodnotu denního průběhu, který jsme zkonstruovali průměrováním konkrétních denních průběhů a jejich vyhlazením na okně.

Posloupnost počtu aut, která projela za periodu (tedy intenzit) dostaneme jako posloupnost převrácených hodnot časových mezer ϑ_t .

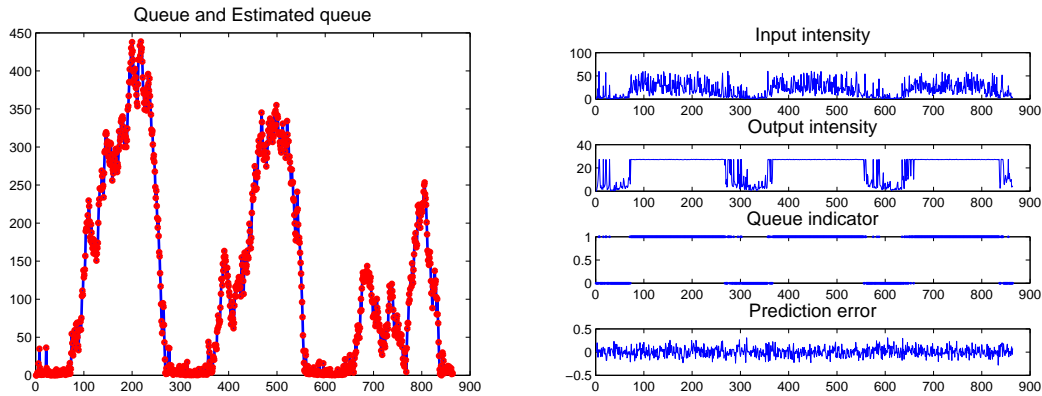
Příklad výsledné simulace je na dalším obrázku



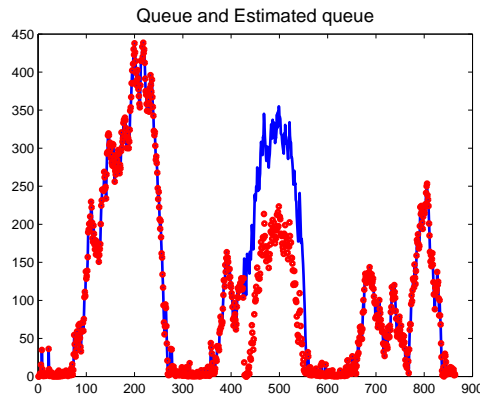
Odhad stavu s lineárním modelem

Pro odhad se pokusíme použít Kalmanův filtr (KF). Slovo pokusíme jsme použili pro to, abychom naznačili, že to nebude tak jednoduché.

Parametry (matice) stavového modelu pro KF jsou uvedeny v (4.5) a (4.6). Výsledek odhadu je na následujících obrázcích



Vlevo je průběh délky kolony a jeho odhad, vpravo další veličiny. Z levého obrázku je parné, že odhad je velmi dobrý. Nicméně, potíží nastane, jestliže nasimulujeme velkou poruchu v odhadu stavu - např. výpadek odhadu, kdy v určitou dobu se odhad stavu vynuluje. V našem případě umístíme poruchu do času $t = 430$ a po 5 kroků nulujeme odhad stavu. Výsledek je na dalším obrázku



Vidíme, že způsobená chyba trvá po celý den a odhad se “chyť” až další den. Důvodem, proč KF nemá tendenci podle měřených výstupů chybu okamžitě kompenzovat, je ten, že náš model (v režimu kdy kolona je) není pozorovatelný. Tento fakt je zřejmý již na první pohled. Výstup modelu je $y_t = Sz_t$ a tedy, nezávisí na stavu. Tedy z kolony odjíždí stále stejné množství automobilů, ať je kolona jakkoli dlouhá (pokud je). Teprve když kolona klesne k nule (tedy kolona není) platí, $y_t = q_t + I_t z_t$ a odhad kolony lze srovnat s naměřeným výstupem.

Matice pozorovatelnosti pro náš příklad je

$$P = [A', M'A']'$$

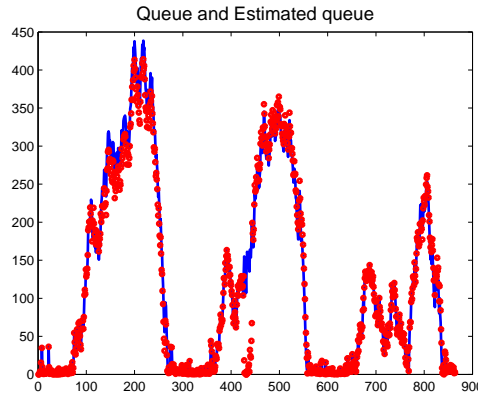
a aby byl model pozorovatelný, pak tato matice musí mít hodnost 2.

Po dosažení $M = \delta = 1$ a $A = 1 - \delta = 0$ pro případ kdy kolona je ($\delta = 1$) dostaneme matici

$$P = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

s hodnotí menší než 2.

Určitým řešením této situace je zvolit nějaký referenční bod, na kterém se dozvíme skutečnou délku kolony. Například využít vzdálený detektor (pokud existuje a je na vhodném místě) a pomocí něho zjišťovat, zda konec kolony je právě na tomto místě. V okamžiku, kdy toto nastane je možno odhad kolony kalibrovat a tím eliminovat případnou odchylku. V našem příkladě jsme takový detektor postavili do vzdálenosti 200. Výsledek je na dalším obrázku



Z obrázku je patrné, že porucha se při dosažení délky kolony 200 ihned kompenzuje.

Odhad stavu s modelem s neznámým parametrem

Pokud připustíme, že saturovaný tok S neznáme, nebo dokonce počítáme s realističtější situací, že se tento saturovaný tok se v čase mění, použijeme opět stavový model podle (4.5), (4.6), ale s neznámým S , které budeme odhadovat. Použijeme pro to rozšíření stavu a linearizaci modelu pomocí Taylorova rozvoje s postupem podle kapitoly 4.2.

Zavedeme rozšířený stav

$$x_t = \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} = \begin{bmatrix} q_t \\ S \end{bmatrix}$$

a stavový model píšeme ve tvaru

$$\begin{bmatrix} x_{1;t+1} \\ x_{2;t+1} \end{bmatrix} = \begin{bmatrix} M & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \begin{bmatrix} N \\ 0 \end{bmatrix} z_t + \begin{bmatrix} F \\ 0 \end{bmatrix},$$

$$y_t = [A, 0] \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + Bz_t.$$

Po dosazení z (4.5), (4.6) a dosazení za $S = x_{2;t}$ dostaneme podle Kapitoly 4.2 funkce g a h ve tvaru

$$\begin{aligned} g &= \begin{bmatrix} \delta_t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \begin{bmatrix} -(\delta_t x_{2;t} + \bar{\delta}_t I_t) \\ 0 \end{bmatrix} z_t + \begin{bmatrix} I_t \\ 0 \end{bmatrix}, \\ h &= [\bar{\delta}_t, 0] \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + (\delta_t x_{2;t} + \bar{\delta}_t I_t) z_t. \end{aligned}$$

Abychom získali tvar stavového modelu, spojíme stavy z první a druhé části modelu

$$\begin{aligned} g &= \begin{bmatrix} \delta_t & -\delta_t z_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \begin{bmatrix} -\bar{\delta}_t I_t \\ 0 \end{bmatrix} z_t + \begin{bmatrix} I_t \\ 0 \end{bmatrix}, \\ h &= [\bar{\delta}_t, \delta_t z_t] \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \bar{\delta}_t I_t z_t. \end{aligned}$$

Vidíme, že obě funkce jsou nyní lineární ve stavu a lze tedy přímo použít Kalmanův filtr s parametry modelu

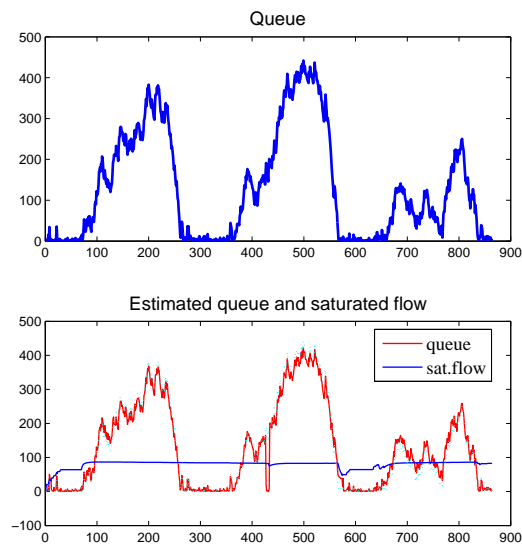
$$M = \begin{bmatrix} \delta_t & -\delta_t z_t \\ 0 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} I_t (1 - \bar{\delta}_t z_t) \\ 0 \end{bmatrix}, \quad A = [\bar{\delta}_t, \delta_t z_t], \quad G = \bar{\delta}_t I_t z_t.$$

Poznámka

Model, který jsme dostali neodpovídá přesně běžnému lineárnímu stavovému modelu. Řízení nemá svou obvyklou část Nz_t ve stavové a Bz_t ve výstupní části modelu, ale je propletené v maticích modelu. To ale nevadí a co je důležité, je to, že stav vystupuje v modelu lineárně Mx_t a Ax_t .

Výsledek

Výsledky příkladu jsou na následujících obrázcích



Program

```

clc, clear all, close all
addpath 'funct'
rand('seed',13579),randn('seed',97531')
% Simulation of queue length in simple crossroad
% with simultaneous estimation of saturated flow
% (restart of queue est. on remote detector)

load df                                % load daily course
nd=3*288;                               % number of data (288 = 1 day)

% simulation
q=zeros(1,nd);
xi=zeros(2,nd); xi(2,2)=20;
et=zeros(1,nd);
ed=drift(nd);

S=85;                                   % saturation flow
z=.32*ones(1,nd);                       % constant green proportion
Rw=.01*eye(2); Rv=.1;                   % state and output covariances
Rx=diag([1 .1]);                         % state est. covariance
for t=2:nd
    Sd=S+1*5*ed(t); Sdt(t)=Sd;         % drift added to sat.flow
    % sim. of input intensity

```

```

lam=df(mod(t,288)+1);           % current intensity
a=5/300;                        % minimal gap between cars
                                % for period time unit
I(t)=genExp(lam,a);             % simulated traff. intensity
% sim. of queue
del=q(t)+I(t)>Sd*z(t);         % queue indicator
if del, q(t+1)=q(t)+I(t)-Sd*z(t);
else, q(t+1)=I(t)*(1-z(t));
end
if del, y(t)=Sd*z(t);
else y(t)=q(t)+I(t)*z(t);
end,
y(t)=y(t)+.1*randn;           % noise added to output
d(t)=del;

% state-space model
d1=xi(1,t)+I(t)>S*z(t);       % estimated queue indicator
d0=1-d1;

M=[d1 -d1*z(t);
   0   1];
N=zeros(2,1);
F=[I(t)*(1-d0*z(t))
   0];
A=[d0 d1*z(t)];
B=0;
G=d0*I(t)*z(t);

[ xi(:,t+1),xf(:,t),Rx,yp,ep]=...
Kalman(xi(:,t),y(t),z(t),M,N,F,A,B,G,Rw,Rv,Rx); % KF
xi(t+1)=max(xi(t+1),0);        % nonzero queue
et(t)=ep;                      % remember pred. error
if abs(t-430)<5; xi(1,t+1)=0; end % error in queue est.
if abs(q(t)-200)<3, xi(1,t+1)=q(t); end % restart of queue est.
if fix(t/288)*288==t, xi(1,t+1)=0; end
end

% Results
aXrdsNL_R

```

Úloha nelineární filtrace

Budeme uvažovat úlohu, při které sledujeme počet aut na parkovišti. Na parkoviště přijíždí měřený tok automobilů. Parkovné za stání je progresivní s množstvím zaparkova-

ných automobilů. Proto také intenzita výjezdů je úměrná zaplnění parkoviště. Výjezdy jsou dva. My sledujeme jen jeden a předpokládáme, že tímto výjezdem odjíždí konstantní poměr odjíždějících automobilů. Cílem úlohy je odhadovat počet zaparkovaných automobilů.

Označíme

ξ_t je množství zaparkovaných automobilů v čase t (stav),

v_t je vjezdová intenzita v čase t (měřená externí porucha),

y_t je měřený výjezd automobilů (výstup).

Pro vjezdovou intenzitu použijeme simulaci z předchozího příkladu, která respektuje denní průběh intenzity.

Model má tvar

$$\begin{aligned}\xi_{t+1} &= \alpha\xi_t + v_t + e_t, \\ y_t &= \beta\xi_t + \epsilon_t.\end{aligned}$$

Známé parametry modelu

Pokud bychom znali koeficienty α a β , byl by model lineární a pro odhad by bylo přímo použít Kalmanův filtr.

Program

```
clc, clear all, close all
addpath 'funct'
% Estimation of nonlinear state-space model

load df % load of the daily course (DC)
nd=288*3; % length of data
x(1)=0; % initial state for simulation
a=.6; % parameters for
b=.5; % simulation
Rw=.001; % model covariance
Rv=.0001; % matrices
Rx=10; % initial state cov. matrix
xi(1,1)=0; % initial state for estimation

% TIMR LOOP
for t=1:nd
    % simulation
    lam=df(mod(t,288)+1); % int. expectat. from DC
```

```

v(t)=genExp(lam,b);           % exponential distribution
x(t+1)=a*x(t)+v(t)+.0001*randn; % state mmodel
y(t)=b*x(t)+.0001*randn;     % output model

% estimation
M=a; N=0; F=v(t);
A=b; B=0; G=0;
[xi(:,t+1),xf(:,t),Rx,yp,ep(t)]=...
    Kalman(xi(:,t),y(t),0,M,N,F,A,B,G,Rw,Rv,Rx); % Kalman filter
xi(1,t+1)=max(xi(1,t+1),0); % restriction to nonnegativity
end

% RESULTS
s=1:nd;
set(figure(1),'position',[300 100 500 400])
plot(s,x(s),s,xi(1,s),'.')
legend('simulated queue','estimated queue',0)
title('Estimation of the queue length')

```

Neznámý parametr ve stavové rovnici

Nejprve budeme předpokládat, že neznáme parametr α ve stavové rovnici. Zavedeme nový stav $x_t = [x_{1;t}, x_{2;t}]' = [\xi_t, \alpha]'$ dostáváme rozšířený model s deterministickou částí

$$\begin{bmatrix} x_{1;t+1} \\ x_{2;t+1} \end{bmatrix} = \begin{bmatrix} x_{2;t} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \begin{bmatrix} v_t \\ 0 \end{bmatrix},$$

$$y_t = [\beta, 0] \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix}.$$

Jedná se o tři rovnice (2 stavové a jedna výstupní) z nichž pouze první je nelineární. Tu budeme rozvíjet pomocí Taylorova rozvoje. Ve shodě s teorií označíme její pravou stranu jako funkci g_1

$$g_1 = x_{2;t}x_{1;t} + v_t$$

a dále derivace jsou

$$\frac{\partial g_1}{\partial x_1} = x_{2;t}, \quad \frac{\partial g_1}{\partial x_2} = x_{1;t}.$$

Rozvoj prvního řádu je

$$\begin{aligned} g_1(\hat{x}) &\doteq \hat{x}_{2;t}\hat{x}_{1;t} + v_t + [\hat{x}_{2;t}, \hat{x}_{1;t}] \left(\begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} - \begin{bmatrix} \hat{x}_{1;t} \\ \hat{x}_{2;t} \end{bmatrix} \right) = \\ &= [\hat{x}_{2;t}, \hat{x}_{1;t}] \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \hat{x}_{2;t}\hat{x}_{1;t} + v_t - [\hat{x}_{2;t}, \hat{x}_{1;t}] \begin{bmatrix} \hat{x}_{1;t} \\ \hat{x}_{2;t} \end{bmatrix}. \end{aligned}$$

Doplňme ostatní lineární rovnice a dostaneme linearizovaný model

$$\begin{bmatrix} x_{1;t+1} \\ x_{2;t+1} \end{bmatrix} = \begin{bmatrix} \hat{x}_{2;t} & \hat{x}_{1;t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \begin{bmatrix} \hat{x}_{2;t}\hat{x}_{1;t} + v_t - [\hat{x}_{2;t}, \hat{x}_{1;t}] \begin{bmatrix} \hat{x}_{1;t} \\ \hat{x}_{2;t} \end{bmatrix} \\ 0 \end{bmatrix},$$

$$y_t = \beta x_t.$$

Po malé úpravě v absolutním členu stavové rovnice dostaneme matice lineárního modelu

$$M = \begin{bmatrix} \hat{x}_{2;t} & \hat{x}_{1;t} \\ 0 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} v_t - \hat{x}_{2;t}\hat{x}_{1;t} \\ 0 \end{bmatrix}, \quad A = \beta.$$

Program

```

clc, clear all, close all
addpath 'funct'
rand('seed',13579),randn('seed',97531')
% Estimation of nonlinear state-space model with the
% unknown parameter in state equation
% The model is
% x(t+1) = a.x(t)+v(t)+w(t)      - a is unknown parameter
% y(t) = 0.5x(t)+v(t)

load df                                % load daily course
nd=288*1;                               % length of data
x(1)=0;                                 % initial state
a=.6;                                   % known parameter
b=.5;                                   % unknown parameter
Rw=.01*eye(2);                          % KF
Rv=.001;                                 % covariance
Rx=diag([10 1e-80]);                   % matrices
xi(1,1)=0; xi(2,1)=.2;                 % initial state estimate

for t=1:nd
    % simulation
    lam=df(mod(t,288)+1);               % expectation
    v(t)=genExp(lam,.02);                % exponential distribution
    x(t+1)=a*x(t)+v(t)+.1*randn;        % state-space
    y(t)=b*x(t)+.01*randn;              % model
    y(t)=max(y(t),.01);                  % restriction

    % filtration
    M=[xi(2,t) xi(1,t)];                 % model

```



```

    0 1];; % matrices
N=zeros(2,1); %
F=[v(t)-xi(1,t)*xi(2,t); 0]; %
A=[b 0]; %
B=0; % model
G=0; % matrices
[xi(:,t+1),xf(:,t),Rx,yp,ep(t)]=...
    Kalman(xi(:,t),y(t),0,M,N,F,A,B,G,Rw,Rv,Rx); % KF
xi(2,t+1)=max(min(xi(2,t+1),1),0); % restriction
end

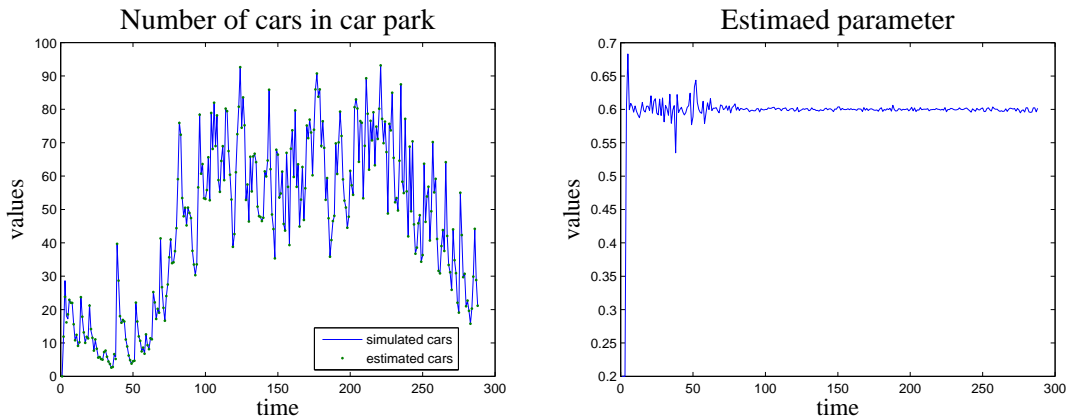
```

```

% RESULTS
aParkNLKp2_R

```

Výsledky



Neznámý parametr ve výstupní rovnici

V tomto případě budeme za neznámý parametr považovat koeficient β z výstupní rovnice stavového modelu. Dostaneme

$$\begin{bmatrix} x_{1;t+1} \\ x_{2;t+1} \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \begin{bmatrix} v_t \\ 0 \end{bmatrix},$$

$$y_t = [x_{2;t}, 0] \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix}.$$

Stavové rovnice jsou nyní lineární a linearizovat budeme výstupní rovnici. Její pravou stranu v minulém odhadu stavu označíme h

$$h = \hat{x}_{2;t} \hat{x}_{1;t}$$

a derivace v minulém odhadu stavu jsou

$$\frac{\partial h}{\partial x_1} = \hat{x}_{2;t}, \quad \frac{\partial h}{\partial x_2} = \hat{x}_{1;t}.$$

Rozvoj pravé strany výstupná rovnice bude

$$h \doteq \hat{x}_{2;t}\hat{x}_{1;t} + [\hat{x}_{2;t}, \hat{x}_{1;t}] \left(\begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} - \begin{bmatrix} \hat{x}_{1;t} \\ \hat{x}_{2;t} \end{bmatrix} \right) =$$

$$[\hat{x}_{2;t}, \hat{x}_{1;t}] \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \hat{x}_{2;t}\hat{x}_{1;t} - [\hat{x}_{2;t}, \hat{x}_{1;t}] \begin{bmatrix} \hat{x}_{1;t} \\ \hat{x}_{2;t} \end{bmatrix}.$$

Celý linearizovaný model bude

$$\begin{bmatrix} x_{1;t+1} \\ x_{2;t+1} \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \begin{bmatrix} v_t \\ 0 \end{bmatrix},$$

$$y_t = [\hat{x}_{2;t}, \hat{x}_{1;t}] \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} - \hat{x}_{1;t}\hat{x}_{2;t},$$

odkud matice modelu jsou

$$M = \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} v_t \\ 0 \end{bmatrix}, \quad A = [\hat{x}_{2;t}, \hat{x}_{1;t}], \quad G = -\hat{x}_{1;t}\hat{x}_{2;t}.$$

Program

```

clc, clear all, close all
addpath 'funct'
% Estimation of nonlinear state-space model with the
% unknown parameter in output equation
% The model is
% x(t+1) = 0.6x(t)+v(t)+w(t)
% y(t)    = b.x(t)+v(t)          - b is unknown parameter

load df                                % load of daily course
nd=288*3;                               % length of data
x(1)=0;                                 % initial state
a=.6;                                   % known parameter
b=.5;                                   % unknown parameter
Rw=.01*eye(2);                          % KF
Rv=.001;                                 % covariance
Rx=diag([10 1e-80]);                    % matrices
xi(1,1)=0; xi(2,1)=.2;                  % initial state estimate

```

```

for t=1:nd
% simulation
lam=df(mod(t,288)+1); % expectation
v(t)=genExp(lam,b); % exponential distribution
x(t+1)=a*x(t)+v(t)+.1*randn; % state-space
y(t)=b*x(t)+.01*randn; % model

% filtration
M=[a 0; % model matrices
 0 1];;
N=zeros(2,1); % model matrices
F=[v(t); 0]; % model matrices
dh=[xi(2,t) xi(1,t)]; % derivative of h
A=dh; % model matrices
B=0; % model matrices
G=xi(1,t)*xi(2,t)-dh*xi(:,t); % model matrices
[xi(:,t+1),xf(:,t),Rx,yp,ep(t)]=...
Kalman(xi(:,t),y(t),0,M,N,F,A,B,G,Rw,Rv,Rx); % KF
xi(2,t+1)=max(min(xi(2,t+1),1),0);
end

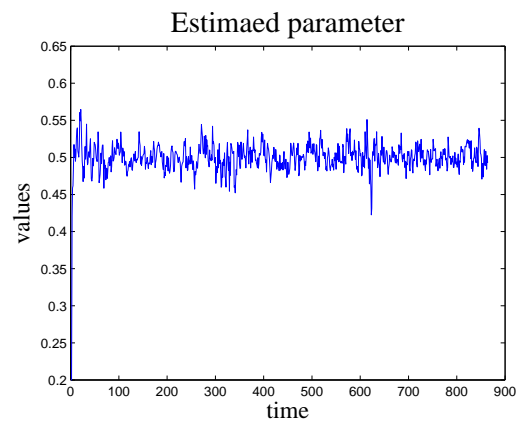
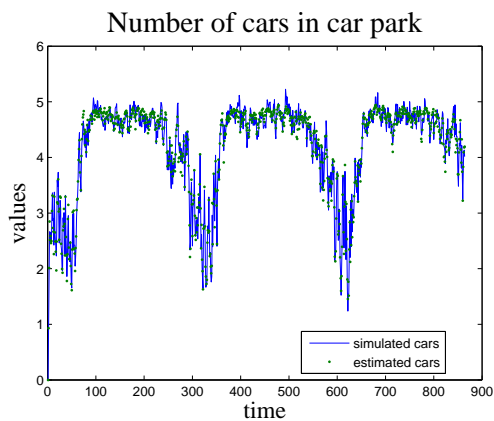
```

```

% RESULTS
aParkNLKp1_R

```

Výsledky



5 Testy hypotéz

Hodiny: 10

Aplikace: test nehomovosti kruhového objezdu podle skoronehod.

5.1 Hypotézy o platnosti modelu

Nacházíme se v situaci, kdy máme konečný počet n_H různých hypotéz H_i , $i = 1, 2, \dots, n_H$ o modelu systému. Na prokázání nejpravděpodobnější hypotézy použijeme apriorní informaci $d(0)$ a změřený vzorek dat $d(1:N) = \{d_t\}_{t=1}^N$.

Modely podle jednotlivých hypotéz jsou

$$f_i(d_t | \psi_{i;t}, \Theta_i), \quad i = 1, 2, \dots, n_H \quad (5.1)$$

a mohou se lišit ve struktuře a parametrech. Parametry jednotlivých modelů mohou být zadány číselně nebo mohou patřit do určité množiny. Například: uvažujeme tři hypotézy a alternativním modelem $f(y_t | \alpha) = \alpha^{y_t} (1 - \alpha)^{1 - y_t}$, $y_t \in \{0, 1\}$. Jednotlivé hypotézy tvrdí $H_1 : \alpha = 0.5$, $H_2 : \alpha < 0.5$ a $H_3 : \alpha > 0.5$.

Cílem bayesovského testování je zjistit, která z hypotéz H_i , $i = 1, 2, \dots, n_H$, o modelu má největší pravděpodobnost při použití apriorní informace a měřených dat.

Poznámka

Na rozdíl od klasického testování hypotéz, kdy se nulovou hypotézou snažíme popřít pomocí alternativní a jediný výsledek který můžeme přijmout je popření nulové hypotézy (nic pozitivního nelze prokázat), v případě bayesovského testování dostáváme pozitivní výsledek - přijímáme nejpravděpodobnější hypotézu.

Cílem testování je určit, která z hypotéz má největší pravděpodobnost P_i :

$$P_i = f(H_i | d(N)), \quad i = 1, 2, \dots, n_H. \quad (5.2)$$

Tyto pravděpodobnosti chceme vyjádřit pomocí modelů (5.1) a apriorní informace o těchto modelech a jednotlivých hypotézách. Použitím Bayesova vzorce dostaneme

$$P_i \propto f(d(1:N) | d(0), H_i) f(H_i | d(0)).$$

Druhá hp $f(H_i | d(0))$ v předchozím výrazu je apriorní hp pro hypotézy, kterou určíme na základě apriorní znalosti.

První hp je sdružená hp dat z výběru popsaná modelem podle i -té hypotézy H_i

$$f(d(1:N) | d(0), H_i) = \prod_{t=1}^N f_i(d_t | \psi_{i;t}),$$

kde f_i značí model podle i -té hypotézy H_i .

Modely, které tato sdružená hp definuje jsou v neparametrickém tvaru. Takové modely nemáme k dispozici. Proto musíme do sdružené hp zavést parametry

$$f(d(1:N)|d(0), H_i) = \int_{\Theta^*} f(d(1:N)|d(0), \Theta_i, H_i) f(\Theta_i|d(0)) d\Theta_i.$$

Druhá hp na pravé straně je apriorní hp pro parametry (modely jsou zadány buď s bodovým parametrem, nebo s oblastí přípustných hodnot nad kterou definujeme jejich apriorní rozdělení).

První hp můžeme nyní rozvést podle řetězového pravidla

$$f(d(1:N)|d(0), \Theta_i, H_i) = \prod_{t=1}^N f_i(d_t|\psi_{i;t}, \Theta_i),$$

kde opět f_i je model podle H_i .

Hustota pravděpodobnosti P_i pro H_i podle (5.2) po dosazení má tvar

$$P_i = f(H_i|d(N)) \propto \int_{\Theta^*} \underbrace{\prod_{t=1}^N f_i(d_t|\psi_{i;t}, \Theta_i)}_{L_i(\Theta_i)} f(\Theta_i|d(0)) d\Theta_i f(H_i|d(0)),$$

kde $L_i(\Theta_i) = \prod_{t=1}^N f_i(d_t|\psi_{i;t}, \Theta_i)$ je likelihood pro model podle hypotézy H_i .

Hodnoty které spočteme nejsou normovány. Pokud chceme opravdu dostat pravděpodobnostní funkci P_i pro hypotézy, normujeme. Pro rozhodování o správné hypotéze to ale není nutné. Jako správnou hypotézu H_{opt} volíme tu, pro kterou jsme vypočetli maximální hodnotu (a tedy i pravděpodobnost)

$$H_{opt} = H_i : i = \arg \max \{P_j\}_{j=1}^{n_H}.$$

Vzorový příklad

Teorii budeme demonstrovat na příkladě, kdy testujeme model s exponenciálním rozdělením

$$f(y_t|a) = a \exp\{-ay_t\}, \quad a > 0, \quad y_t \geq 0.$$

Hypotézy o tomto modelu jsou

$$H_1 : a < a_0,$$

$$H_2 : a = a_0,$$

$$H_3 : a > a_0.$$

Na základě datového vzorku y_1, y_2, \dots, y_N máme rozhodnout, která z hypotéz je pravdivá.

Řešení:

Likelihood je

$$L(a) = \prod_{t=1}^N a \exp\{-ay_t\} = a^N \exp\{-aS\}, \quad (5.3)$$

kde $S = \sum_{t=1}^N y_t$.

Pro hypotézy obecně platí

$$P_i = \int_{\Theta_i^*} L(\Theta_i) f(\Theta_i|d(0)) d\Theta_i f(H_i|d(0)),$$

kde L_i je likelihood pro model podle hypotézy H_i , $f(\Theta_i|d(0))$ je apriorní hp pro odhad parametrů modelu podle hypotézy H_i a $f(H_i|d(0))$ je apriorní pravděpodobnost i -té hypotézy. Likelihood jsme obecně vyjádřili v (5.3), pravděpodobnosti hypotéz jsou čísla. Podle hypotézy H_2 vybíráme jedinou možnost parametru a to $a = a_0$. Odpovídající apriorní hp bude mít tedy tvar Diracovy funkce $\delta(a, a_0)$. Pro volbu apriorních hp odhadu parametrů pro zbylé hypotézy vyjdeme z modelu $a \exp\{-ay_0\}$, kde y_0 je apriorní hodnota výstupu. Pro konstrukci apriorní hp musíme normovat. Pro H_3 dostaneme

$$f(y|y_0) = \frac{y_0^2}{k} a \exp\{-ay_0\},$$

kde $k = (1 + a_0 y_0) \exp\{-a_0 y_0\}$ a pro H_1

$$f(y|y_0) = \frac{y_0^2}{1-k} a \exp\{-ay_0\}.$$

Zabudování této apriorní informace znamená: zvýšit N o jedna (jako přidaná jedna data), k S přičíst y_0 (hodnota dat) a integrál násobíme výrazem $\frac{y_0^2}{1-k}$ pro H_1 respektive $\frac{y_0^2}{k}$ pro H_2 .

Dostaneme

$$\begin{aligned} P_1 &= \frac{y_0^2}{1-k} \int_0^{a_0} a^N \exp\{-aS\} da \frac{1}{3}, \\ P_2 &= a_0^N \exp\{-a_0 S\} \frac{1}{3}, \\ P_3 &= \frac{y_0^2}{k} \int_{a_0}^{\infty} a^N \exp\{-aS\} da \frac{1}{3}. \end{aligned}$$

Potřebujeme ještě spočítat integrál $\int_0^x a^N \exp\{-aS\} da$. Po substituci $aS = b$, $S da = db$ dostaneme

$$\int_0^x a^N \exp\{-aS\} da = \frac{1}{S^{N+1}} \gamma(N+1, xS),$$

kde γ je neúplná gama funkce a x je mez integrálu (tady $x = 1$).

Výpočet

$$\int_0^x a^N \exp\{-aS\} da = |aS = b, Sda = db| =$$

$$= \int_0^{xS} \left(\frac{b}{S}\right)^N \exp\{-b\} \frac{db}{S} = \frac{1}{S^{N+1}} \int_0^{xS} b^N \exp\{-b\} db = \frac{\gamma(N+1, xS)}{S^{N+1}}.$$

Výsledky testování pro simulovanou hodnotu $a = 2$ a hranici modelů $a_0 = 2$ a velikost výběru 200 jsou

$$P = [0.2463, 0.6801, 0.0736],$$

kde P jsou pravděpodobnosti pro jednotlivé hypotézy. Odtud skutečně vítězí druhá hypotéza - tedy rovnost.

Testovaný parametr je převrácenou hodnotou střední hodnoty rozdělení. Bodový odhad střední hodnoty (průměr z datového vzorku) je $\bar{y} = 0.5297$ a jeho převrácená hodnota $\hat{a} = \frac{1}{\bar{y}} = 1.8877$. Vidíme tedy, že neurčitost v pravděpodobnostech P je dána neurčitostí dat.

Program

```

clc, clear all, close all
% Testing of hypotheses for exponential model
%   !!! gammainc is int_x^inf (tail) !!!

% Simulation
nd=200;
a=2.2; % simulated parameter
y=-log(rand(1,nd))/a; % data

% Testing
a0=2; % parameter for hypotheses
y0=1/a0; % prior y (equal to H2)
n0=1; % number of prior data items
S=sum(y)+y0; % statistics
N=nd+n0; % statistics

k=(a0*y0+1)*exp(-a0*y0); % from prior data
I=(N+1)*log(S)-log(y0^2); % log constant
g=gammaln(N+1,a0*S); % incomplete gamma function
I1=gammaln(N+1)+log(1-g)-I-log(1-k); % log gama for H1
Lh=N*log(a0)-a0*S; % likelihood

```

```

I3=gammaIln(N+1)+log(g)-I-log(k);      % log gama for H3

lnP(1)=I1-log(3);                      % logarithms of
lnP(2)=Lh-log(3);                      % probabilities of
lnP(3)=I3-log(3);                      % hypotheses
pp=lnP-max(lnP);                       % normalization of logs
P=exp(pp);                              % exponential
P=P/sum(P);                             % normalization sum=1

% RESULTS
disp('Probabilities of hypotheses '),P
[xxx opt]=max(P);
fprintf('\nThe hypothesis %d wins\n',opt)

```

5.2 Aplikace

Test bezpečnosti kruhového objezdu

Na jednom rameni kruhového objezdu sledujeme výjimečné situace, které vybočují z hranic běžného provozu nebo při kterých mohlo dojít k nehodě (tzv. skoro-nehody). Jako skoro-nehody bereme např. prudké brzdění, náhlé vybočení z daného směru, nadměrné přiblížení vozidel a podobně. Usuzujeme, že čím více takových skoro-nehod nastane, tím je dané místo nebezpečnější a volá po nějaké opravě.

Model, popisující dané místo bude vyjadřovat podíl skoro-nehod v běžném provozu křižovatky, tedy

$$f(d_t|\alpha) = \alpha^{d_t} (1 - \alpha)^{1-d_t}, \quad (5.4)$$

kde $d_t \in \{0, 1\}$ charakterizuje jízdu t -tého vozidla $d_t = 0$ OK, $d_t = 1$ skoro-nehoda, (skoro-nehody mohou být i odstupňovány); $\alpha \in (0, 1)$ je poměr skoro-nehod v provozu H_j .

Pro testování hypotéz budeme uvažovat 3 modely se stejnou strukturou, ale lišící se v parametrech

1. H_1 : $\alpha = \alpha_0$, kde α_0 je daná hranice mezi bezpečím a nebezpečím,
2. H_2 : $\alpha < \alpha_0$, tj. místo je bezpečné,
3. H_3 : $\alpha > \alpha_0$, tj. místo je nebezpečné.

Podle teoretického úvodu budeme porovnávat pravděpodobnosti jednotlivých hypotéz $P_i = f(H_i|d(N))$ pro $i = 1, 2, 3$ a datový vzorek $d(N)$ kde N je množství změřených dat - průjezdů ramenem kruhového objezdu s hodnotami 0 (v pořádku) a 1 (skoro-nehoda). Typy skoro-nehod nerozlišujeme. Pro pravděpodobnosti P_i platí

$$P_i \propto \int_{\Theta^*} L_i(\Theta_i) f(\Theta_i|d(0)) d\Theta_i f(H_i|d(0)),$$

kde

$$L_i(\Theta_i) = \prod_{t=1}^N f_i(d_t|\psi_{i;t}, \Theta_i).$$

V pravděpodobnostech P_i je

- $L_i(\Theta_i) = L_i(\alpha_i)$ je likelihood z modelů (5.4) s parametry α_i - podíl skoronehod na průjezdech automobilů vybraným ramenem sledovaného kruhového objezdu,
- $f_i(d_t|\psi_{i;t}, \Theta_i) = f_i(d_t|\alpha_i)$ je statický model skoronehod s alternativním rozdělením

$$f_i(d_t|\alpha_i) = \alpha_i^{d_t} (1 - \alpha_i)^{1-d_t}, \quad \alpha_i \in O_i,$$

kde O_i je oblast přípustných parametrů podle hypotézy H_i .

- $f(\Theta_i|d(0)) = f(\alpha_i|d(0))$ jsou apriorní hp pro parametry α_i z oblasti O_i , $i = 1, 2, 3$; v našem příkladě budeme tyto hp uvažovat jako rovnoměrné, tedy nemáme žádné preference,
- $f(H_i|d(0))$ jsou apriorní pravděpodobnosti jednotlivých hypotéz; také je zvolíme rovnoměrně rozdělené.

Podle formulace hypotéz v úvodu příkladu budou jednotlivé oblasti definovány takto

1. $O_1 = \{\alpha_0\}$, tedy O_1 je jednobodová množina. Nad ní máme nyní konstruovat apriorní hp $f(\alpha)$, $\alpha \in O_1$. Jedná se o degenerovanou hp nenulovou jen v jediném bodě a s požadavkem, aby integrál přes ní byl roven jedné. To je Diracův impulz, umístěný v bodě α_0

$$f(\alpha|d(0)) = \delta(\alpha, \alpha_0).$$

Pro tento impulz platí

$$\int_{-\infty}^{\infty} g(\alpha) \delta(\alpha, \alpha_0) d\alpha = g(\alpha_0),$$

což je vlastnost, které dále využijeme.

2. $O_2 = (0, \alpha_0)$, který specifikuje α menší, než daná hranice α_0 a charakterizuje tak bezpečnou situaci. Apriorní hp bude

$$f(\alpha|d(0)) = \frac{1}{\alpha_0}, \quad \text{pro } \alpha \in (0, \alpha_0), \text{ jinak } 0.$$

3. $O_3 = (\alpha_0, 1)$, který specifikuje α větší, než daná hranice α_0 a charakterizuje tak nebezpečnou situaci. Apriorní hp bude

$$f(\alpha|d(0)) = \frac{1}{1 - \alpha_0}, \text{ pro } \alpha \in (\alpha_0, 1), \text{ jinak } 0.$$

Apriorní hp pro hypotézy bude

$$f(H_i|d(0)) = \frac{1}{3}, \quad i = 1, 2, 3.$$

Likelihood pro alternativní rozdělení

Likelihood je dán součinem modelů - v našem případě alternativních modelů

$$L_i(\alpha) = \prod_{t=1}^N \alpha_i^{d_t} (1 - \alpha_i)^{1-d_t} = \alpha_i^{\sum_{t=1}^N d_t} (1 - \alpha_i)^{\sum_{t=1}^N (1-d_t)} = \alpha_i^\nu (1 - \alpha_i)^{N-\nu},$$

kde ν je počet jedniček v datovém výběru, tj. počet pozorovaných skoronehod.

Integrál z likelihoodu

Pro vyčíslení pravděpodobnosti P_i potřebujeme vypočítat integrály z likelihoodů L_i podle jednotlivých hypotéz. Nejjednodušší je situace pro H_1 kdy parametr známe $\alpha = \alpha_0$. Bude

$$L_1(\alpha) = \int_0^1 \alpha^\nu (1 - \alpha)^{N-\nu} \delta(\alpha, \alpha_0) d\alpha = \alpha_0^\nu (1 - \alpha_0)^{N-\nu}.$$

Pro H_2 je

$$L_2(\alpha) = \int_0^{\alpha_0} \alpha^\nu (1 - \alpha)^{N-\nu} \frac{1}{\alpha_0} d\alpha = \frac{1}{\alpha_0} B_{inc}(\alpha_0, \nu, N - \nu).$$

Pro H_3 je situace obdobná jako pro H_2 až na obor integrace

$$\begin{aligned} L_3(\alpha) &= \int_{\alpha_0}^1 \alpha^\nu (1 - \alpha)^{N-\nu} \frac{1}{1 - \alpha_0} d\alpha = \\ &= \int_0^1 \alpha^\nu (1 - \alpha)^{N-\nu} \frac{1}{1 - \alpha_0} d\alpha - \int_0^{\alpha_0} \alpha^\nu (1 - \alpha)^{N-\nu} \frac{1}{1 - \alpha_0} d\alpha = \\ &= \frac{1}{1 - \alpha_0} [B(\nu, N - \nu) - B_{inc}(\alpha_0, \nu, N - \nu)], \end{aligned}$$

kde $B_{inc}(\alpha_0, \nu, N - \nu)$ je nekompletní beta funkce a $B(\nu, N - \nu)$ je beta funkce. Tyto funkce je možno nalézt v Matlabu po názvem `betainc` a `beta`. Pozor. Funkce `betainc`

je v podílovém tvaru, takže potřebný integrál získáme jako `betainc*beta` - použití viz program. Beta funkce je uvedena v příloze (9.1).

Pravděpodobnosti P_i tedy budou

$$\begin{aligned} P_1 &= \frac{1}{3} \alpha_0^\nu (1 - \alpha_0)^{N-\nu}, \\ P_2 &= \frac{1}{3} \frac{1}{\alpha_0} B_{inc}(\alpha_0, \nu, N - \nu), \\ P_3 &= \frac{1}{3} \frac{1}{1 - \alpha_0} [B(\nu, N - \nu) - B_{inc}(\alpha_0, \nu, N - \nu)]. \end{aligned}$$

Maximální hodnota P_i ukazuje na vítěznou hypotézu.

Poznámka

Hypotéza H_1 je proti ostatním příliš slabá - ozkoušejte. Proto v programu testujeme hypotézy takto

$$H_1: \alpha \in (-h, h)$$

$$H_2: \alpha \in (0, \alpha_0 - h)$$

$$H_2: \alpha \in (\alpha_0 + h, 1).$$

kde h je malé, předem zvolené číslo.

Program k příkladu

```
clc, clear all, close all
% Testing of safety on a roundabout (semiaccidents)

nd=100; % number of data
a0=.03; % border of safety
h=.01; % al=a0 means al=a0\pm h
% h must be smaller than a0

Phs=[];
switch 1 % set type of example
  case 1, nn=2:(nd-3); % set possible poz. results
  case 2, nn=42; % set single poz. result
end
for nu=nn % loop for al (al=nu/nd)
  B0=beta(nu-1,nd-nu-1); % beta function
  b2=B0*betainc(a0-h,nu-1,nd-nu-1); % for <
  b3=B0-B0*betainc(a0+h,nu-1,nd-nu-1); % for >
```

```

b1=B0-b2-b3; % for =

if abs(h)<1e-5
    P(1)=a0^nu*(1-a0)^(nd-nu)/3; % prob. for =
else
    P(1)=b1/(3*2*h); % prob. for (a1-h, a1+h)
end
P(2)=b2/(3*(a0-h)); % prob. for <
P(3)=b3/(3*(1-a0-h)); % prob. for >

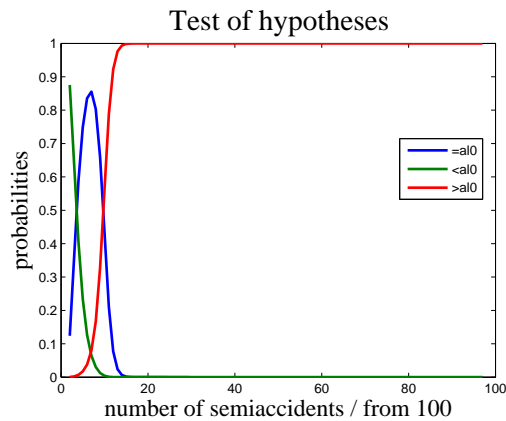
Ph=P/sum(P); % normalizing
Phs=[Phs; Ph]; % vector of probs

end

% RESULTS
aTestHyp_R

```

Výsledky



V tomto příkladě jsme uvažovali počet pozorování roven 100. Kritická hodnota podílu skoronehod byla $\alpha_0 = 0.03$. Pro každý možný datový výběr jsme postupně uvažovali 1, 2, 3 \dots 100 skoronehod a jednotlivé případy jsme zaznamenali podle osy x . Na obrázku tak vidíme vývoj pravděpodobností jednotlivých hypotéz pro podíl skoronehod od nuly až do jedné.

6 Logistická regrese

Hodiny: 11

Aplikace: klasifikace nehod.

Logistickou regresi použijeme, jestliže se pomocí skupiny veličin jak diskretních tak i spojitých snažíme vysvětlit chování diskretní veličiny. Jako příklad můžeme zmínit modelování typu nehody (lehká, vážná, s úmrtím) pomocí hodnot veličin, které tuto nehodu doprovázely (světlo, povrch vozovky, rychlost při střetu atd.). Tuto specifickou úlohu charakterizuje diskretní výstup a alespoň jedna spojitá nezávislá veličina. Pak nelze použít ani diskretní ani regresní model a je namístě model logistické regrese.

S úlohou logistické regrese jsme se již setkali v předmětu Stochastické systémy. Tam jsme ale pro jednoduchost počítali s tím, že výstup má jen dvě hodnoty. Tady tuto úlohu rozšíříme tak, aby zahrnovala i případy, kdy výstup má více různých hodnot.

Budeme uvažovat jednu modelovanou veličinu y_t s přípustnými hodnotami $0, 1, \dots, m$ s kategorickým rozdělením

$$\begin{array}{c|cccc} y & 0 & 1 & \cdots & m \\ \hline f(y|p) & p_0 & p_1 & \cdots & p_m \end{array}$$

a n nezávislých veličin x_1, x_2, \dots, x_n .

Model

Výstupem modelu jsou logaritmy podílů $\frac{p_i}{p_0}$ (funkce logit) které modelujeme pomocí regrese na nezávislé veličiny

$$\begin{aligned} \ln\left(\frac{p_1}{p_0}\right) &= b_{10} + b_{11}x_1 + \cdots + b_{1n}x_n + \epsilon_1, \\ \ln\left(\frac{p_2}{p_0}\right) &= b_{20} + b_{21}x_1 + \cdots + b_{2n}x_n + \epsilon_2, \\ &\dots \\ \ln\left(\frac{p_m}{p_0}\right) &= b_{m0} + b_{m1}x_1 + \cdots + b_{mn}x_n + \epsilon_m. \end{aligned}$$

Pro p_0 pak platí

$$p_0 = 1 - p_1 - p_2 - \cdots - p_m.$$

Označíme $b_{i0} + b_{i1}x_1 + \cdots + b_{in}x_n = xb_i$ a s nulovými šumy ϵ_i můžeme pro p_i psát

$$\begin{aligned} p_1 &= \frac{\exp\{xb_1\}}{1 + \sum_{i=1}^m \exp\{xb_i\}}, \\ p_2 &= \frac{\exp\{xb_2\}}{1 + \sum_{i=1}^m \exp\{xb_i\}}, \\ &\dots \end{aligned}$$

$$p_m = \frac{\exp\{xb_m\}}{1 + \sum_{i=1}^m \exp\{xb_i\}},$$

a

$$p_0 = 1 - \sum_{i=1}^m p_i = \frac{1}{1 + \sum_{i=1}^m \exp\{xb_i\}},$$

což je inverzní vztah k uvedenému logistickému modelu.

Likelihood a odhad

Pro výpočet likelihoodu uvažujeme data $\{y_t, x_t\}_{t=1}^N$.

Protože modelovaná veličina má kategoričké rozdělení, budeme počítat likelihood jako součin těchto rozdělení, kde za p_i dosadíme z výše uvedených vztahů

$$L = \prod_{t=1}^N \prod_{i=0}^m (p_i)^{\delta(i; y_t)} = \prod_{t=1}^N \frac{\prod_{i=1}^m \exp\{\delta(i; y_t) x_t b_i\}}{1 + \sum_{i=1}^m \exp\{x_t b_i\}}.$$

Tento výraz je třeba maximalizovat vzhledem k parametrům b .

Poznámka: Použitím log-likelihoodu se uvedený výraz ještě trochu zjednoduší. To ale pro nás není důležité, protože pro tuto optimalizaci již existují zavedené postupy - např. v Matlabu funkce `mnrfit` a `mnrval`, nebo naše funkce `logReg` - viz 11.7.

Klasifikace

Poté, co jsme z trénovací množiny $\{y_t, x_t\}_{t=1}^N$ odhadli parametry b logistického modelu, jsme připraveni tento model použít pro klasifikaci. To provedeme následujícím způsobem:

1. Změříme vzorek x a chceme jej přiřadit jedné ze tříd $1, 2, \dots, m$ (hodnot y).
2. Vzorek x dosadíme do inverzního modelu (pro veličiny p_i).
3. Vzorek zařadíme do třídy, které odpovídá maximální hodnota pravděpodobnosti p .

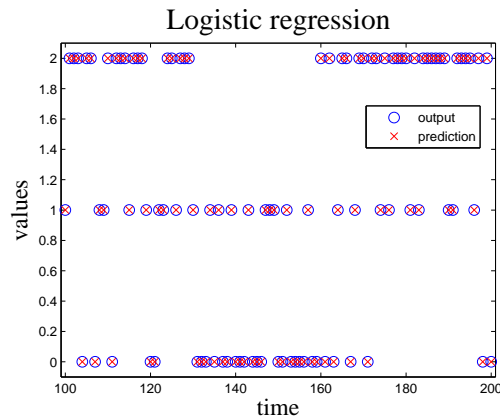
Vzorový příklad

Použití logistické regrese ukážeme na simulačním příkladě. Jako nezávislé veličiny vezmeme x_1 - posunutý normální šum, x_2 - posunutý rovnoměrná šum a x_3 sinusový signál (viz program). Hodnoty y přiřadíme takto: veličiny sečteme $z = x_1 + x_2 + x_3$ a potom pro $z \geq 0$ je $y = 2$, pro $z \in (-1, 0)$ je $y = 1$ a pro $z \leq -1$ je $y = 0$.

Odhad i predikce se provede pomocí funkce `logReg` (11.7), která

1. použitím matlabské funkce `fminsearch` maximalizuje likelihood,
2. pomocí vzorců ze str. 6 počítá pravděpodobnosti jednotlivých hodnot y ,
3. pro každý regresní vektor najde bodový odhad y , tj. tu hodnotu která má maximální pravděpodobnost.

Experiment pracuje s 200 vzorky. Výsledky, pro druhou polovinu experimentu, jsou na obrázku



Program

```

clc, clear all, close all
addpath 'funct'
%% Logistic regression

nd=200; % length of data

% independent variables
x1=1+randn(1,nd);
x2=-2+rand(1,nd);
x3=sin(15*(1:nd)/nd);

% generation of output
z=x1'+x2'+x3';
y=ones(nd,1);
for i=1:nd
    if z(i)>0 y(i)=2; end % assignment of
    if z(i)<-1 y(i)=0; end % y - values
end

```

```

% regression vector
x=[ones(nd,1) x1' x2' x3'];

% parameter estimation and output prediction
[py yp yr b]=logReg([],y,x);

% RESULTS
disp('Number of wrong classifications ')
sum(y~=yr)

vLogReg_R

```

6.1 Aplikace

Klasifikace nehod

Jako ilustrační příklad uvedeme aplikaci logistické regrese na klasifikaci nehod v závislosti na vedlejších veličinách, změřených nebo odhadnutých v okamžiku nehody. Charakteristickou vlastností takovéto úlohy je relativní nedostatek existujících dat a nemožnost aktivního získání dalších dat.

V reálném případě je třeba nejdříve určit hodnoty výstupu - tj. rozhodnout, jak budeme nehody charakterizovat. V našem případě, jak jsme již uvedli v úvodu kapitoly, budeme rozlišovat nehody 0 = lehké, 1 = těžké a 2 = s úmrtím.

Dále pak zjišťujeme, které veličiny doprovázející nehody jsou k dispozici, tj. jejich hodnoty jsou zaznamenávány a které mohou souviset s klasifikací těchto nehod. Při výběru těchto veličin jsme omezeni tím, co je k dispozici a výběr vhodných veličin je spíše uměním než exaktní vědou. To, co je k dispozici je názor expertů a metody určení struktury modelu, o kterých jsme mluvili v Kapitole 2.3. Expertní znalost je ale v těchto případech nedocenitelná.

V našem příkladě zvolíme jako vedlejší veličiny:

osvětlení:	1 dobré, 2 šero, 3 tma
povrch vozovky:	1 suchý, 2 mokrý, 3 kluzký
směr vozovky:	1 přímý, 2 zatáčka
rychlost vozidla:	spojitá veličina

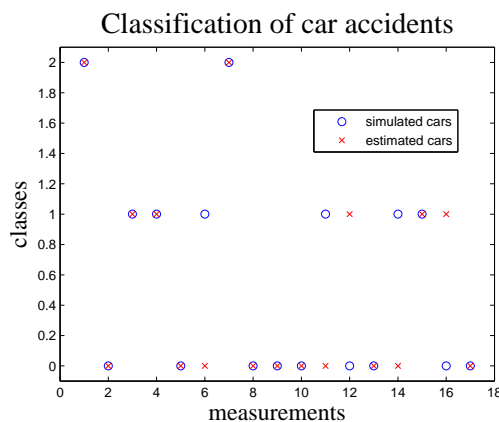
Model, který použijeme má tvar

$$\ln \left(\frac{p_{1;t}}{p_{0;t}} \right) = b_{01} + b_{11}x_{1;t} + b_{21}x_{2;t} + b_{31}x_{3;t} + b_{41}x_{4;t} + \epsilon_{1;t},$$

$$\ln \left(\frac{p_{2;t}}{p_{0;t}} \right) = b_{02} + b_{12}x_{1;t} + b_{22}x_{2;t} + b_{32}x_{3;t} + b_{42}x_{4;t} + \epsilon_{1;t},$$

kde $p_i = f(y_t = i|x_t)$, $i = 0, 1, 2$, y_t je typ nehody v čase t , x_j je j -tá vedlejší veličina, a b_{ji} je regresní koeficient u j -té veličiny modelující i -tý typ nehody.

Jako data použijeme vzorek, uvedený v programu. Tento vzorek odpovídá rozsahem menší dopravní oblasti a snaží se vystihnout obecné pravdy (jako že velká rychlost na kluzké vozovce je nebezpečná). Odhad modelu a zpětná predikce typu nehody jako kontrola jsou uvedeny na následujícím obrázku



Simulované typy nehod pro jednotlivé datové záznamy jsou označeny kroužky, odhadnuté typy nehod z příslušných regresních vektorů a odhadnutých parametrů jsou křížky. Vidíme, že většina simulovaných a odhadnutých dat se shoduje a tedy, že model má určitý smysl.

Poznámka

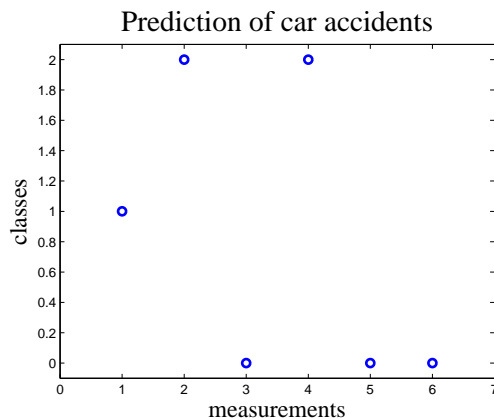
Existují testy kvality provedené logistické regrese, jako je např. Waldův test nebo test podílem likelihoodů. Tyto testy nebudeme zavrňovat. Pro nás je ale důležitý cíl, pro který regresi provádíme a tím je schopnost klasifikace. Proto zjišťujeme kvalitu modelu právě touto schopností klasifikace. Jistě by se slušelo provádět odhad na jedné množině dat a testovat na druhé, disjunktní. Jestliže ale máme takový extrémní nedostatek dat, děláme co můžeme a musíme se spokojit s testováním na stejné množině.

Odhadnutý model můžeme použít také k předpovědi očekávaného typu nehody pro libovolně zvolené situace (tj. osvětlení, povrchu, směru a rychlosti) i když takové situace vůbec neexistují a nebo jsou třeba teprve plánovány. Tímto experimentem lze také podpořit nebo naopak zpochybnit plánovanou dopravní stavbu.

V našem případě jsme zvolili následující data

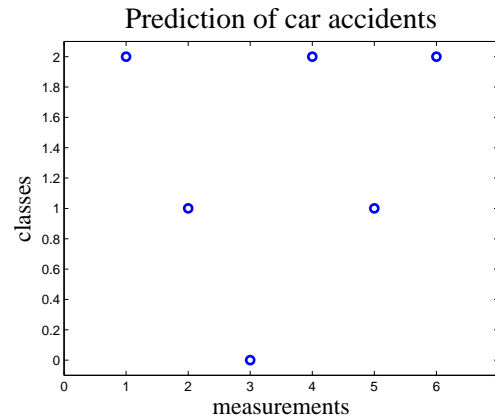
vzorek	osvětlení	povrch	směr	rychlost
1	1	1	1	50
2	1	1	1	150
3	2	2	2	50
4	2	2	2	150
5	3	3	3	50
6	3	3	3	150

Vzorky 1 a 2 uvádějí situaci, kdy je všechno dobré a malou a velkou rychlost. Podobně vzorky 3 a 4 v situaci střední a 5 a 6 pro situaci špatnou. Výsledné přiřazení očekávaného typu nehody je na následujícím obrázku



Tento obrázek necháme prohlédnout expertovi, ten se zamyslí a řekne: to je nesmysl - v nejhorších podmínkách a při rychlosti 150 km/h to nemůže skončit jinak než smrtí (poslední vzorek - indikuje se jen těžká havárie).

Experta vezmeme vážně a chceme podle jeho prohlášení do-zabudovat do modelu apriorní (tady spíše dodatečnou) expertní informaci že “při nejhorších podmínkách a rychlosti 150 km/h havárie končí smrtí”. Tomu odpovídají data $y = 2$, $x_1 = 3$, $x_2 = 3$, $x_3 = 3$, $x_4 = 150$. Provedeme tedy znovu odhad modelu a k měřeným datům přidáme ještě 4 krát expertní data. Čtyřnásobné opakování tuto informaci posiluje - jako by tato situace byla pozorována 4 krát. Po novém odhadu a provedení testu dostaneme



Vidíme, že po dodatečné informaci model již expertní názor respektuje.

Program

```

clc, clear all, close all
addpath 'funct'
% Classification of traffic accidents
% y type of accident: 0 (mild) 1 (severe) 2 (death)
%
% x1 light: 1 (bright) 2 (gloom) 3 (dark)
% x2 road surface: 1 (dry) 2 (wet) 3 (slippery)
% x3 road type: 1 (stright) 2 (curve)
% x4 speed of car: continuous

% measured data
dt=[...
    % type of accident
    % light
    % road surfice
    % road type
    % speed of car
    2 1 1 1 145
    0 2 1 2 68
    1 3 2 3 94
    1 2 2 3 66
    0 1 2 1 15
    1 3 1 2 95
    2 2 2 2 104
    0 1 2 1 53
    0 2 2 1 68

```

```

0   3  1  1  39
1   3  3  2  8
0   3  1  3  41
0   2  1  1  62
1   2  2  2  98
1   2  3  3  115
0   2  3  2  69
0   3  3  1  103
];

% expert data
if 1 % expert data 0=no, 1=yes
dp=[...
2   3  3  3  150
2   3  3  3  150
2   3  3  3  150
2   3  3  3  150
];
dt=[dt; dp];
end

% tested regression vectors
xt=[...
1  1  1  50
1  1  1  150
2  2  2  50
2  2  2  150
3  3  3  50
3  3  3  150
];

y=dt(:,1);
x=dt(:,2:5);
nd=length(y);

% model estimation
[py yp yr b]=logReg([],y,x);

% testing of regression vectors
[pyt ypt yrt]=logReg(b,y,xt);

% Results
disp ''
disp('Estimated parameters')
```

b

aNehody_R

7 Odhad statické směsi

Hodiny: 12

Aplikace: určení módu jedoucího auta.

7.1 Odhad modelu ve tvaru směsi komponent

Model směsi komponent je tvořen množinou n_c komponent a diskrétní veličinou c_t , která v každém okamžiku t ukazuje na aktivní komponentu, tzv. ukazovátko.

Komponenty

Komponenty jsou obecně dynamické modely

$$f(d_t|c, \Theta_c, \psi_t), \quad c \in \{1, 2, \dots, n_c\} = c^*,$$

kde d_t jsou modelovaná data (řízení neuvažujeme), c označuje číslo komponenty, Θ_c jsou parametry c -té komponenty a ψ_t je regresní vektor (který bereme společný všem komponentám).

Ukazovátko

Ukazovátko modelujeme pomocí kategorického modelu

$$f(c_t|\alpha, d(t-1)) \underbrace{=}_{\text{nezávislost}} f(c_t|\alpha) = \alpha_{c_t}.$$

Odhad modelu směsi komponent není přímočarý. Po přímém použití Bayesova vzorce dostaneme vztah pro přepočítání statistik, ten ale není použitelný, protože nároky na jeho spočítání neustále rostou s množstvím zpracovaných dat. V odvození algoritmu pro odhad parametrů modelu směsi proto provedeme následující aproximaci. Vyjdeme z předpokladu, že aktivní komponenta je známá a v čase t má hodnotu c_t . Potom ukazovátko, jako obecně náhodná veličina má hp

$$\delta(c; c_t), \quad c \in c^*.$$

Ve skutečnosti ale aktivní komponentu neznáme, ukazovátka c_t je skutečně náhodná veličina a hp ukazovátka odhadujeme jako podmíněnou střední hodnotu na základě toho, co skutečně známe, tj. data $d(t)$

$$E[\delta(c; c_t) | d(t)] = \sum_{c=1}^{n_c} \delta(c; c_t) f(c_t | d(t)) = Pr(c_t = c | d(t)) = w_{c;t},$$

kde výslednou pravděpodobnost označíme $w_{c;t}$ a celý vektor, který odpovídá hp $f(c_t | d(t))$ je w_t a budeme o něm hovořit jako o vektoru vah jednotlivých komponent. Jedná se o aktuální váhy, ve kterých je zahrnuta informace i z nově změřených dat d_t . Podle maximální složky tohoto vektoru můžeme tedy provádět odhad aktuální komponenty nebo, jinak řečeno, provádíme **úlohu klasifikace** dat do tříd, které představují jednotlivé komponenty.

Zatím jsme ale hp $f(c_t | d(t))$ nezkonstruovali. To provedeme v následujících odstavcích. Uvidíme, že úloha konstrukce zmíněné hp je úzce svázána s úlohou odhadu parametrů komponent i ukazovátka.

Vyjdeme ze sdružené hp neznámých veličin c_t , Θ a α podmíněné dosud změřenými daty

$$\mathcal{J} = f(c_t, \Theta, \alpha | d(t)).$$

Použitím Bayesova vzorce, rozvojem sdružené hp a sdružením členů, které k sobě podle parametrů patří dostaneme

$$\mathcal{J} = f(d_t | c_t, \psi_t, \Theta_c) f(\Theta | d(t-1)) \times \alpha_{c_t} f(\alpha | d(t-1)), \quad (7.1)$$

kde jednotlivé hp zleva jsou: model c_t -té komponenty, apriorní hp pro odhad Θ , model ukazovátka $f(c_t | \alpha) = \alpha_{c_t}$ a apriorní hp pro odhad α . V tomto rozvoji jsme také použili předpoklady o nezávislosti: model komponenty nezávisí na α a datech starších než v regresním vektoru, nezávislost parametrů Θ a α , model ukazovátka nezávisí na datech.

Jestliže jsou k dispozici příslušné apriorní hp, jsme schopni z \mathcal{J} zkonstruovat vektor vah $w_{c;t}$

$$\begin{aligned} w_{c;t} &= f(c_t | d(t)) = \int_{\Theta^*} \int_{\alpha^*} f(c_t, \Theta, \alpha | d(t)) d\alpha d\Theta = \\ &= \int_{\Theta^*} f(d_t | c_t, \psi_t, \Theta_c) f(\Theta | d(t-1)) d\Theta \int_{\alpha^*} \alpha_{c_t} f(\alpha | d(t-1)) d\alpha. \end{aligned} \quad (7.2)$$

Dále si všimneme struktury sdružené hp \mathcal{J} (7.1) jejího integrálu - vektoru vah w_t (7.2). Ty jsou tvořeny vždy dvěma samostatnými částmi vztahujícími se buď k parametru Θ nebo α .

Nejprve pro sdruženou hp (7.1). První část je

$$\mathcal{P}_{c_t}^{\Theta} = f(d_t | c_t, \psi_t, \Theta_c) f(\Theta | d(t-1)),$$

což je součin modelu c_t -té komponenty s apriorní hp pro Θ . To je podle Bayesova vzorce nenormovaná aposteriorní hp pro odhad Θ . Její převrácená normalizační konstanta je dána jako integrál z této hp

$$\mathcal{I}_{c_t}^{\Theta} = \int_{\Theta^*} f(d_t|c_t, \psi_t, \Theta_c) f(\Theta|d(t-1)) d\Theta,$$

což je první část výrazu (7.2).

Podobně je to i s druhými částmi výrazů (7.1) a (7.2). Druhou částí sdružené hp (7.1) je aposteriorní hp pro odhad parametru α

$$\mathcal{P}_{c_t}^{\alpha} = \alpha_{c_t} f(\alpha|d(t-1))$$

s normalizační konstantou danou druhou částí výrazu (7.2)

$$\mathcal{I}_{c_t}^{\alpha} = \int_{\alpha^*} \alpha_{c_t} f(\alpha|d(t-1)) d\alpha.$$

Výrazy \mathcal{P}^{Θ} a \mathcal{P}^{α} souvisí s odhadem parametrů Θ a α , součin výrazů \mathcal{I}^{Θ} a \mathcal{I}^{α} určuje hp $f(c_t|d(t)) = w_{c_t;t}$. Dále se budeme zabývat konstrukcí těchto výrazů.

Odhad parametrů Θ

Odhad parametru Θ s pomocí komponenty c_t (tj. té správné aktivní) určíme podle Bayesova vzorce takto

$$\mathcal{P}_{c_t}^{\Theta} = f(\Theta|d(t)) \propto f(d_t|c_t, \psi_t, \Theta_c) f(\Theta|d(t-1)),$$

kde model má normální rozdělení a v souladu s požadavkem spočitatelnosti (reprodukovatelnosti apriorní hp) volíme hp parametrů ve tvaru součinu inverzních Gauss-Wishartových distribucí (viz ..). Model píšeme výhodně v součinném tvaru. Dostaneme

$$\prod_{c=1}^{n_c} GiV(V_{c;t}, \kappa_{c;t}) \propto \prod_{c=1}^{n_c} f(d_t|c, \psi_t, \Theta_c)^{\delta(c;c_t)} \prod_{c=1}^{n_c} GiV(V_{c;t-1}, \kappa_{c;t-1}).$$

Kroneckerovu funkci δ v exponentu modelu nahradíme aproximací w a dostaneme přepočet statistik

$$V_{c;t} = V_{c;t-1} + w_{c;t} \Psi_t \Psi_t', \quad \kappa_{c;t} = \kappa_{c;t-1} + w_{c;t}, \quad (7.3)$$

pro $c = 1, 2, \dots, n_c$.

Z přepočtených statistik můžeme buď určit bodové odhady parametrů nebo zkonstruovat celou aposteriorní hp (viz ...)

Odhad parametru α

Podobně jako pro parametr Θ počítáme podle Bayesova vzorce

$$\mathcal{P}_{c_t}^\alpha = f(\alpha|d(t)) \propto \alpha_{c_t} f(\alpha|d(t-1)),$$

kde opět pro hp parametrů zvolíme konjugovanou hp ve tvaru Dirichletova rozdělení a model vyjádříme v součinném tvaru

$$\prod_{c=1}^{n_c} \alpha_c^{\nu_{c;t}} \propto \prod_{c=1}^{n_c} \alpha_c^{\delta(c;c_t)} \prod_{c=1}^{n_c} \alpha_c^{\nu_{c;t-1}}.$$

Po aproximaci $w \rightarrow \delta$ dostaneme přepočet statistik

$$\nu_{c;t} = \nu_{c;t-1} + w_{c;t}, \quad (7.4)$$

pro $c = 1, 2, \dots, n_c$.

Výsledkem může být opět bodový odhad nebo celá aposteriorní hp.

Konstrukce vah w_t

Pro přepočet statistik v odhadu parametrů jsme potřebovali aktuální váhy komponent $w_t = [w_{1;t}, w_{2;t}, \dots, w_{n_c;t}]'$ které určují pravděpodobnosti toho, že daná komponenta je aktivní (tj. nejlépe popisuje změřený datový vzorek d_t). Tento vektor je určen rovnicí (7.2) a platí pro něj

$$w_{c;t} = \mathcal{I}_c^\Theta \mathcal{I}_c^\alpha.$$

Integrál \mathcal{I}_c^Θ :

$$\mathcal{I}_c^\Theta = \int_{\Theta^*} f(d_t|c, \psi_t, \Theta_c) f(\Theta|d(t-1)) d\Theta,$$

což je likelihood komponenty c . Tento likelihood je možno spočítat, jeho tvar je ale dosti složitý a neprůhledný. Veliké zjednodušení dostaneme, použijeme-li bodový odhad $\hat{\Theta}_{t-1}$ parametru Θ . Potom

$$\mathcal{I}_c^\Theta = f(d_t|c, \psi_t, \hat{\Theta}_{c;t-1}) \equiv \hat{m}_{c;t},$$

tedy přímo model c -té komponenty s dosazeným bodovým odhadem za parametr Θ . Pro přehlednost jej označíme $\hat{m}_{c;t}$.

Integrál \mathcal{I}_c^α :

$$\mathcal{I}_c^\alpha = \int_{\alpha^*} \alpha_c f(\alpha|d(t-1)) d\alpha = \frac{\nu_{c;t-1}}{\sum_{i=1}^{n_c} \nu_{i;t-1}} = \hat{\alpha}_{c;t-1},$$

což je bodový odhad parametru α_c .

Vektor aktuálních vah w_t tedy bude mít prvky

$$w_{c;t} = \mathcal{I}_c^\Theta \mathcal{I}_c^\alpha = \hat{m}_{c;t} \hat{\alpha}_{c;t-1}, \quad c \in \mathcal{C}^*, \quad (7.5)$$

kde $\hat{m}_{c;t} = f(d_t | c, \psi_t, \hat{\Theta}_{c;t-1})$.

Sumarizujeme a dostáváme **algoritmus odhadu** pro jeden krok v časové smyčce.

Máme k dispozici:

- statistiky odhadu z minulého kroku $V_{c;t-1}$, $\kappa_{c;t-1}$, $\nu_{c;t-1}$ pro $c \in \mathcal{C}^*$;
- odhady parametrů spočtené ze statistik $\hat{\Theta}_{t-1}$ a $\hat{\alpha}_{t-1}$ a
- odhady komponent $\hat{m}_{c;t} = f(d_t | c, \psi_t, \hat{\Theta}_{c;t-1})$.

Provedeme

1. Změříme nová data d_t .
2. Spočteme “vzdálenosti” komponent od d_t : $\hat{m}_{c;t} = f(d_t | c, \psi_t, \hat{\Theta}_{c;t-1})$.
3. Určíme integrály $\mathcal{I}_c^\Theta = \hat{m}_{c;t}$ a $\mathcal{I}_c^\alpha = \hat{\alpha}_{c;t-1}$.
4. Spočteme váhy $w_{c;t} = \mathcal{I}_c^\Theta \mathcal{I}_c^\alpha = \hat{m}_{c;t} \hat{\alpha}_{c;t-1}$.
5. Provedeme přepoččet statistik obou modelů

$$V_{c;t} = V_{c;t-1} + w_{c;t} \Psi_t \Psi_t', \quad \kappa_{c;t} = \kappa_{c;t-1} + w_{c;t},$$

$$\nu_{c;t} = \nu_{c;t-1} + w_{c;t}.$$

6. Vypočteme bodové odhady parametrů komponent a ukazovátka

$$\hat{\theta}_{c;t} = V_{\psi;t}^{-1} V_{y\psi}, \quad \hat{r}_{c;t} = \frac{V_y - V_{y\psi}' V_{\psi}^{-1} V_{y\psi}}{\kappa_{c;t}}, \quad \hat{\alpha}_{c;t} = \frac{\nu_{c;t}}{\sum_{i=1}^{n_c} \nu_{c;t}},$$

kde $V_{c;t} = \begin{bmatrix} V_y & V_{y\psi} \\ V_{y\psi}' & V_{\psi} \end{bmatrix}$ je rozdělená informační matice komponenty c .

7. Zvýšíme časový index $t = t + 1$ a jdeme na bod 1.

Výsledkem je

- klasifikace datového vzorku d_t do komponenty s maximální vahou $w_{c;t}$,
- použití datového vzorku pro učení, tj. odhad parametrů komponent a ukazovátka.

Vzorový příklad

Odvozený algoritmus budeme demonstrovat na příkladě se simulovanými daty. Simulaci provedeme následujícím programem

```

clc , clear all , close all
% Simulation of data from static mixture with 5 components

nd=50000;                % number of data
nc=5;                    % number of components

% centers of components
cen{1}=[5; 3];          cen{2}=[-5; -4];    cen{3}=[10; -5];
cen{4}=[-4; 3];         cen{5}=[5; -3];

% covariance of components
cov{1}=[1 0;0 1];      cov{2}=[2 0;0 1];    cov{3}=[1 0;0 3];
cov{4}=[1 2;2 1];     cov{5}=[1 2;1 3];

als=[2 3 2 5 3];
als=fnorm(als);        % pointer parameter

% simulation
for t=1:nd
    % generation of active component
    cs(t)=sum(rand>cumsum(als))+1;
    % generation of data
    y(:,t)=cen{cs(t)}+cov{cs(t)}*randn(2,1);
end

save ySim y cs cen cov als    % save of the data and
                             % simulated pointer, centers and coves
                             % of components, pointer parameter

scatt(y)

```

Výsledná data jsme uložili na disk a při klasifikaci je zase z disku natáhneme.

Vlastní program klasifikace je následující

```

clc , clear all , close all
% Estimation of mixture with static pointer
% - simulated data with two dimensional output

nd=20000;                % number of data for est.

```

```

nc=5; % number of components

load ySim % data load
yIni=y(:,nd+(1:200)); % initial output
y=y(:,1:nd); % output

% INITIALIZATION
typ=0; % static mixture
[th co al V ka nu]=iniMix(nc,yIni,typ);
w=zeros(nc,nd); Et1=w; Et2=w;
for c=1:nc,
    Et1(c,1)=th{c}(1); % remembering initial
    Et2(c,1)=th{c}(2); % parameters
end
w(:,1)=ones(nc,1); % vector weights

% ESTIMATION
fprintf('running .....|\n. '),itime=0;
for t=2:nd % TIME LOOP -----
    itime=itime+1; % indicator of time-loop
    if itime>(nd-1)/20, fprintf(' '), itime=0; end % progress

    yt=y(:,t); % output measuring

    % distances of yt from components
    for c=1:nc
        [xxx p(c,1)]=Gauss2(yt,th{c},co{c}); % value of component
    end
    p=p-min(p); p=exp(p);
    if p'*p<1e-18, p=ones(size(p)); end
    % weights
    wp=p.*al; % new vector weights
    w(:,t)=wp/sum(wp); % normalization of w

    % update of statistics
    Ps=[yt;1]; % regression vector
    for c=1:nc
        V{c}=V{c}+w(c,t)*Ps*Ps'; % info. matrix
        ka(c)=ka(c)+w(c,t); % counter
        [th{c} co{c}]=v2thN(V{c}/ka(c),2); % point estimates
    end
    nu=nu+w(:,t); % Dirichlet stats
    al=fnorm(nu); % point estimate

```

```

    for c=1:nc, % remembering of
        Et1(c,t)=th{c}(1); % evolution
        Et2(c,t)=th{c}(2); % of parameter
    end % estimates
end

[xxx ct]=max(w); % active components

```

```

% RESULTS
vMixStPr_R

```

Tady zadáme počet kroků časové smyčky a počet odhadovaných komponent. Dále natáhneme data, část zvolíme jako apriorní data pro inicializaci a dále označíme data pro časovou smyčku.

V části inicializace se volá funkce `iniMix`, která vygeneruje počáteční parametry (především rozmístění) komponent a ukazovátka. Proměnné `Etc1` a `Etc2` slouží k zapamatování vývoje bodových odhadů parametrů komponent.

V části estimace se nejdříve zpracovává indikace postupu časové smyčky. Dále se z natanžených dat y vzorkuje aktuální datový vektor yt . Pro tento vektor se počítají hodnoty všech hp komponent s aktuálně odhadnutými parametry. Protože tyto hodnoty mohou být velmi malé, počítají se logaritmy. Od těch se odečte maximální hodnota (tak, že maximální prvek je nula) a pak se teprve aplikuje funkce exponenciála. Pokud ani tato procedura nepomůže, volí se hodnoty rovnoměrně.

Váhy w_t se spočtou tak, že se hodnoty komponent ve změřených datech násobí stacionárními váhami al (hodnoty komponent říkají, jak “blízko” jsou aktuální data jednotlivým komponentám - udávají aktuální pravděpodobnost, že aktuální data patří jednotlivým komponentám; hodnoty vektoru al představují četnosti historických odhadů aktuálních komponent - tedy pravděpodobnosti, že daná komponenta je aktivní aniž bychom znali aktuální data). Váhy w_t tedy berou v úvahu obě pravděpodobnosti - jak aktuální, tak i stacionární. Výsledná pravděpodobnost příslušnosti aktuálních data je tedy kombinací informace z aktuálně měřených dat i toho, jak často byla která komponenta v minulosti aktivní.

Dále se provádí přepočítání statistik komponent i ukazovátka podle příslušných vzorců (7.3) a (7.4) a počítají se bodové odhady (2.13), (2.14) a (2.7).

Poslední část časové smyčky je již jen zapamatování aktuálních bodových odhadů pro jejich pozdější vykreslení.

Na konci programu se tisknou a zobrazují výsledky.

Použité funkce jsou:

– inicializace algoritmu odhadu

```
function [th coal V ka nu]=iniMixD(nc, yIni, typ, ph, phc)
```

```

% [th co al V ka nu]=iniMixD(nc,yIni,typ)    heuristic mixture
%                                           initialization
% th      component regression coefficients
% co      comp. covariances
% al      pointer model parameter
% V,ka,nu corresponding statistics
% nc      number of components
% yIni    initial data
% typ     type of mixture  0=static , 1=dynamic

if nargin<4
    ph=100;                                % reducing of V,ka to decrease
end                                        % influence of initial data
                                           % to parameter evolution

if nargin<5
    phc=10;                                % reducing of component coves
end

% COMPONENTS
[ny nd]=size(yIni);
ka=nd*ones(1,nc);                         % initial comp. counters
cen=mean(yIni')';                          % overall center
dev=std(yIni')';                           % overall deviation
for c=1:nc                                  % for all components do
    a=cen+dev.*randn(ny,1);                 % ini. comp. centers
    yy=a*ones(1,nd)+...
        .5*diag(dev)*randn(ny,nd); % ini. data
    Ps=[yy; ones(1,nd)]';                  % regression vector
    V{c}=Ps'*Ps;                           % initial information matrix
    V{c}=V{c}/ph; ka(c)=ka(c)/ph; % correction of comp. stats
    Vr=V{c}/(10*ka(c));                    % norm. and red. inf. matrix
    [th{c} co{c}]=v2thN(Vr,ny);           % ini. point estimates
end

% POINTER
if typ==1 % dynamic mixture
    nu=5+rand(nc);                         % initial pointer statistics
    al=fnorm(nu,2);                        % point est. of par. alpha
else % static mixture
    nu=5+rand(nc,1);                       % initial pointer statistics
    al=fnorm(nu);                           % point est. of par. alpha
end

```

– hustota pravděpodobnosti normálního rozdělení je v knihovně funkcí 11.5.

Jako výsledky dostaneme jednak tisk konečných center komponent

Centres of components:

-4.4047 -3.4377

5.5821 2.6665

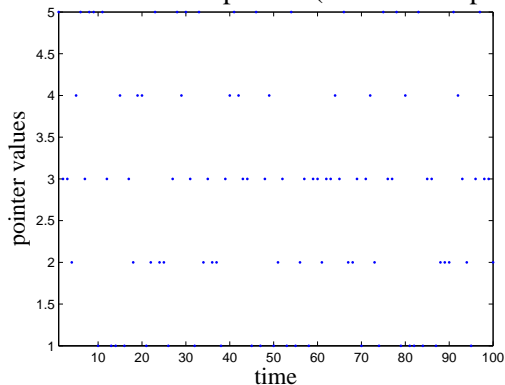
-4.0363 3.0498

10.0102 -5.6054

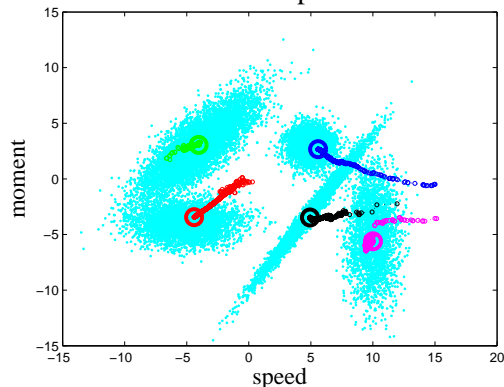
4.9299 -3.4665

jednak grafy: vlevo je posledních 100 hodnot ukazovátka (vidíme, že všechny komponenty jsou průběžně obsazovány); vpravo jsou datové klastry a na nich je zobrazen vývoj odhadů center komponent (všechny komponenty jsou správně detekovány a určeny).

Evolution of the pointer (last 100 samples)



Evolution of component centers



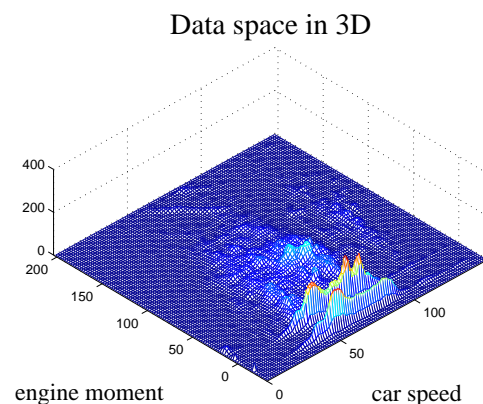
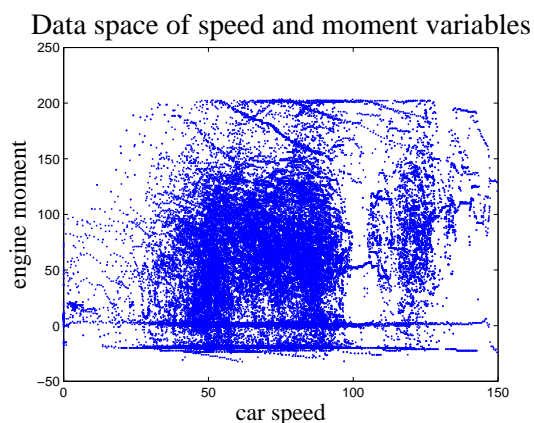
7.2 Aplikace

Klasifikace cesty za jízdy automobilu

V tomto příkladě ukážeme využití modelu směsi ke klasifikaci prostředí, kterým projíždí automobil. Jedná se např. o městské komunikace, okresní silnice, silnice 1. třídy, rychlostní komunikace nebo dálnice. Takové “pracovní módy” budeme hledat v dvojici měřených veličin, a to rychlosti automobilu a momentu motoru. Samozřejmě, v reálné situaci bychom využili větší množství měřených veličin. Tento jednoduchý příklad využijeme pro dvojí účel: (i) ukážeme využití modelu směsi pro klasifikaci, (ii) ukážeme modifikaci řešení, která lépe vyhovuje zadané úloze.

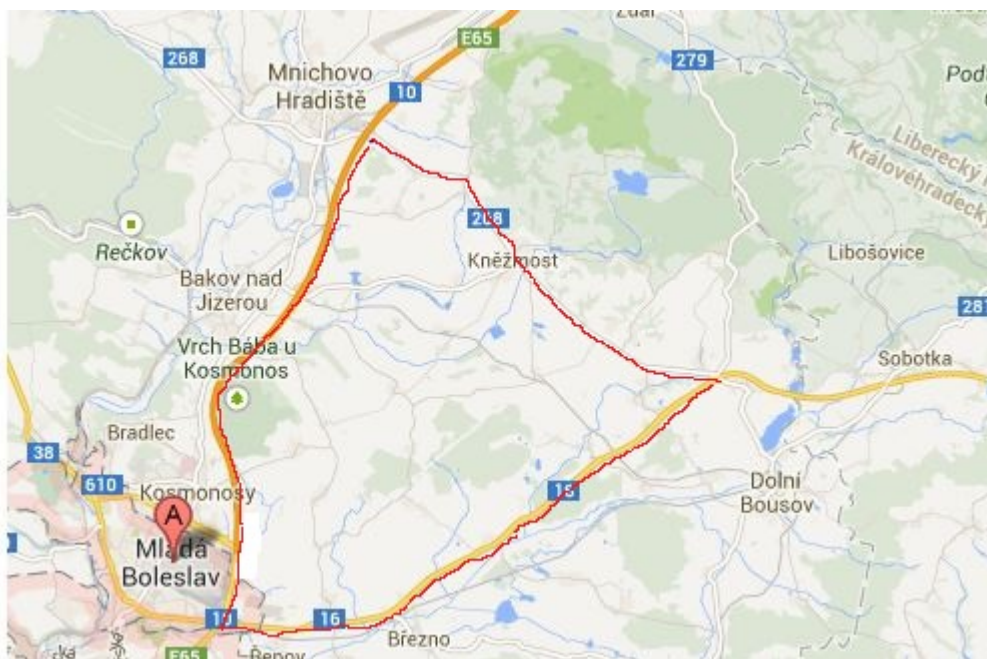
V příkladu navážeme na úvodní aplikaci o levném řízení automobilu. Aktuální režim levné jízdy je jistě ovlivněn také typem vozovky, po které automobil jede. Proto jako součást úlohy levného řízení můžeme považovat také úlohu klasifikace vozovky.

Data, která jsme si vypůjčili z úlohy o levném řízení tvoří datový prostor zobrazený na následujících obrázcích



Je zobrazen datový vzorek o rozsahu 75 000 dat změřený na osmi průjezdech testovacím okruhem z úvodní úlohy. Vlevo je pohled shora, vpravo je tentýž datový prostor, ale v 3D zobrazení. Na levém a podobně i pravém obrázku jsou patrné základní shluky odpovídající prostředí, kterým automobil projíždí. Prakticky uprostřed nad rychlostmi 50, 70 a 90 jsou tři velké shluky, odpovídající jízdě ve městě, horších a lepším silnicích mimo město. V pravé části je jízda po dálnici. V horní části obrázku je oblast předjíždění nebo prostě prudkého zrychlování. Dole jsou dva zajímavé a velmi výrazné klastry. Nahoře je jízda na volnoběh a pod ním brzdění motorem. Oba tyto klastry se objevují prakticky ve všech rychlostech. Často se objevuje ještě výrazný ostrý klastr v počátku datového prostoru, který odpovídá stání s běžícím motorem. Tento klastr zde chybí a my chválíme řidiče, že nenechává běžet motor, když nejede.

Cílem experimentu bude odhadovat prostředí, kterým automobil projíždí. Výsledky lze porovnat s realitou - testovacím okruhem



Start okruhu je vlevo dole. Pokračuje nahoru po dálnici, pak odbočuje doprava po silnici druhé třídy a uzavírá se po silnici první třídy. Délka celého okruhu je přibližně 40 km.

Bezprostřední řešení

Datový vzorek modelujeme směsí statických komponent se statickým ukazovátkem - tím říkáme, že v datovém prostoru jsou pravděpodobnostní kopce (shluky dat [rychlost; moment] (viz obrázky datového prostoru) a že (i) tyto kopce v prostoru “sedí” a v čase se nepohybují (statické komponenty) a dále, (ii) ukazovátka nemá tendenci k pravidelným přechodům z jednoho shluku do druhého (statické ukazovátka).

Model budeme odhadovat, tj. budeme určovat střední hodnoty (pozice) a kovarianční matice (šířky) jednotlivých komponent (datových shluků). V každém kroku odhadu modelu se odhaduje také aktivní komponenta. Pokud je model dobře odhadnut, každý mód systému (datový shluk) pokrývá jedna komponenta. Odhad aktivní komponenty je tedy ekvivalentní s určením okamžitého módu systému, který je dán typem prostředí, kterým automobil projíždí (město, mimo město, dálnice ...).

V prvním pokusu, kterým chceme především demonstrovat samotný odhadovací algoritmus provedeme inicializaci odhadu dosti obecně. Vyjdeme z apriorních dat, tj. dat, změřených ještě před započítáním odhadování. Spočteme jejich střední hodnotu a směrodatnou odchylku bez ohledu na jednotlivé pracovní módy systému. Určíme počet komponent (např. z xy -grafu apriorních dat, nebo z expertní znalosti) a počáteční středy těchto komponent “rozhodíme” kolem spočteného středu dat ve vzdálenosti korespondující se spočtenou směrodatnou odchylkou dat.

Odhadovací algoritmus ale potřebuje počáteční statistiky, tj. statistiky, ze kterých se vypočtou námi stanovené bodové odhady parametrů. To provedeme nejlépe tak, že s danými bodovými odhady komponent generujeme data, která potom zkomprimujeme do statistik. Pro statické komponenty ve tvaru

$$y_t = \theta_c + e_t, \quad e_t \sim N(0, r_c)$$

budeme počítat takto:

Pro $t = 1, 2, \dots, n_I$

$$y_t = \theta_c + \sqrt{r_c} \cdot \text{randn}(2, 1), \quad (7.6)$$

$$\Psi = [y_t; 1],$$

$$V = V + \Psi\Psi', \quad \kappa = \kappa + 1. \quad (7.7)$$

kde $\sqrt{r_c}$ je horní faktor kovarianční matice r_c pro který platí $\sqrt{r_c} \cdot \sqrt{r_c}' = r_c$ (tento faktor počítá funkce `uut` a pro diagonální matici r_c je to opět diagonální matice s odmocninami prvků na diagonále); Ψ je rozšířený regresní vektor. V a κ jsou statistiky komponent. Statistiku ukazovátka volíme např. jako vektor desítek (to odpovídá tomu, že nemáme preference o vahách komponent a tato informace pochází z $10 \cdot n_c$ apriorních dat (tedy není příliš silná, vezmeme-li v úvahu, že pro odhad používáme desítky tisíc dat).

Odhad realizuje následující program (uvádíme jen časovou smyčku, celý program lze nalézt mezi programy).

```

for t=2:nd
    yt=y(:,t); % output measuring % 1

    % distances of yt from components
    for c=1:nc % 2
        [xxx p(c,1)]=...
            GaussN(yt,th{c},co{c}); % value of component % 3
    end
    p=p-max+(p); p=exp(p); % 4
    if p'*p<1e-18, p=ones(size(p)); end % 5
    wp=p.*a1; % new vector weights % 6

    % normalizarion of weights
    w(:,t)=wp/sum(wp); % normalization of w % 7

    % update of statistics
    Ps=[yt;1]; % regression vector % 8
    for c=1:nc
        V{c}=V{c}+w(c,t)*Ps*Ps'; % info. matrix % 9
        ka(c)=ka(c)+w(c,t); % counter %10
        [th{c} co{c}]=v2thN(V{c}/ka(c),2); % point estimates %11
    end
end

```

```

end
nu=nu+w(: , t);           % Dirichlet stats      %12
al=fnorm(nu);             % point estimate      %13
end

```

Na začátku časového cyklu (příkaz %1) změříme nová data yt .

Dále (cyklus %2) spočteme “vzdálenosti” změřených dat od jednotlivých komponent. Data dosadíme do modelu komponenty, ve kterém figurují současné bodové odhady parametrů. Čím větší bude vypočtená hodnota hp modelu, tím spíše k tomuto modelu data patří. Protože hp normálního modelu rychle klesají k nule, budou hodnoty vzdálenějších modelů (tj. těch, co mají centrum dále od změřených dat) velmi blízké nule. Proto se nepočítá přímo hodnota hp modelu ale její logaritmus.

V příkazu %4 se hodnoty normalizují tak, aby největší z nich byla nula a pak se z nich počítá exponenciála.

Pokud ani tato normalizace nepomůže, volí se vzdálenosti rovnoměrně (příkaz %5).

Příkaz %6 počítá váhy wt (okamžité váhy komponent vzhledem k naměřenému datovému vektoru yt a příkaz %7 je normalizuje na pravděpodobnosti.

Příkazy %8 - %10 provádí přepočty statistik komponent, příkaz %11 počítá bodové odhady parametrů komponent.

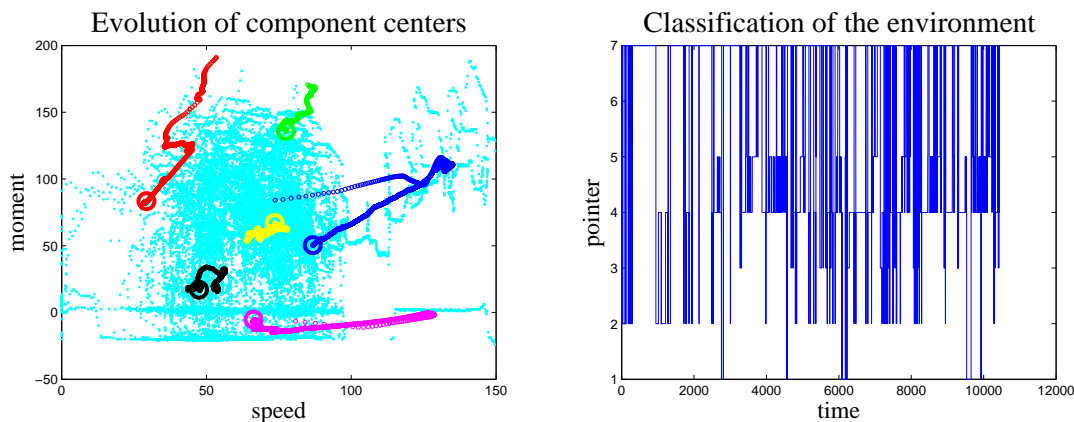
Příkaz %12 je přepočty statistik pro ukazovátka, příkaz %13 výpočet bodového odhadu parametru ukazovátka.

Pro experiment byla použita data ze tří jízd na testovacím okruhu, tedy asi ze 120 km jízdy. Zobrazují se data z jízdy na posledním okruhu. Připomeňme, že první třetina okruhu je jízda po dálnici, druhá třetina po vedlejších silnicích a poslední je jízda po silnici 1. třídy, která ovšem projíždí několika obcemi.

Výsledek odhadu je následující:

влево - na datovém prostoru je zobrazen vývoj odhadu center komponent,

вправо - je zobrazen vývoj ukazovátka, které přiřazuje přicházející data do jednotlivých komponent.



Z obrázků je patrné, že výsledek není dobrý. Komponenty se “slezly na jednu hromadu” prakticky bez ohledu na existující datové klastry. Jedním z patrných důvodů je to, že z jízdy po dálnici je relativně málo dat a ta nestačí k tomu aby si “udržela” svou komponentu. Konečný závěr je následující: pro odhad těchto dat je potřeba mnohem více využít apriorní znalost (obecnou i té z apriorních dat).

Řešení s důrazem na apriorní informaci

Přímá aplikace algoritmu odhadování směsi komponent pro dvourozměrná data rychlost auta - moment motoru prakticky selhala. Ačkoli jednotlivé shluky v datovém prostoru byly dokonce okem postřehnutelné, vzhledem k obecné inicializaci algoritmu a také proto, že datové shluky měly poměrně daleko ke gaussovskému tvaru, jak se pro komponenty předpokládá, nebyla detekce všech shluků úspěšná⁶. Proto chceme v dalším pokusu dát mnohem větší důraz na apriorní informaci, v tomto případě získanou právě z analýzy datového prostoru. V tomto případě, kdy datový prostor je dimenze dva a lze jej zobrazit v grafu, nebude analýza složitá. Nicméně, i ve složitějších případech lze, i když mnohem pracněji, takovou analýzu provést. Apriorní informaci vložíme do inicializace algoritmu, a to ve dvou krocích:

1. Odhadneme apriorní centra a kovarianční matice jednotlivých komponent (jde o pozice a šířky apriorních komponent).
2. Určíme, s jakou silou se tyto apriorní charakteristiky mají prosadit proti přicházejícím datům (jak moc máme apriorní informaci fixovat).

Ad 1. Již pouhým pohledem na graf datového prostoru můžeme lokalizovat přibližně sedm základních pozic pro komponenty. Jsou to body

⁶Skutečně, když se podíváme na obrázek dat na začátku této kapitoly vidíme, že zejména u datového shluku pocházejícího z jízdy na dálnici, máme relativně málo dat a o nějakém kopečku ve tvaru Gaussovky nemůže být řeč.

[50;-20]; [50;0]; [50;80]; [80;80]; [100;-20]; [100;0]; [130;100]

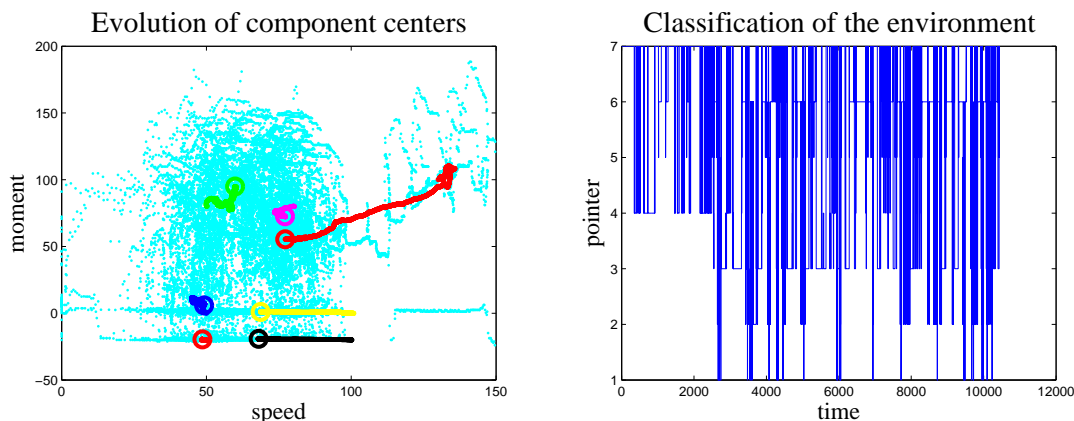
kovariance můžeme určit jako diagonální (Gaussovky s průřezem kružnice) a hodnotami na diagonále (tj. směrodatnými odchylkami jednotlivých veličin) např. 15 (šířka Gaussovky je $3 \times$ směrodatná odchylka).

Ad 2. Při konstrukci statistik budeme postupovat stejně, jako v předchozím příkladě - z modelů komponent s apriorními bodovými odhady parametrů nasimulujeme data a z nich zkonstruujeme statistiky - viz (7.6) a (7.7). Tím jsme realizovali krok 1. Určili jsme statistiky, které “posadí” komponenty tak, jak jsme určili. Nyní jim máme přiřadit významnost. Tak, jak jsme statistiky určili, mají váhu tolika dat, kolik jsme jich nagenerovali a z kolika jsou určeny. To by v podstatě mohlo stačit. Pokud ale chceme počátečním statistikám přiřadit jen malou významnost (jsme přesvědčeni, že odhad i z malého počtu změřených dat “nezabloudí”, měli bychom k dispozici jen velmi málo dat pro určení statistik - v extrémním případě jen jedno měření. Proto lze postupovat takto:

Přidáme-li k informační matici V několik (např. n) stejných regresních vektorů $\Psi\Psi'$, pak je to totéž, jako bychom přidali vektor $n\Psi\Psi'$. A naopak, jestliže informační matici, ke které jsme přidali n regresních vektorů vydělíme n , dostaneme matici s přidáním jediným vektorem. Odtud plyne: Vyrobíme-li informační matici a počítadlo ze stovky dat a matici i počítadlo vydělíme stem, máme statistiky jakoby z jediného datového záznamu. Jestliže pak informační matici i počítadlo násobíme číslem k , dostaneme statistiky se stejnými bodovými odhady jako mají původní statistiky ze stovky dat, ale jejich významnost odpovídá k datům. Toto k pak můžeme nastavit buď vzhledem k celkovému množství zpracovávaných dat, nebo experimentálně tak, aby se odhady držely přibližně tak, kde je chceme a přitom bylo zřejmé, že se doladují podle měřených dat.

Uvedené body 1. a 2. mohou sloužit jako jeden z obecných návodů jak zacházet s apriorní znalostí a jak ji prosazovat proti měřeným datům.

Výsledkem jsou následující obrázky:



Vidíme, že situace se trochu zlepšila, nicméně spokojeni nejsme. Naši nespokojenost vyvolává především průběh ukazovátka, který je značně chaotický.

Řešení motivované odhadem směsi komponent

To co nám v předchozích pokusech nejvíce vadilo, byla neschopnost modelu udržet udržet některé pozice komponent. Důvodem bylo zřejmě to, že odpovídající shluky byly málo dotovány daty (např. relativně krátká jízda po dálnici) a tak “silnější” shluky příslušnou komponentu “přetáhly” na svoji stranu. Pokud bychom tomuto přetažení čelili tím, že bychom dali větší váhu apriorní informaci, došli bychom až do situace, že počáteční odhady parametrů zůstanou fixovány a měřená data je prakticky neovlivní. Proto provedeme dvě základní změny:

1. Datový prostor vytvořený apriorními daty⁷ zhruba rozdělíme na oblasti, kde čekáme (nebo spíše pozorujeme) datové shluky. Pro každou vytvořenou oblast najdeme v datovém prostoru ta data, která jí náleží, a těmito daty odhadneme příslušnou komponentu.
2. Algoritmus klasifikace rozdělíme na dvě části (tak, jak se to při klasifikaci většinou dělá): *(i)* učení - tomu odpovídá odhad komponent na příslušných oblastech a *(ii)* testování - při kterém na základě měřených dat odhadujeme aktivní komponenty, ale už neprovádíme nové odhady parametrů komponent.

Tímto způsobem zabráníme přílišné volnosti algoritmu, který dovoluje přesuny komponent mezi klastry a při tom ponecháme dostatečnou volnost pro “usazení” komponent na klastrech, protože nijak nebráníme volnému pohybu komponenty uvnitř své oblasti při její identifikaci.

V našem případě statických komponent je situace velmi jednoduchá. Pohledem na datový prostor určíme hranice pro intervalové dělení hodnot jednotlivých veličin - rychlost: 20 - 70 - 105 a moment: -5 - 10. Dostáváme tak $4 \times 3 = 12$ obdélníkových oblastí, které nabídneme pro vytvoření komponent. Pro každou oblast najdeme odpovídající data a pro ně odhadneme statický regresní model. Tak vytvoříme komponenty. Statistiku pro ukazovátka vytvoříme jako vektor jehož prvky jsou počty dat, příslušející jednotlivým oblastem (komponentám).

Po naučení algoritmu postupně měříme přicházející data, spočteme pro ně “vzdálenosti” od komponent, korigujeme parametrem ukazovátka a po normování na jednotkový součet dostáváme pravděpodobnosti w_t toho, že jednotlivé komponenty jsou aktivní. Zvolíme komponentu s největší pravděpodobností.

Algoritmus klasifikace je prováděn následujícím programem

⁷Apriorní data v tomto případě musí být dosti rozsáhlá. Algoritmus bude čerpat informaci jen z nich. Další měřená data se buď nevyužijí, nebo je možno použít k aktualizaci algoritmu.

```

clc, clear all, close all
addpath 'funct'
% Estimation of scalar mixture with static pointer
% - data from driven car
% - estimation + clustering

nd=30000; % number of data for est.
b1=[20 70 105]; % borders for y
b2=[-5 10];
nc=(length(b1)+1)*(length(b2)+1);

load dataKF
csl=DataKF.cslen;
load dt27 % data load
y=dt27(:,1:end); % output
ny=size(y,1);

% INITIALIZATION
w(:,1)=ones(nc,1); % vector weights
for c=1:nc, V{c}=1e-8*eye(3); ka(c)=1; end

% ESTIMATION (learning)
for t=1:75000 % use all data for learning
    c1=sum(y(1,t)>b1); % horizontal position (speed)
    c2=sum(y(2,t)>b2); % vertical position (moment)
    c=length(b1)*c1+c2+1; % coding of the window
    Ps=[y(:,t); 1]; % regression vector
    V{c}=V{c}+Ps*Ps'; % information matrix
    ka(c)=ka(c)+1; % counter
end
for c=1:nc
    [th{c} co{c}]=v2thN(V{c}/ka(c),ny); % comp. point estimates
end
nu=ka(:); %ones(nc,1); % pointer statistics
al=fnorm(nu); % pointer point estimates
disp('Learning ended')

% CLASSIFICATION
nd=7500;
for t=2:nd % TIME LOOP _____

    yt=y(:,t); % output measuring

    % distances of yt from components

```

```

for c=1:nc
    [xxx p(c,1)]=GaussN(yt,th{c},co{c}); % value of component
end
p=p-max(p); p=exp(p);
if p'*p<1e-18, p=ones(size(p)); end
wp=p.*al; % new vector weights

% normalizariion of weights
w(:,t)=wp/sum(wp); % normalization of w

% update of statistics
nu=nu+w(:,t); % Dirichlet stats
al=fnorm(nu); % point estimate
end

[xxx ct]=max(w); % active components

% RESULTS
aMixStLr2_R

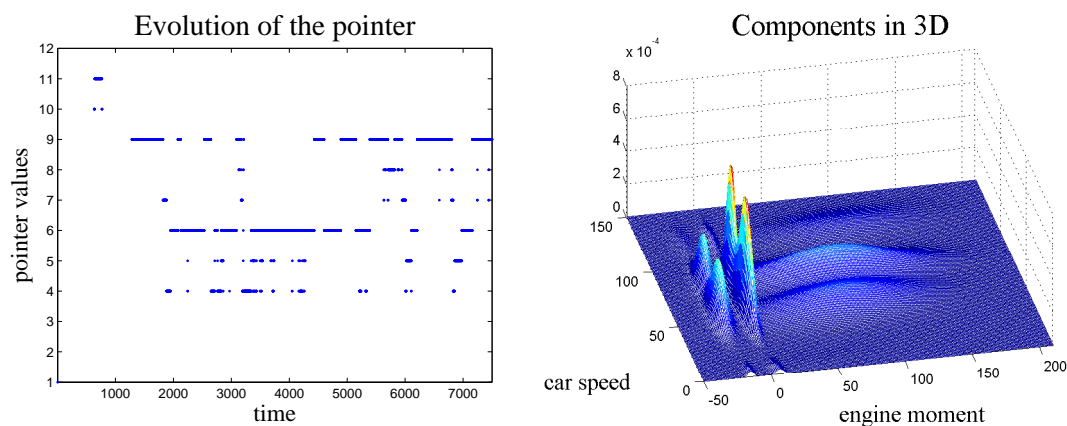
```

Vektor stacionárních vah α jednotlivých komponent je

c	1	2	3	4	5	6	7	
α_c	0.0026	0.0033	0.0052	0.0764	0.0965	0.2779	0.0484	
				8	9	10	11	12
				0.0795	0.3074	0.0058	0.0062	0.0909

Odtud je patrné, že komponenty 1, 2, 3, 10 a 11 prakticky neexistují - v dané oblasti nebyl nalezen podstatný shluk. Tyto komponenty by se mohly z modelu vyjmout.

Klasifikace má následující výsledek



Je zřejmé, že

1. komponenty “sedí” na rozumných místech, přičemž nerozumné zmizely.
2. Průběh hodnot ukazovátka se uklidnil a na začátku skutečně indukují jízdu po dálnici.

Netvrdíme, že toto řešení je ideální, nicméně jsme s ním celkem spokojeni.

8 Odhad dynamické směsi

Hodiny: 13

Aplikace: předpověď módu jedoucího auta.

8.1 Odhad směsi s dynamickým ukazovátkem

Model směsi komponent je opět tvořen množinou n_c komponent a diskretním ukazovátkem c_t , které uvažujeme ve formě Markovského řetězce, tedy procesu závislého na minulé hodnotě c_{t-1} .

Komponenty

Komponenty jsou stejné, jako u směsi se statickým ukazovátkem

$$f(d_t|c, \Theta_c, \psi_t), \quad c \in \{1, 2, \dots, n_c\} = c^*,$$

kde d_t jsou modelovaná data (řízení neuvažujeme), c označuje číslo komponenty, Θ_c jsou parametry c -té komponenty a ψ_t je regresní vektor (který bereme společný všem komponentám).

Ukazovátko

Ukazovátko je v tomto případě závislé na své minulé hodnotě a modelujeme jej jako Markovský řetězec pomocí stavového modelu

$$f(c_t|c_{t-1}, \alpha, d(t-1)) \underbrace{=}_{\text{nezávislost}} f(c_t|c_{t-1}, \alpha) = \alpha_{c_t|c_{t-1}},$$

kde index $c_t|c_{t-1}$ má stejný význam jako c_t, c_{t-1} , pouze se jím zdůrazňuje, že se jedná o podmíněnou pravděpodobnost tak, jako u hp modelu. Jedná se tedy o maticový parametr, kde ale sloupcovým indexem je c_t a řádkovým c_{t-1} . Jedná se o diskretní model, o kterém jsme již mluvili v kapitole o modelech.

Odhad modelu dynamické směsi je zatížen stejným problémem jako u statické směsi. Přímočarý algoritmus je s postupem času stále složitější, a rychle se stane nespočitatelným. Aproximaci, která algoritmus odhadu zjednoduší, provedeme obdobně jako ve statickém případě. Funkci $\delta(c, \tilde{c}; c_t, c_{t-1}) = \delta(c, c_t) \delta(\tilde{c}, c_{t-1})$, kde c a \tilde{c} jsou obecné indexy aktuální a minulé aktivní komponenty a který říká, že aktuální přechod mezi komponentami byl z c_{t-1} na c_t , odhadneme jako jeho podmíněnou střední hodnotou

$$\begin{aligned} E[\delta(c, \tilde{c}; c_t, c_{t-1}) | d(t)] &= \sum_{c=1}^{n_c} \delta(c, \tilde{c}; c_t, c_{t-1}) f(c_t, c_{t-1} | d(t)) = \\ &= Pr(c_t = c, c_{t-1} = \tilde{c} | d(t)) = W_{c, \tilde{c}; t}. \end{aligned}$$

Výsledkem je matice přechodových pravděpodobností, kde libovolný prvek matice W_t , tedy prvek $W_{i,j;t}$, určuje pravděpodobnost, že systém přešel z módu j do módu i ⁸.

Úlohu klasifikace aktuální datové položky d_t do tříd, které představují jednotlivé komponenty provedeme tak, že v aktuální váhové matici sečteme řádky a v součtu najdeme maximální prvek. Index tohoto prvku určuje třídu, do které patří d_t .

Úlohu predikce lze řešit jen při použití dynamického modelu pro ukazovátka. Jedná se o důležitou úlohu - předpověď v klasifikaci, tj. odhad komponenty, do které bude patřit příští nebo další budoucí datový vzorek. Tato úloha je velice důležitá jestliže sledujeme pracovní módy určitého systému (například jedoucího automobilu) a mezi módy jsou některé, které představují určité nebezpečí (např. příliš vysoká rychlost vzhledem k povaze vozovky). Indikace vlastní havárie již přichází pozdě. Je třeba předvídat budoucí módy a jestliže se v nich objeví havárie, je třeba řidiče varovat. Úloze predikce aktivní komponenty se budeme věnovat na konci této kapitoly.

⁸Jedná se o tabulku sružených pravděpodobností - pravděpodobnost, že systém byl v \dots a přešel do \dots . Její normalizace se provede tak, aby součet všech prvků byl roven jedné. Přechod na podmíněné pravděpodobnosti spočívá jen v renormalizaci tak, aby součty v řádcích byly rovny jedné.

Podobně je tomu i s odhadem parametru α . Je to matice podmíněných pravděpodobností, ale lze ji odhadovat jako pravděpodobnosti sružené a ihned normalizovat na součty řádků rovny jedné.

Odvození algoritmu odhadu

Při odvození algoritmu klasifikace a predikce, který je ovšem vždy spojen s úlohou odhadu parametrů jak komponent tak i modelu ukazovátka, budeme postupovat obdobně jako v případě statické směsi.

Vyjdeme ze sdružené hp neznámých veličin c_t , c_{t-1} , Θ a α podmíněné dosud změřenými daty

$$\mathcal{J} = f(c_t, c_{t-1}, \Theta, \alpha | d(t)).$$

Použitím Bayesova vzorce, rozvojem sdružené hp a sdružením členů, které k sobě podle parametrů patří dostaneme

$$\begin{aligned} \mathcal{J} &= f(d_t | c_t, \psi_t, \Theta_c) f(\Theta | d(t-1)) \times f(c_t, c_{t-1} | \alpha, d_{t-1}) f(\alpha | d(t-1)) = \\ &= f(d_t | c_t, \psi_t, \Theta_c) f(\Theta | d(t-1)) \times \alpha_{c_t | c_{t-1}} f(\alpha | d(t-1)) \times f(c_{t-1} | d_{t-1}), \end{aligned} \quad (8.1)$$

kde jednotlivé hp zleva jsou: model c_t -té komponenty, apriorní hp pro odhad Θ , model ukazovátka $f(c_t | c_{t-1}, \alpha) = \alpha_{c_t | c_{t-1}}$, apriorní diskretní hp pro odhad ukazovátka a apriorní hp pro odhad α . Použité předpoklady nezávislosti jsou prakticky stejné, jak v případě statické směsi; model komponenty předpokládáme závislý jen na aktuální komponentě, nikoli minulé.

Z hp \mathcal{J} lze zkonstruovat matici sdružených pravděpodobností $W_{c, \tilde{c}; t}$ ze které lze marginalizací určit vektor vah $w_{c; t}$ pomocí kterého se provádí klasifikace dat

$$\begin{aligned} W_{c_t, c_{t-1}; t} &= f(c_t, c_{t-1} | d(t)) = \int_{\Theta^*} \int_{\alpha^*} \mathcal{J} d\alpha d\Theta = \\ &= \int_{\Theta^*} f(d_t | c_t, \psi_t, \Theta_c) f(\Theta | d(t-1)) d\Theta \int_{\alpha^*} \alpha_{c_t | c_{t-1}} f(\alpha | d(t-1)) d\alpha f(c_{t-1} | d(t-1)), \end{aligned} \quad (8.2)$$

kde proti statickému případu přibyla diskretní hp $f(c_{t-1} | d(t-1))$, což je obyčejný vektor pravděpodobností s významem apriorního odhadu aktivní komponenty c_t . Tento vektor je apriorním rozdělením odhadu aktivní komponenty $w_{c_t; t}$; tedy přísluší mu označení

$$f(c_{t-1} | d(t-1)) = w_{c_{t-1}; t-1}.$$

Dále si všimneme části \mathcal{J} týkající se Θ a α .

Nejprve

$$\mathcal{P}_{c_t}^\Theta = f(d_t | c_t, \psi_t, \Theta_{c_t}) f(\Theta | d(t-1)),$$

což je výraz zcela stejný jako ve statickém případě. Jedná se o nenormovanou aposteriorní hp pro Θ s aproximovanou normalizační konstantou

$$\mathcal{I}_{c_t}^\Theta = f(d_t | c_t, \psi_t, \hat{\Theta}_{c_t; t-1}),$$

kde jsme opět použili bodový odhad parametru Θ .

Druhé části výrazů (8.1) a (8.2) jsou

$$\mathcal{P}_{c_t|c_{t-1}}^\alpha = \alpha_{c_t|c_{t-1}} f(\alpha|d(t-1))$$

a její normalizační konstanta

$$\mathcal{I}_{c_t|c_{t-1}}^\alpha = \int_{\alpha^*} \alpha_{c_t|c_{t-1}} f(\alpha|d(t-1)) d\alpha.$$

Dále ukážeme konstrukci těchto výrazů a jejich použití pro odhad parametrů a klasifikaci dat. Predikce komponent je nová záležitost a tu si necháme na konec kapitoly.

Odhad parametrů Θ

Odhad parametru Θ je stejný jako pro statickou směs. Přepočet statistik je

$$V_{c;t} = V_{c;t-1} + w_{c;t} \Psi_t \Psi_t', \quad \kappa_{c;t} = \kappa_{c;t-1} + w_{c;t},$$

pro $c = 1, 2, \dots, n_c$. Vektor w_t dostaneme sečtením řádek matice pravděpodobností W_t , tedy $w_{c;t} = \sum_{\tilde{c}=1}^{n_c} W_{c,\tilde{c};t}$.

Z přepočetných statistik můžeme buď určit bodové odhady parametrů nebo zkonstruovat celou aposteriorní hp.

Odhad parametru α

Podobně jako pro parametr Θ počítáme podle Bayesova vzorce

$$\mathcal{P}_{c_t|\tilde{c}}^\alpha = f(\alpha|d(t)) \propto \alpha_{c_t|\tilde{c}} f(\alpha|d(t-1)),$$

kde opět pro hp parametrů zvolíme konjugovanou hp ve tvaru Dirichletova rozdělení a model vyjádříme v součinném tvaru

$$\prod_{c,\tilde{c} \in c^*} \alpha_{c|\tilde{c}}^{\nu_{c,\tilde{c};t}} \propto \prod_{c,\tilde{c} \in c^*} \alpha_{c|\tilde{c}}^{\delta(c,\tilde{c};c_t,c_{t-1})} \prod_{c,\tilde{c} \in c^*} \alpha_{c|\tilde{c}}^{\nu_{c,\tilde{c};t-1}}.$$

Po aproximaci $w \rightarrow \delta$ dostaneme přepočet statistik

$$\nu_{c,\tilde{c};t} = \nu_{c,\tilde{c};t-1} + W_{c,\tilde{c};t},$$

pro $c = 1, 2, \dots, n_c$, $\tilde{c} = 1, 2, \dots, n_c$.

Výsledkem může být opět bodový odhad nebo celá aposteriorní hp.

Výpočet integrálůIntegrál \mathcal{I}_c^Θ :

$$\mathcal{I}_c^\Theta = \int_{\Theta^*} f(d_t|c, \psi_t, \Theta_c) f(\Theta|d(t-1)) d\Theta.$$

Tento integrál počítáme opět pomocí bodového odhadu $\hat{\Theta}_{t-1}$ parametru Θ . Stejně jako pro statickou směs platí

$$\mathcal{I}_c^\Theta = f(d_t|c, \psi_t, \hat{\Theta}_{c;t-1}) \equiv \hat{m}_{c;t},$$

tedy je to přímo model c -té komponenty s dosazeným bodovým odhadem za parametr Θ .

Integrál \mathcal{I}_c^α :

$$\mathcal{I}_{c|\tilde{c}}^\alpha = \int_{\alpha^*} \alpha_{c|\tilde{c}} f(\alpha|d(t-1)) d\alpha = \frac{\nu_{c|\tilde{c};t-1}}{\sum_{i=1}^{n_c} \nu_{i|\tilde{c};t-1}} = \hat{\alpha}_{c|\tilde{c};t-1},$$

což je bodový odhad $\hat{\alpha}_{c|\tilde{c};t-1}$ ⁹ parametru $\alpha_{c|\tilde{c}}$.

Matice vah $W_{c,\tilde{c};t}$ podle (8.2) dostaneme takto

$$W_{c,\tilde{c};t} = \mathcal{I}_c^\Theta \mathcal{I}_{c|\tilde{c}}^\alpha w_{\tilde{c};t-1} = \hat{m}_{c;t} \hat{\alpha}_{c|\tilde{c}} w_{\tilde{c};t-1}. \quad (8.3)$$

Vektor vah $w_{c;t}$ je marginála matice vah, tedy

$$w_{c;t} = \sum_{\tilde{c}=1}^{n_c} W_{c,\tilde{c};t}. \quad (8.4)$$

Nyní lze uvést **algoritmus odhadu** pro jeden krok v časové smyčce

Máme k dispozici:

- statistiky odhadu z minulého kroku $V_{c;t-1}$, $\kappa_{c;t-1}$, $\nu_{c;t-1}$ pro $c \in c^*$,
- odhady parametrů spočtené ze statistik $\hat{\Theta}_{t-1}$ a $\hat{\alpha}_{t-1}$,
- odhady komponent $\hat{m}_{c;t} = f(d_t|c, \psi_t, \hat{\Theta}_{c;t-1})$.

Provedeme

⁹Při odvození výsledku integrálu je třeba použít vícerozměrnou beta funkci

$$B(\nu_{i,j}) = \prod_{j=1}^n \frac{\prod_{i=1}^n \Gamma(\nu_{i,j})}{\Gamma(\sum_{i=1}^n \nu_{i,j})},$$

kteřá je definovaná pro maticový argument ν stupně n .

1. Změříme nová data d_t .
2. Spočteme “vzdálenosti” komponent od d_t : $\hat{m}_{c;t} = f\left(d_t|c, \psi_t, \hat{\Theta}_{c;t-1}\right)$.
3. Určíme integrály $\mathcal{I}_c^\Theta = \hat{m}_{c;t}$ a $\mathcal{I}_{c|\tilde{c}}^\alpha = \hat{\alpha}_{c|\tilde{c};t-1}$.
4. Spočteme matici vah $W_{c|\tilde{c};t} = \mathcal{I}_c^\Theta \mathcal{I}_{c|\tilde{c}}^\alpha w_{c,\tilde{c};t-1} = \hat{m}_{c;t} \hat{\alpha}_{c|\tilde{c};t-1} w_{\tilde{c}|\tilde{c};t-1}$ a vektor vah $w_{c;t} = \sum_{\tilde{c}=1}^{n_c} W_{c,\tilde{c};t}$ (\tilde{c} odpovídá komponentě c_{t-2}).
5. Provedeme přepočítání statistik obou modelů

$$V_{c;t} = V_{c;t-1} + w_{c;t} \Psi_t \Psi_t', \quad \kappa_{c;t} = \kappa_{c;t-1} + w_{c;t},$$

$$\nu_{c|\tilde{c};t} = \nu_{c|\tilde{c};t-1} + W_{c,\tilde{c};t}.$$

6. Vypočteme bodové odhady parametrů komponent

$$\hat{\theta}_{c;t} = V_{\psi;t}^{-1} V_{y\psi}, \quad \hat{r}_{c;t} = \frac{V_y - V_{y\psi}' V_{\psi}^{-1} V_{y\psi}}{\kappa_{c;t}}, \quad \hat{\alpha}_{c|\tilde{c};t} = \frac{\nu_{c,\tilde{c};t}}{\sum_{i=1}^{n_c} \nu_{c|i;t}},$$

kde $V_{c;t} = \begin{bmatrix} V_y & V_{y\psi} \\ V_{y\psi}' & V_{\psi} \end{bmatrix}$ je rozdělená informační matice komponenty c .

7. Zvýšíme časový index $t = t + 1$ a jdeme na bod 1.

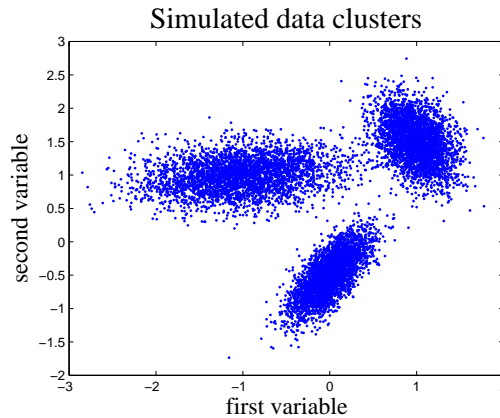
Výsledkem je

- **klasifikace** datového vzorku d_t do komponenty s maximální vahou $w_{c;t}$,
- použití datového vzorku pro učení, tj. **odhad parametrů** komponent a ukazovátka.

Vzorový příklad

Uvedený algoritmus budeme demonstrovat na příkladě se simulovanými dvourozměrnými daty. Cílem tohoto příkladu je obdobně jako u statické směsi ukázat programové příkazy, které algoritmus realizují. I když se matematika spojená s algoritmem může zdát dosti divoká, vlastní programové příkazy i jejich kombinace jsou velmi jednoduché.

Simulovanou soustavu zvolíme jako množinu tří statických komponent, které v časové smyčce přecházejí pravidelně jedna na druhou 1 – 2 – 3 – 1 atd. Datové shluky jsou následující



Program, který data nasimuloval je

```

%% Simulation of two dimensional mixture with dynamic pointer
% -----
clc, clear all, close all
rand('seed',1359),randn('seed',9753)

nd=20000;                                % number of simulated data

% centers and covariances of all components
cen{1}=[1;1.5];      cov{1}=.05*[1 -.5;-.5 2];
cen{2}=[-1;1];      cov{2}=.3*[1 .1;.1 .2];
cen{3}=[0;-.5];     cov{3}=.05*[1 1;1 2];

% pointer parameter
%%% pp=[0 1 0; 0 0 1; 1 0 0];
pp=[.1 .8 .1; .1 .1 .8; .8 .1 .1];
pp=fnorm(pp,2);
cp=cumsum(pp')';

nc=length(cen);
ny=length(cen{1});
for c=1:nc, sdv{c}=uut(cov{c}); end      % cov -> std
ct(1)=1;                                % initial component

% SIMULATION
for t=2:nd
    ct(t)=sum(rand>cp(ct(t-1),:))+1;      % pointer generation
    yt(:,t)=cen{ct(t)}+sdv{ct(t)}*randn(ny,1); % data generation
end

```

```
% SAVING THE DATA SAMPLE
```

```
save dtCyc yt ct
```

```
% save data , pointer
```

Program je celkem přehledný, pouze funkce `uut` potřebuje vysvětlení. Je to funkce, která provádí odmocninu pozitivně definitní kovarianční matice C na součin dvou faktorů S a S' , kde faktorem je horní trojúhelníková matice

$$C = SS'.$$

Je to obdoba směrodatné odchylky počítané z rozptylu. Funkce je uvedena v Knihovně funkcí 11.14.

Vlastní program, realizující algoritmus odhadu dynamické směsi je velmi podobný algoritmu pro statickou směs. Pouze místo přímého výpočtu aktuálního vektoru vah komponent w_t se nejdříve počítá matice sdružených vah W_t jejíž prvky udávají pravděpodobnosti toho, že minulá aktivní komponenta byla j a současná je i , tedy

$$W_{i,j;t} = Pr(c_t = i, c_{t-1} = j | d(t)).$$

Tato matice je normalizována tak, že součet všech jejích prvků je roven jedné.

Vektor w_t se pak spočte jako marginála této sdružené pravděpodobnosti pro veličinu c_t

$$w_{c;t} = \sum_{k=1}^{n_c} W_{c,k;t}.$$

Program je následující

```
clc , clear all , close all
addpath 'funct'
```

```
%% Estimation of mixture with dynamic pointer
```

```
%% - data from driven car
```

```
%% - two dimensional output (speed , moment)
```

```
%% - estimation + clustering
```

```
nc=3; % number of components
```

```
nd=10000; % length of data
```

```
ni=200; % length of initial data
```

```
% test (circle of components)
```

```
load dtCyc yt ct % data load
```

```
cc=ct; clear ct;
```

```
y=yt(:,1:nd); % output
```

```
yIni=yt(:,(nd+1):(nd+ni)); % initial output
```

```
ph=length(yIni); phc=10;
```

```

% INITIALIZATION
typ=1; % dynamic mixture
[th co al V ka nu]=iniMix(nc,yIni,typ,ph,phc);
w=zeros(nc,nd); Et1=w; Et2=w;
for c=1:nc,
    Et1(c,1)=th{c}(1); % remembering initial
    Et2(c,1)=th{c}(2); % parameters
end
w(:,1)=ones(nc,1)/nc; % vector weights

% ESTIMATION
fprintf('running ..... | \ n. '),itime=0;
for t=2:nd % TIME LOOP _____
    itime=itime+1;
    if itime>(nd-1)/20, fprintf('.'), itime=0; end

    yt=y(:,t); % output measuring

    % distances of yt from components
    for c=1:nc
        [xxx p(c)]=Gauss2(yt,th{c},co{c}); % value of component
    end
    p=p-min(p); p=exp(p); % log -> exp
    if p'*p<1e-18, p=ones(size(p)); end % patch for all 0
    wm=(w(:,t-1)*p).*al; % matrix weights
    wm=wm/sum(sum(wm)); % normalization of wm
    w(:,t)=sum(wm); % new vector weights

    % update of statistics
    Ps=[yt;1]; % regression vector
    for c=1:nc
        V{c}=V{c}+w(c,t)*Ps*Ps'; % info. matrix
        ka(c)=ka(c)+w(c,t); % counter
        [th{c} co{c}]=v2thN(V{c}/ka(c),2); % point estimates
    end
    nu=nu+wm; % Dirichlet stats
    al=fnorm(nu,2); % point estimate

    for c=1:nc, % remembering of
        Et1(c,t)=th{c}(1); % evolution
        Et2(c,t)=th{c}(2); % of parameter
    end % estimates
end
end

```



```
[xxx ct]=max(w); % active components

% RESULTS
vMixDyEstSi_R
```

8.2 Predikce aktivní komponenty

Jak již bylo zmíněno, o predikci aktivní komponenty má smysl mluvit teprve ve spojení s odhadem dynamické směsi. U statické směsi je sice možno uvažovat dynamické komponenty ze kterých je možno predikovat. Je to ale predikce jen v rámci jedné komponenty. Změnu aktivní komponenty nejsme schopni předpovědět. A je to právě změna komponenty, co bývá při předpovědi nejdůležitější. Například - rychlost automobilu se postupně zvětšuje a nás zajímá, kdy se dostaneme do módu, kdy nehoda je již jen dílem náhody. Tj. módu, který odpovídá nehodové komponentě.

Predikci n_p kroků dopředu lze s modelem směsi komponent formulovat takto: určete hp

$$f(c_{t+n_p}|d(t)). \quad (8.5)$$

Zde si nebudeme komplikovat situaci řízením, nebo řízení považujeme za předem známé na celém intervalu predikce. Uvedená hp říká, že odhadujeme aktivní komponentu v čase $t + n_p$, tedy n_p kroků dopředu od současného okamžiku t . Známe minulé data včetně současně naměřeného d_t .

Pro konstrukci této prediktivní hp budeme opět uvažovat sdruženou hp všech neznámých veličin (pro přehlednost položíme $n_p = 2$)

$$\mathcal{J} = f(c_{t+2}, c_{t+1}, c_t, c_{t-1}, \Theta, \alpha | d(t)),$$

ze které prediktivní hp (8.5) dostaneme sumací přes c_{t+1}, c_t a c_{t-1} a integrací přes Θ a α . Použitím Bayesova vzorce a rozvojem podle řetězového pravidla (podobně jako při odvození algoritmu odhadu) dostaneme

$$\begin{aligned} \mathcal{J} &\propto f(d_t, c_{t+2}, c_{t+1}, c_t, c_{t-1}, \Theta, \alpha | d(t-1)) = \\ &= f(d_t | c_t, \Theta, \psi_t) f(c_{t+2} | c_{t+1}, \alpha) f(c_{t+1} | c_t, \alpha) f(c_t | c_{t-1}, \alpha) \times \\ &\quad \times f(c_{t-1} | d(t-1)) f(\Theta | d(t-1)) f(\alpha | d(t-1)), \end{aligned}$$

kde $f(c_{t-1} | d(t-1)) = w_{c_{t-1}, t-1}$ je apriorní rozdění pro odhad aktivní komponenty.

Pokud některé proměnné v podmínkách chybí, pak je to z důvodu nezávislosti.

Sdruženou hustotu uspořádáme podle parametrů Θ a α a predikci (8.5) můžeme vyjádřit takto

$$f(c_{t+2} | d(t)) = \sum_{c_{t+1}} \sum_{c_t} \sum_{c_{t-1}} \int_{\Theta^*} f(d_t | c_t, \Theta, \psi_t) f(\Theta | d(t-1)) d\Theta \times$$

$$\times \int_{\alpha^*} f(c_{t+2}|c_{t+1}, \alpha) f(c_{t+1}|c_t, \alpha) f(c_t|c_{t-1}, \alpha) f(\alpha|d(t-1)) d\alpha \times w_{c_{t-1};t-1}.$$

K vyčíslení obou integrálů použijeme bodové odhady parametrů¹⁰. Tak dostaneme

$$f(c_{t+2}|d(t)) \doteq \sum_{c_{t+1}} \sum_{c_t} \sum_{c_{t-1}} \hat{m}_{c_t;t} \times \hat{\alpha}_{c_{t+2}|c_{t+1};t-1} \hat{\alpha}_{c_{t+1}|c_t;t-1} \hat{\alpha}_{c_t|c_{t-1};t-1} \times w_{c_{t-1};t-1}.$$

Porovnáme-li tento výraz s výrazem pro výpočet vektoru vah $w_{c_t;t}$ (8.2), vidíme, že $w_{c_t;t} \doteq \sum_{c_{t-1}} \hat{m}_{c_t;t} \hat{\alpha}_{c_t|c_{t-1};t-1} w_{c_{t-1};t-1}$, a tedy po dosazení do hp predikce

$$f(c_{t+2}|d(t)) \doteq \sum_{c_{t+1}} \sum_{c_t} \hat{\alpha}_{c_{t+2}|c_{t+1};t-1} \hat{\alpha}_{c_{t+1}|c_t;t-1} w_{c_t;t}.$$

Maticově lze tento výsledek zapsat jako

$$f(c_{t+2}|d(t)) = (\hat{\alpha}_{t-1})^2 w_t.$$

Tento výsledek lze velmi dobře interpretovat: (i) odhadneme aktuální komponentu na základě měřených dat, (ii) tuto komponentu dvakrát “naslepo” predikujeme, už bez podpory dat. Nicméně, pokud je v přepínání komponent alespoň trochu pravidelnosti, má tato predikce smysl.

Obecně tedy bude

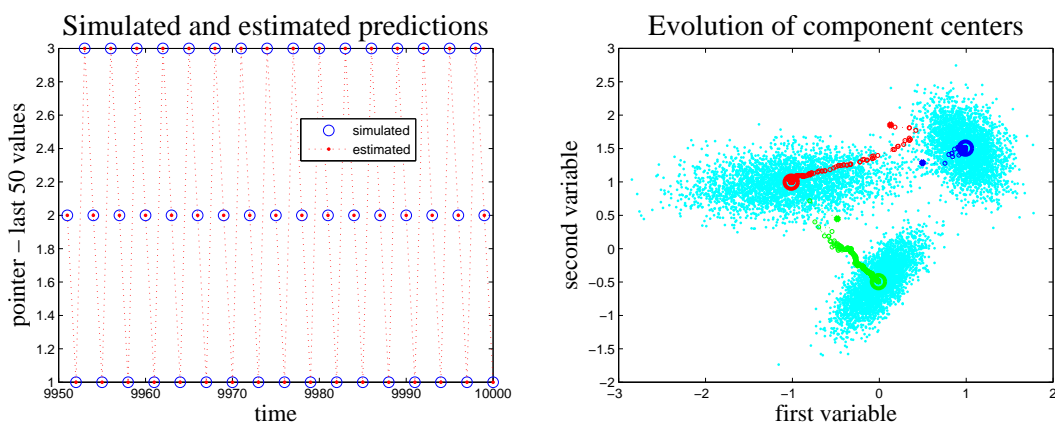
$$f(c_{t+n_p}|d(t)) = (\hat{\alpha}_{t-1})^{n_p} w_t.$$

Vzorový příklad

V tomto příkladě navážeme na předchozí příklad se simulovanými daty, která jsou generována z komponent, které se pravidelně střídají. To je ideální případ pro predikci, neboť z dobře odhadnuté současné komponenty již poměrně přesně určíme požadovanou budoucí komponentu. Takové pravidelné střídání komponent je však v praxi prakticky nemožné a tedy střídání, i když existuje, je spíše více než méně nepravidelné.

Dvourozměrnou veličinu $y_t = [y_1, y_2]'$ generujeme ze tří komponent, které se pravidelně střídají 1 – 2 – 3 – 1 atd. Generované klastry jsou zobrazeny na straně 110. Výsledek experimentu je následující

¹⁰Připomeňme: Integrál $\int g(x) f(x|d(t-1)) dx$ který vyjadřuje funkci $g(x)$ při neznámém x popsáním pomocí hp $f(x|d(t-1))$ nahradíme pomocí $g(\hat{x}_{t-1})$, kde \hat{x}_{t-1} je bodový odhad neznámé veličiny x počítaný na základě dat $d(t-1)$. Podstatou je nahrazení $f(x|d(t-1))$ Diracovým impulzem $\delta(x; \hat{x}_{t-1})$.



Na levé straně vidíme budoucí aktivní komponenty a jejich odhady (tedy predikce aktivních komponent), na pravé straně je vývoj středů odhadovaných komponent. Jedná se deseti-krokovou předpověď. V tomto jednoduchém případě, kdy komponenty jsou zřetelně odděleny, jsou výsledky přesné i při tak dlouhé předpovědi, jako je 10 kroků.

Poznámka

Jediná potíž, které se ale při náhodném startu algoritmu odhadu nelze vyhnout, je záměna komponent. Při simulaci byla první komponenta ta, co leží vpravo nahoře, druhé vlevo nahoře a třetí dole (viz simulovaná centra v programu). Odhadový algoritmus si ale komponenty přiřadí podle toho, kde náhodou leží apriorní centra - na obrázku je první komponenta červená (dole), druhá modrá (vlevo nahoře) a třetí zelená (vpravo nahoře). V kontrolním grafu bychom pak našli neshodu dat a odhadů. Odhady by byly příslušně posunuty.

Abychom mohli tento posuv korigovat, musíme najít zobrazení, které bude mapovat posunutá čísla komponent na správná. Máme tedy diskrétní veličinu x se třemi hodnotami (v našem případě) s pořadím hodnot ze simulace a veličinu y se stejnými, ale zamíchanými, hodnotami. Transformace mezi x a y je vyjádřena transformační maticí T tak, že $y = Tx$. Jedná se o diskrétní model a odhad jeho přechodové matice. Pro každou dvojici x_i a y_i provedeme přepočít

$$T(x_i, y_i) = T(x_i, y_i) + 1$$

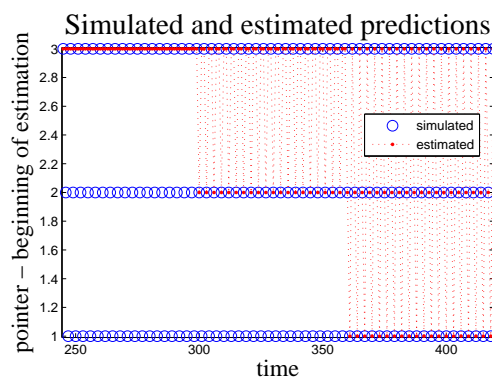
Výslednou matici T lze použít pro korekci posunutí aktivních komponent danou startem algoritmu. Výpočet matice T provádí funkce

$$[pt \ vt] = c2c(cc, ct);$$

kde cc jsou simulovaná centra, ct odhadovaná centra a vt je transformační matice. Vektor pt je konstruován tak, že $pt(cc)$ koresponduje s ct .

Transformační vektor pt v našem příkladě je $pt = [2, 1, 3]$ protože 1. klastr je modelován druhou komponentou, 2. klastr pomocí první a 3. klastr zůstává jako třetí.

Obrázky, které jsme výše demonstrovali jsou z časového intervalu, kdy model byl již odhadnut. Samotný odhad samozřejmě chvíli trvá, protože startujeme z náhodných počátečních hodnot parametrů. Vývoj odhadu center komponent byl již patrný z předchozího obrázku. Nyní ještě ukážeme počáteční odhady aktivních komponent, které vlivem špatného odhadu komponent samotných, budou také nesprávné.



Vidíme, že do kroku 300 se jako aktivní komponenta odhaduje pouze jedna (třetí). Mezi kroky 300 a 360 se odhad trochu zlepšil (odhadují se dvě komponenty). Od kroku 360 až do konce experimentu je již odhad perfektní.

Zkusíme ještě tentýž příklad, ale s více neurčitostí v pořadí, ve kterém jsou komponenty generovány. Parametr α pro simulaci zvolíme

$$\begin{bmatrix} 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \\ 0.8 & 0.1 & 0.1 \end{bmatrix}.$$

Odhad tohoto parametru je

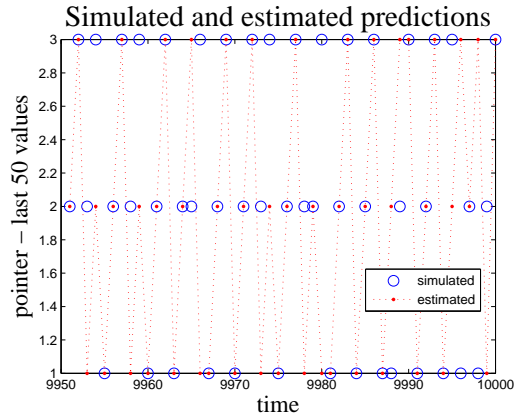
$$\begin{bmatrix} 0.110 & 0.795 & 0.095 \\ 0.092 & 0.111 & 0.797 \\ 0.813 & 0.086 & 0.101 \end{bmatrix},$$

což je poměrně dobré. Odhad center komponent byl

Centra	1. komp.	2. komp.	3. komp.
simulace	$[1, 1.5]'$	$[-1, 1]'$	$[0, -0.5]'$
odhad	$[-0.01, -0.51]'$	$[0.99, 1.5]'$	$[-1, 1]$

což je také velmi dobré (komponenty jsou přeházeny).

Predikce aktivní komponenty na dva kroky dopředu, pro 50 posledních vzorku z experimentu dlouhého 10000 kroků, je vynesena na následujícím obrázku



Vzhledem k parametru α , kde pravděpodobnost chyby predikce pro každou komponentu je 20%, je predikce aktivní komponenty odpovídající.

Program

Řešenou úlohu realizuje následující program

```

clc, clear all, close all
addpath 'funct'
%% Estimation and prediction of mixture with dynamic pointer
%% - simulated two-dimensional data
%% - estimation + prediction
%
% Calls: iniMix, Gauss2, v2thN, c2c

nc=3; % number of components
np=2; % length of prediction
nd=10000; % length of data
ni=200; % length of initial data

% test (circle of components)
load dtCyc yt ct % data load
cc=ct; clear ct;
y=yt(:,1:nd); % output
yIni=yt(:,(nd+1):(nd+ni)); % initial output
ph=length(yIni); phc=10;

% INITIALIZATION
typ=1; % dynamic mixture
[th co al V ka nu]=iniMix(nc, yIni, typ, ph, phc);
w=zeros(nc, nd); Et1=w; Et2=w;

```

```

for c=1:nc,
    Et1(c,1)=th{c}(1);           % remembering initial
    Et2(c,1)=th{c}(2);           % parameters
end
w(:,1)=ones(nc,1)/nc;           % vector weights

% ESTIMATION
fprintf('running .....|\n. '),itime=0;
for t=2:nd                         % TIME LOOP -----
    itime=itime+1;
    if itime>(nd-1)/20, fprintf('.'), itime=0; end

    yt=y(:,t);                     % output measuring

    % distances of yt from components
    for c=1:nc
        [xxx p(c)]=Gauss2(yt,th{c},co{c}); % value of component
    end
    p=p-min(p); p=exp(p);           % log -> exp
    if p'*p<1e-18, p=ones(size(p)); end % patch for all 0
    wm=(w(:,t-1)*p).*al;           % matrix weights
    wm=wm/sum(sum(wm));            % normalization of wm
    w(:,t)=sum(wm);                % new vector weights

    % prediction of components
    wp(:,t)=(al^np)'*w(:,t);
    cn(t)=cc(t+np);

    % update of statistics
    Ps=[yt;1];                       % regression vector
    for c=1:nc
        V{c}=V{c}+w(c,t)*Ps*Ps';    % info. matrix
        ka(c)=ka(c)+w(c,t);         % counter
        [th{c} co{c}]=v2thN(V{c}/ka(c),2); % point estimates
    end

    nu=nu+wm;                         % Dirichlet stats
    al=fnorm(nu,2);                  % point estimate

    for c=1:nc,                       % remembering of
        Et1(c,t)=th{c}(1);           % evolution
        Et2(c,t)=th{c}(2);           % of parameter
    end
end
end

```

```
% RESULTS
vMixDyPreSi_R
```

8.3 Aplikace

Ukázat jednoduchou aplikaci na úlohu předpovědi aktuální komponenty není vůbec snadné. Již samotný odhad matice pravděpodobností přechodu systému mezi jednotlivými módy je dosti náročné na množství potřebných dat (situace je podobná, jako při odhadu kovarianční matice - obecně, pro odhad střední hodnoty jsou potřeba desítky dat, pro odhad kovarianční matice to jsou stovky až tisíce). A to ještě za předpokladu, že přechody mezi komponentami vykazují alespoň základní stabilitu. Tady je potřeba si uvědomit, že to, v čem není určitý řád, prostě nelze odhadnout. Pokusíme se tedy pokračovat v Aplikaci z minulé kapitoly (klasifikace cesty jedoucího automobilu) a typ cesty předpovídat. V další části se pokusíme do předpovědi zařadit ještě další veličiny kromě rychlosti a momentu.

Predikce cesty za jízdy automobilu

K dispozici máme 16 měřených veličin. Z nich jsme vybrali následující veličiny.

veličina	zkratka	číslo
aktuální spotřeba	spotřeba	1
aktuální rychlost automobilu	rychlost	2
otočení volantu	volant	3
stlačení plynového pedálu	plyn	4
tlak na brzdový pedál	brzda	5
zařazený rychlostní stupeň	kvalt	6
moment motoru	moment	7
otáčky motoru	otáčky	8
příčné zrychlení automobilu	zrychlení	9
stáčivá rychlost automobilu	stáčení	10
ujetá vzdálenost / periodu	vzdálenost	11
aktuální čas	čas	12
aktuální poloha X	polohaX	13
aktuální poloha Y	polohaY	14
aktuální nadmořská výška	výška	15
směr jízdy	směr	16

Prvním krokem k budování modelu je výběr vhodných veličin. Chceme využít model dynamické směsi se statickými komponentami¹¹. Komponenty tedy budou regresní modely

¹¹Připomínáme: Statické komponenty popisují datové shluky, které se s časem v datovém prostoru nepohybují (např. při nájedzu na dálnici vzroste rychlost vždy na hodnotu kolem 120 km/h a taková

typu

$$d_t = \theta_c + e_{c;t}, \quad (8.6)$$

kde θ jsou střední hodnoty komponent a $e_{c;t} \sim N(0, r_c)$ kde r_c jsou kovariance komponent - tedy θ_c popisují centra a r_c šířky kopečků v datovém prostoru. d_t je modelovaný vektor jehož složky máme vybrat z veličin z tabulky. Tento měřený datový vektor, jako bod datového prostoru, bude vytvářet datové shluky, na které se budeme (při odhadování) snažit “napasovat” Gaussovky (8.6). Samozřejmě, že různě vybraný datový vektor bude vytvářet různé shluky. Naší snahou je zvolit datový vektor d_t tak, aby shluky byly co nejzřetelnější a tím i nejlépe detekovatelné (jak pro tvorbu modelu, tak i pro odhadování aktuální komponenty).

Uvedené lze shrnout takto: Hledáme veličiny, které se pro různá prostředí, kterými automobil projíždí, samy shlukují a přitom jsou na sobě pokud možno nezávislé. Protože možností je příliš na jejich úplný výčet, uvedeme zde jen příklady.

Rychlost

Rychlost je pro shlukování asi nejlepší. Je jiná ve městě, na okresní silnici a dálnici. Přitom intervaly rychlosti v jednotlivých oblastech jsou poměrně stabilní. Nicméně rychlost je velmi přesně úměrná ujeté dráze. Proto nemá cenu tyto dvě veličiny vybrat společně.

Plyn

Stlačení plynu souvisí s akcelerací automobilu. Lze asi počítat s tím, že na dálnici nebude plyn stlačen příliš a jeho změny budou minimální. Na okresních silnicích se plyn bude měnit více, ale největší změny budou asi ve městě. To ale platí jen velmi zhruba - při předjíždění na dálnici může být plyn stlačen více a při opatrné jízdě na okresní silnici se plyn nemusí příliš měnit. Vhodnost plynu pro detekci oblastí není příliš velká. Navíc, stlačení plynu téměř deterministicky souvisí s momentem motoru.

Brzda

Brzda je problematická veličina. Při správné jízdě se správný řidič brzdy prakticky nedotkne - a když, tak jen na úplné zastavení. Sebraná data tento fakt potvrzují. Brzda tedy vypovídá více méně jen o zastavení vozidla, případně výjimečně o situacích, které řidič “na plynu” špatně odhadl a musel si přibrzdit. Ve městech bude těchto situací více, ale o shlukování lze mluvit.

Kvalt

Kvalt je stejně problematický, jako brzda. Silnější auta, jako je Škoda Octavia, na které byla data měřena, jede prakticky stále na stupeň 5 nebo 6. Takže z kvaltu se toho zase moc nepozná.

Otáčky

zůstává). Dynamická směs (tedy dynamické ukazovátko) říká, že přechody mezi komponentami mají určitý řád (např. jedu-li po dálnici, tak je určitá pravděpodobnost, že budu pokračovat po dálnici nebo, že sjedu na okresní silnici nebo sjedu ve městě).

Součin kvaltu a otáček motoru vypovídá o rychlosti a tím i o ujeté dráze. Jinak z nich asi nic moc dalšího nezjistíme.

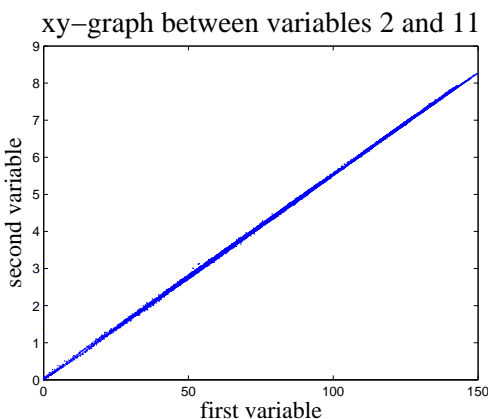
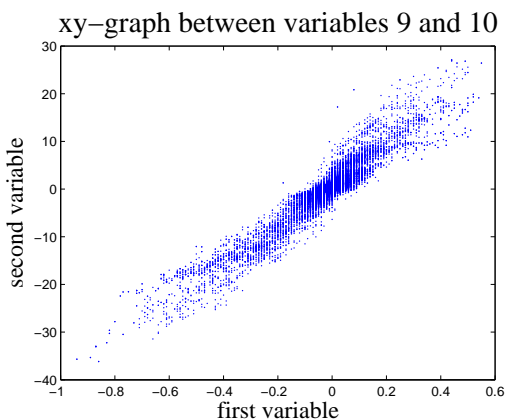
Příčné zrychlení a Stáčivá rychlost

Tyto dvě veličiny spolu úzce souvisí. Proto stačí pracovat jen s jednou z nich. Obě vypovídají o zatáčení automobilu (případně kombinovaní s rychlostí). Zatáčení bude jistě jedna z veličin, které dobře vypovídají o prostředí, kterým automobil projíždí. Na dálnice se jede přímo, okresní silnice zatáčí ale jistě ne do pravého úhlu a městské silnice zatáčí často a hodně. Tyto veličiny souvisí se směrem jízdy, i když ne přímo, ale spíše se změnami směru.

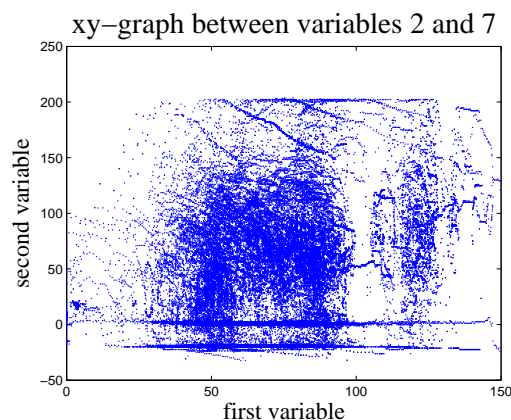
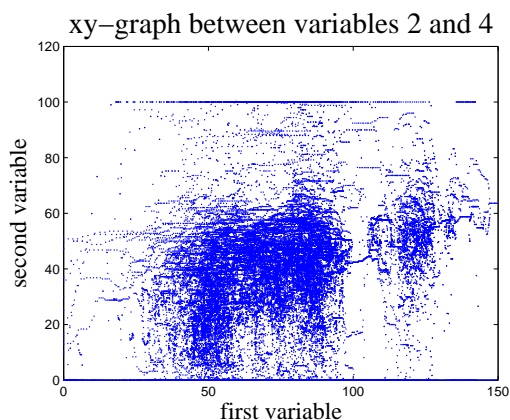
Nadmořská výška

Změna nadmořské výšky vypovídá o stoupání nebo klesání dráhy automobilu. Je sice pravda, že kopce asi nebudou na dálnici, mohou být na okresních silnicích ale mohou být, a to i výrazně, v některých městech. Proto asi tato veličina sama o sobě nebude pro nás příliš významná.

Souvislost dvou veličin je nejlépe sledovat v xy-grafu. Některé si dále ukážeme:



Vlevo je souvislost mezi dostředivým zrychlením a stáčivou rychlostí. Vpravo jsou proti sobě vyneseny veličiny rychlosti a ujetá dráha. V obou případech vidíme, že mezi veličinami existuje přesnější nebo volnější přímá úměra, tedy že jednu z veličin lze dobře nahradit druhou.



Tady jsou vyneseny dvojice rychlost - plyn a rychlost - moment. Vidíme, že informace kterou nesou je velmi podobná.

Nicméně dvě veličiny je největší množství, které lze takto zkoumat. Při větším počtu veličin již vizuální výsledky nedostaneme a musíme se spoléhat na číselné charakteristiky, které si za tím účelem definuje. Úplné zhodnocení modelu přináší ale nakonec až úspěch nebo neúspěch samotné aplikace, pro kterou jsme model vytvářeli. Tento test by se rozhodně neměl provádět až před obecenstvím, ale velmi důkladně a hlavně mnohem dříve!

V našem případě do datového vektoru vybereme veličiny: [2 4 7 8 16] to je [rychlost, plyn, moment, otáčky, směr]. Odhad modelu provedeme standardním způsobem pro výběr s 5000 daty; pro inicializaci využijeme 2000 dat mimo datový výběr. Původní data jsou vzorkována 5 krát za vteřinu, tedy s periodou 200 msec. To je pro naše účely příliš husté (z hlediska změny prostředí se skoro nic neděje), proto data 10 krát zředíme (vezmeme každé desáté měření) a dostaneme periodu vzorkování 2 sec¹².

Pro účely testování je vhodné pracovat s veličinami, které nabývají podobně velkých hodnot. Proto vybrané veličiny standardním způsobem normujeme - odečteme jejich průměry a dělíme směrodatnými hodnotami. Tak dostaneme veličiny s nulovou střední hodnotou a jednotkovým rozptylem. Ovšem vztahy mezi nimi, o které nám jde, zůstávají stejné.

Odhadem jsme získali následující parametry (statistiky) modelu

Centra komponent

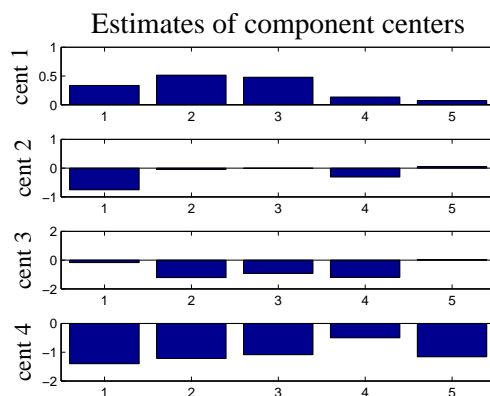
¹²To sice také není žádná sláva, ale při dalším ředění bychom už narazili na nedostatek dat. Původní celý datový soubor je dlouhý 75000 vzorků a měřící automobil ujel asi 320 km. Po deseti násobném zředění máme 7500 dat. Z nich bereme 5000 dat pro odhad a 2000 dat na inicializaci. A jsme na dorazu.

Při tom v naší periodě 2 sec a průměrné rychlosti 90 km/k řidič ujede 50 metrů. Tedy např. na 10 kilometrové cestě po dálnici dostaneme 200 vzorků z jedné oblasti, tj. těmito daty je "živena" jediná komponenta, zatímco ostatní jsou vlastně bez dat.

Odtud je vidět (i) že by bylo dobré periodu vzorkování ještě prodloužit, (ii) jak složitá je situace při odhadu směsi, (iii) že ani 75000 měření nemusí být pro odhad směsi dostatečné množství dat.

$$\begin{aligned}\theta_1 &= [0.3352, 0.5145, 0.4785, 0.1337, 0.0722]', \\ \theta_2 &= [-0.7556, -0.0441, -0.0092, -0.3127, 0.0511]', \\ \theta_3 &= [-0.1727, -1.2152, -0.9208, -1.1971, 0.0305]', \\ \theta_4 &= [-1.4011, -1.2193, -1.0858, -0.4981, -1.1610]'. \end{aligned}$$

Graficky je můžeme znázornit takto



Vidíme, že centra jsou rozmístěna v datovém prostoru rozumně - jednotlivé komponenty mají různá centra a tak (doufejme) pokrývají různé shluky.

Vzájemné uspořádání komponent lze rovněž vyjádřit maticí vzájemných Kullback-Leiblerových “vzdáleností” komponent, které kromě vzájemné polohy středů berou v úvahu i shodu / neshodu kovariančních matic¹³.

Tato matice je

$$\begin{bmatrix} 0 & 7.0 & 4141.8 & 525.7 \\ x & 0 & 2381.7 & 345.7 \\ x & x & 0 & 9.8 \\ x & x & x & 0 \end{bmatrix},$$

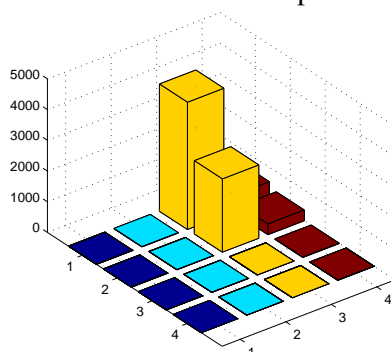
kde svislý i vodorovný index označuje komponentu a čísla jsou vzdálenosti. Na diagonále jsou nuly - stejné komponenty mají nulovou vzdálenost a v dolní trojúhelníku jsou opačné vzdálenosti (ty jsou jiné, ale my budeme brát jen ty v horním trojúhelníku - dolní budou jiné, ale podobné). Nám jde především o to, jestli se některé dvě komponenty “nesjely” na stejný shluk. Je vidět, že to se nestalo. Grafické zobrazení je následující

¹³Kullback-Leiblerova vzdálenost dvou normálních rozdělení je

$$\frac{1}{2} \left[\text{tr}(r_2^{-1}r_1) + (\mu_2 - \mu_1)' r_1^{-1} (\mu_2 - \mu_1) - n - \ln \frac{\det r_1}{\det r_2} \right],$$

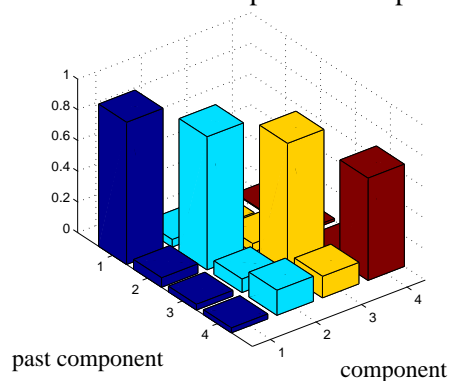
kde μ_1, μ_2 jsou střední hodnoty, r_1, r_2 jsou kovariance a n je rozměr náhodné veličiny, kterou hustoty popisují.

KL distences of components

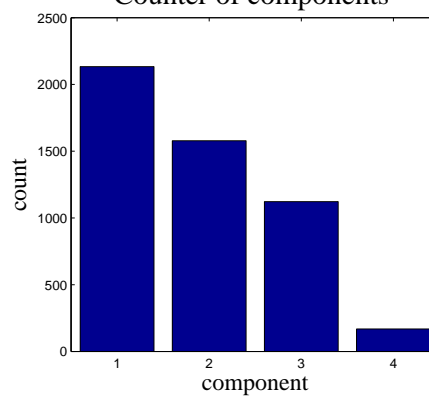


Dále nás bude zajímat obsazení komponent daty (tj. statistika κ modelu) a přechody mezi komponentami (parametr α ukazovátka). Ty jsou na následujících obrázcích

Estimate of the parameter alpha

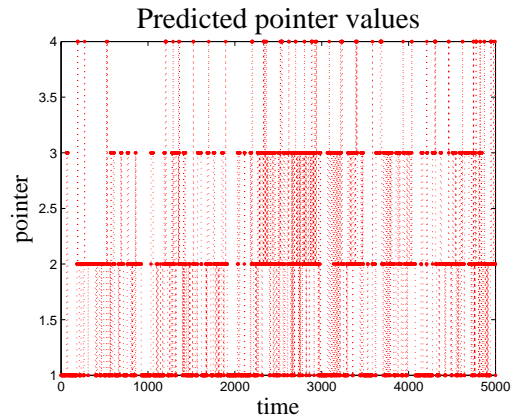


Counter of components



Vlevo jsou vyznačeny prvky parametru α . Vidíme, že velké jsou především diagonální prvky, což svědčí o tom, že pravidelnost v přechodech není velká. Vpravo je obsazení komponent. Poslední čtvrtá komponenta má v sobě poměrně málo dat, jinak je obsazenost dobrá. Tento poslední obrázek, zejména parametr α nedává pro predikci mnoho nadějí.

Průběh ukazovátka je následující



Vidíme, že oblasti se za jízdy auta mění - poslední komponenta je navštěvována méně než ostatní. Nicméně přeskoky mezi komponentami se zdají rozumné. Každopádně, o tom, jestli přeskoky odpovídají přejezdům z konkrétní oblasti do konkrétní oblasti bychom se přesvědčili až při skutečné jízdě automobilem, průběžném odhadu a porovnání odhadovaných a skutečných oblastí.

A to hovoříme o odhadu. Predikce, která se opírá a parametr α bude, vzhledem k hodnotám parametru, velmi špatná.

Program, který predikci realizuje je následující

```

clc , clear all , close all
addpath 'funct '
seed(51,187)
% Estimation and prediction of mixture with dynamic pointer
% - real multi-dimensional data (controlled car)
% - estimation + prediction
%
% Calls: iniMix , Gauss2 , v2thN , c2c

nc=4; % number of components
np=5; % length of prediction
nd=5000; % length of data
ni=2000; % length of initial data
ch=[1 2 4 7 8 16];
ch=[2 4 7 8 10]; % involved variables
% 1 actual car consumption
% 2 actual car speed
% 3 driving wheel
% 4 gas pedal
% 5 brake
% 6 gear of transmittion
% 7 engine moment

```

```

% 8 engine revs
% 9 cross acceleration
% 10 speed of rotation
% 11 distance/period
% 15 altitude
% 16 direction of the way
ny=length(ch);

load DatAuto dt          % data load
dt=scal(dt);            % scaling of data

dt(15,:)= [0 diff(dt(15,:))]; % altitude -> slope
dt(17,:)= [0 diff(dt(2,:))]; % speed -> acceleration
dt(18,:)= [0 diff(dt(4,:))]; % gas -> gas increment
dtt=dt(:,1:10:end);
y=dtt(ch,1:nd);          % output
yIni=dtt(ch,(nd+1):(nd+ni)); % initial output
ph=length(yIni); phc=10;

% INITIALIZATION
if 1
    typ=1;                % dynamic mixture
    [th co al V ka nu]=iniMix(nc,yIni,typ,ph,phc);
else
    th{1}=[50;50]; th{2}=[50;100]; th{3}=[130;130];
    th{4}=[100;0]; th{5}=[90;120]; th{6}=[120;100];
    th{7}=[90;120]; th{8}=[120;100];
    [th co al V ka nu]=iniMix2(nc,yIni,th); ka=ka*.1;
end

w=zeros(nc,nd);
w(:,1)=ones(nc,1)/nc;    % vector weights

% ESTIMATION
fprintf('running ..... | \ n. '),itime=0;
for t=2:nd                % TIME LOOP -----
    itime=itime+1;
    if itime>(nd-1)/20, fprintf('.'), itime=0; end

    yt=y(:,t);            % output measuring

    % distances of yt from components
    for c=1:nc
        [xxx p(c)]=GaussN(yt,th{c},co{c}); % value of component
    end
end

```

```

end
p=p-max(p); p=exp(p); % log -> exp
if p'*p<1e-18, p=ones(size(p)); end % patch for all 0
if sum(isinf(p))>0,
    p=zeros(size(p));
    iI=find(isinf(p));
    p(iI)=1;
end
wm=(w(:,t-1)*p).*al; % matrix weights
wm=wm/sum(sum(wm)); % normalization of wm
w(:,t)=sum(wm); % new vector weights

% prediction of components
wp(:,t)=(al^np)'*w(:,t);

% update of statistics
Ps=[yt;1]; % regression vector
for c=1:nc
    V{c}=V{c}+w(c,t)*Ps*Ps'; % info. matrix
    ka(c)=ka(c)+w(c,t); % counter
    [th{c} co{c}]=v2thN(V{c}/ka(c),ny); % point estimates
end

nu=nu+wm; % Dirichlet stats
al=fnorm(nu,2); % point estimate
end

% RESULTS
s=nd-50+1:nd; % range fpr plot
[xxx ct]=max(w); % estimated active components
[xxx cp]=max(wp); % predicted active components

aMixDyPreRe_R

```

9 Přílohy

9.1 Speciální funkce

Kroneckerova funkce

Kroneckerova funkce $\delta(x, x_0)$ je posloupnost hodnot, které jsou všechny nulové až na hodnotu s indexem x_0 , která je rovna jedné.

Diracova funkce

Diracova funkce $\delta(x, x_0)$ vychází z teorie zobecněných funkcí. Je definována jako operátor, který z funkce $g(x)$ vybere právě hodnotu funkce g v bodě x_0 , tedy

$$\delta(x, x_0) * g(x) = g(x_0).$$

Vyjádříme-li tento operátor v integrálním tvaru, dostaneme

$$\int_{-\infty}^{\infty} g(x) \delta(x, x_0) dx = g(x_0). \quad (9.1)$$

Proto lze na operátor δ pohlížet jako na funkci, která je všude nulová kromě bodu x_0 . Zde má hodnotu nekonečno, ovšem tak, aby platil vztah (9.1).

Funkci $\delta(x, x_0)$ budeme považovat za limitní případ hp náhodné veličiny x jejíž hodnota je x_0 a tato náhodná veličina je prakticky deterministická. Její střední hodnota ne x_0 a rozptyl nula.

Gama funkce

Gama funkce je definována takto

$$\Gamma(x) = \int_0^{\infty} t^{x-1} \exp\{-t\} dt, \quad x \geq 0.$$

Pro $n = 1, 2, \dots$ platí

$$\Gamma(n) = (n-1)!$$

Dále platí důležitý vzorec

$$\Gamma(x+1) = x\Gamma(x).$$

Beta funkce

Definice beta funkce je

$$B(x, y) = \int_0^1 t^{x-1} (1-t)^{y-1} dt, \quad x, y > 0,$$

nebo pomocí gama funkce

$$B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

Pro více argumentů $x = [x_1, x_2, \dots, x_n]$ je definována **mnohorozměrná beta funkce** předpisem

$$B(x) = \frac{\prod_{i \in i^*} \Gamma(x_i)}{\Gamma(\sum_{i \in i^*} x_i)}$$

Kromě obyčejné beta funkce zavedeme ještě tzv. **zobecněnou beta funkci** pro tabulkový argument $\gamma_{i|j}$ kde např. γ je dáno tabulkou

	$i = 1$	$i = 2$	$i = 3$
$j = 1$	$\gamma_{1 1}$	$\gamma_{2 1}$	$\gamma_{3 1}$
$j = 2$	$\gamma_{1 2}$	$\gamma_{2 2}$	$\gamma_{3 2}$
$j = 3$	$\gamma_{1 3}$	$\gamma_{2 3}$	$\gamma_{3 3}$

Všimněme si, že pokud matici definovanou tabulkou označíme jako a , pak platí $\gamma = a'$. Zobecněná beta funkce je definována takto

$$B(x) = \prod_{j \in j^*} \frac{\prod_{i \in i^*} \Gamma(\gamma_{i|j})}{\Gamma(\sum_{k \in i^*} \gamma_{k|j})}.$$

9.2 Doplnění na čtverec

Používá se pro minimalizaci kvadratického výrazu nebo pro integraci v konvoluci dvou normálních rozdělání (tady má význam rozkladu normální sdružené hp na podmíněnou a marginální).

Skalární případ

Pro skalární veličiny x a y a konstanty a, b, c platí

$$ax^2 + 2bxy + cy^2 = a \left[x^2 + 2x \frac{b}{a} y + \left(\frac{b}{a} y \right)^2 - \left(\frac{b}{a} y \right)^2 \right] + cy^2 =$$

$$a \left(x + \frac{b}{a} y \right)^2 + cy^2 - \frac{b^2}{a} y^2 = a \left(x + \frac{b}{a} y \right)^2 + \frac{ac - b^2}{a} y^2.$$

Vektorový případ

Pro veličiny x a y ve formě sloupcových vektorů a konstantní matice A, B, C odpovídajících rozměrů, A a C symetrické, platí

$$x'Ax + 2x'By + y'Cy = x'Ax + 2x'AA^{-1}By + (A^{-1}By)'AA^{-1}By - (A^{-1}By)'AA^{-1}By + y'Cy =$$

$$= \underbrace{(x + A^{-1}By)'A(x + A^{-1}By)}_{\text{kvadrát}} + \underbrace{y'(C - B'A^{-1}B)y}_{\text{zbytek}}.$$

Poznámka

Ve vektorovém případě výrazu ax^2 odpovídá $x'Ax$. V tomto smyslu se dějí i úpravy (konstanta - matice) musí být uprostřed.

Výrazy lze ověřit roznásobením konce a porovnáním se začátkem.

9.3 Přirozené podmínky řízení

Přirozené podmínky řízení (Natural Conditions of Control, NCC) jsou podmínky nutné pro to, aby bylo možno konzistentně provést odhadování s modelem, který v sobě obsahuje část s řídicí veličinou, tedy modelem popsaným hp jako např. $f(y_t|u_t, d(t-1), \Theta)$ nebo $f(x_{t+1}|x_t, u_t)$, kde u_t je řízení a Θ nebo x_t jsou odhadované veličiny. Tyto podmínky lze odvodit z předpokladu, že ten kdo odhaduje, je také ten, kdo řídí. Přitom odhad i řízení je počítáno pouze z informace, která je v čase t obsažena v minulých datech $d(t-1)$. Odtud plyne, že jak odhad, tak i řízení v sobě neobsahují další informaci než tu, kterou nesou minulá data $d(t-1)$. Proto např. pro odhad parametrů $f(\Theta|u_t, d(t-1))$ platí, že veškerá dostupná informace o parametru Θ je již obsažena v datech $d(t-1)$ a v řízení u_t již další informace není. Proto je možno jej z podmínky vypustit

$$f(\Theta|u_t, d(t-1)) = f(\Theta|d(t-1)). \quad (9.2)$$

Obrácený vztah lze odvodit obdobnou úvahou, nebo jej lze pomocí Bayesova vzorce odvodit z (9.2)

$$\begin{aligned} f(u_t|d(t-1), \Theta) &= \underbrace{f(\Theta|u_t, d(t-1)) \frac{f(u_t|d(t-1))}{f(\Theta|d(t-1))}}_{\text{Bayes}} = \\ &= \underbrace{f(\Theta|d(t-1)) \frac{f(u_t|d(t-1))}{f(\Theta|d(t-1))}}_{\text{podle (9.2)}} = f(u_t|d(t-1)). \end{aligned}$$

Podobné úvahy lze místo o parametru vést i o odhadovaném stavu.

9.4 Bayesův vzorec**Odvození Bayesova vzorce**

Odvození je velice jednoduché. Uvažujme tři náhodné veličiny A , B a C a sdruženou hp pro A , B podmíněnou C .

$$f(A, B|C) = \begin{cases} f(A|B, C) f(B|C) & \text{z jedné strany, nebo} \\ f(B|A, C) f(A|C) & \text{z druhé strany.} \end{cases}$$

Porovnáním obou výrazů na pravé straně dostaneme

$$f(A|B, C) f(B|C) = f(B|A, C) f(A|C).$$

Z této rovnosti pak lze vyjádřit buď $f(A|B, C)$ nebo $f(B|A, C)$ podle potřeby, a tak získáme Bayesův vzorec

$$f(B|A, C) = \frac{f(A|B, C) f(B|C)}{f(A|C)}. \quad (9.3)$$

Hlavní význam Bayesova vzorce je v tom, že přepočítává apriorní hp $f(B|C)$ z čitatele na pravé straně vzorce na aposteriorní hp $f(B|A, C)$ na levé straně. Apriorní hp popisuje náhodnou veličinu B jen v závislosti na náhodné veličině C , zatímco aposteriorní hp využívá informaci také z náhodné veličiny A , a to prostřednictvím hp $f(A|B, C)$.

Hp ve jmenovateli pravé strany výrazu (9.3) nezávisí na B a je tedy jen normalizační konstantou, kterou lze získat integrací (sumací) čitatele

$$f(A|C) = \int_{B^*} f(A|B, C) f(B|C) dB,$$

což, jak dobře víme, odpovídá vzorci pro úplnou pravděpodobnost.

Aplikace Bayesova vzorce

Pro účely odhadu neznámých parametrů modelu $f(y_t|\psi_t, \Theta)$ zvolíme

- A je výstup soustavy y_t ,
- B jsou odhadované parametry Θ a
- C jsou stará data $d(t-1)$ (případně s řízením u_t).

Dostáváme Bayesův vzorec

$$f(\Theta|d(t)) = \frac{f(y_t|\psi_t, \Theta) f(\Theta|d(t-1))}{f(y_t|d(t-1))}$$

podle (??) s tím, že

– stará data $d(t-1)$ v modelu soustavy v podmínce lze nahradit regresním vektorem ψ_t ,

– v případě řízené soustavy, když data jsou $d_t = \{y_t, u_t\}$, je třeba předpokládat **přirozené podmínky řízení** (9.2), při kterých platí

$$f(\Theta|u_t, d(t-1)) = f(\Theta|d(t-1)),$$

tedy Θ a u_t jsou podmíněně nezávislé, jestliže známe stará data $d(t-1)$.

9.5 Multinomiální rozdělení

Multinomiální rozdělení popisuje diskrétní náhodnou veličinu, tj. veličinu, která může nabývat jen konečného počtu hodnot $y \in \{1, 2, \dots, n_l\}$ a jejíž jednotlivé hodnoty nastávají s pevnými pravděpodobnostmi. Speciálním případem tohoto rozdělení pro $n_l = 2$, jehož hodnoty jsou ale většinou označovány 0 a 1, je rozdělení alternativní.

Hustotu pravděpodobnosti multinomiálního rozdělení je možno vyjádřit ve formě tabulky

$$\frac{y}{f(y)} \left| \begin{array}{cccc} 1 & 2 & \cdots & n_l \\ p_1 & p_2 & \cdots & p_{n_l} \end{array} \right.,$$

kde p_i jsou pravděpodobnosti, a tak platí $p_i \geq 0$, $i = 1, 2, \dots, n_l$ a $\sum_{i=1}^{n_l} p_i = 1$.

Jiné možné vyjádření multinomiálního rozdělení je

$$f(y) = p_y, \quad y = 1, 2, \dots, n_l.$$

Model diskrétního systému je podmíněná hp, která pro každou konfiguraci hodnot veličin v podmínce popisuje modelovanou veličinu pomocí multinomiálního rozdělení

$$f(y|\psi, \Theta) = \Theta_{y|\psi}.$$

Tuto hp můžeme vyjádřit ve formě tabulky např. pro $y \in \{1, 2\}$ a $\psi = [u, v]'$, kde $u, v \in \{1, 2\}$

$$f(y|u, v)$$

$[u, v]$	$y = 1$	$y = 2$
1, 1	$\Theta_{1 11}$	$\Theta_{2 11}$
1, 2	$\Theta_{1 12}$	$\Theta_{2 12}$
2, 1	$\Theta_{1 21}$	$\Theta_{2 21}$
2, 2	$\Theta_{1 22}$	$\Theta_{2 22}$

kde $\Theta_{i|jk}$ jsou podmíněné pravděpodobnosti (proto je také jejich index rozdělen svislítkem). Proto musí platit nezápornost $\Theta_{i|jk} \geq 0$, $\forall i, j, k$ a dále součet parametrů pro každou konfiguraci podmínky musí dát jedničku $\sum_{i=1}^2 \Theta_{i|jk} = 1$, $\forall j, k$.¹⁴ Pro účely odhadu je výhodné formálně tento model vyjádřit v tzv. **součinném tvaru**

$$f(y|\psi, \Theta) = \prod_{i \in y^*} \prod_{\varphi \in \psi^*} \Theta_{i|\varphi}^{\delta(i|\varphi, y|\psi)}, \quad (9.4)$$

¹⁴Tento požadavek je po úvaze opět zřejmý. Jestliže nastalo $u = j$ a $v = k$, má y právě dvě možnosti: $y = 1$ nebo $y = 2$.

kde i je index, φ je multiindex (vektorový index), y^* , ψ^* označují množiny hodnot (čísel nebo vektorů) příslušných veličin a $\delta(i|\varphi, y|\psi)$ je Diracův impulz, tj. rovná se jedné pro $i|\varphi = y|\psi$ a jinak je nula. Přepis do součinnového tvaru je skutečně formální, protože cokoli na nulu je jedna a po vynásobení zůstane jen $\Theta_{y|\psi}$.

9.6 Dirichletovo rozdělení

Konjugovaným rozdělením¹⁵ k multinomiálnímu je rozdělení Dirichletovo

$$f(\Theta|d(t)) = \frac{1}{B(\nu_t)} \prod_{i \in y^*} \prod_{\varphi \in \psi^*} \Theta_{i|\varphi}^{\nu_{i|\varphi;t}}, \quad (9.5)$$

kde

ν_t je statistika rozdělení se stejnou strukturou jako má parametr Θ - viz tabulka v odstavci 9.5 a další značení také odpovídá značení zavedenému v tomto odstavci,

$B(\nu)$ je zobecněná beta funkce

$$B(\nu) = \prod_{\varphi \in \psi^*} \frac{\prod_{i \in y^*} \Gamma(\nu_{i|\varphi})}{\Gamma\left(\sum_{i \in y^*} \nu_{i|\varphi}\right)}, \quad (9.6)$$

kde $\Gamma(\cdot)$ je gama funkce definovaná vztahem

$$\Gamma(x) = \int_0^\infty t^{x-1} \exp(-t) dt, \quad (9.7)$$

pro kterou platí

$$\Gamma(x+1) = x\Gamma(x), \quad x \in R^+. \quad (9.8)$$

9.7 Normální rozdělení

Uvažujme normální regresní model s regresní vektorem ψ_t , koeficienty θ a rozptylem šumu r , kde značíme $\Theta = \{\theta, r\}$. Jeho rovnice je

$$y_t = \psi_t' \theta + e_t, \quad e_t \sim N(0, r).$$

Podmíněná hp tohoto modelu má tvar

$$f(y_t|\psi_t, \Theta) = \frac{1}{\sqrt{2\pi}} r^{-0.5} \exp\left\{-\frac{1}{2r} (y_t - \psi_t' \theta)^2\right\}. \quad (9.9)$$

¹⁵Konjugované rozdělení je takové rozdělení apriorní hp parametrů, které s daným rozdělením, použitým pro model soustavy, produkuje v Bayesově vzorci aposteriorní rozdělení, které si zachovává stejný tvar. Tedy nedochází k tomu, že se tvar aposteriorní hp v každém čase odhadování stává stále složitější, až je nakonec aposteriorní hp pro výpočty nepoužitelná.

Střední hodnota modelu je

$$E[y_t|\psi_t, \Theta] = \psi_t' \theta,$$

rozptyl je

$$D[y_t|\psi_t, \Theta] = r.$$

Pro účely odhadování je výhodné exponent modelu (9.9) ještě upravit. Budeme postupovat následovně:

- exponent rozdělíme tak, abychom dostali součin dvou členů, z nichž jeden bude obsahovat jen data a druhý jen parametry

$$y_t - \psi_t' \theta = -[-1 \theta'] \begin{bmatrix} y_t \\ \psi_t \end{bmatrix} = -[y_t \ \psi_t] \begin{bmatrix} -1 \\ \theta \end{bmatrix}$$

(minus je vytknuto formálně a pod kvadrátem se ztratí).

- kvadrát napíšeme jako součin a dosadíme předchozí výrazy

$$\begin{aligned} (y_t - \psi_t' \theta)^2 &= (y_t - \psi_t' \theta) (y_t - \psi_t' \theta) = \\ &= [-1 \theta'] \begin{bmatrix} y_t \\ \psi_t \end{bmatrix} [y_t \ \psi_t] \begin{bmatrix} -1 \\ \theta \end{bmatrix} = [-1 \theta'] D_t \begin{bmatrix} -1 \\ \theta \end{bmatrix}, \end{aligned}$$

kde $D_t = \begin{bmatrix} y_t \\ \psi_t \end{bmatrix} [y_t \ \psi_t]$ je tzv. datová matice.

Model (9.9) je s uvedenou úpravou možno zapsat takto

$$f(y_t|\psi_t, \Theta) = \frac{1}{\sqrt{2\pi}} r^{-0.5} \exp \left\{ -\frac{1}{2r} [-1 \theta'] D_t \begin{bmatrix} -1 \\ \theta \end{bmatrix} \right\}. \quad (9.10)$$

9.8 Inverzní Gauss-Wishartovo (GiW) rozdělení

Toto rozdělení vzniká jako rozdělení součinu normálních rozdělení (tj. jako rozdělení likelihoodu pro normální model). Má tvar

$$f(\Theta|d(t)) \propto r^{-0.5\kappa_t} \exp \left\{ -\frac{1}{2r} [-1 \theta'] V_t \begin{bmatrix} -1 \\ \theta \end{bmatrix} \right\}, \quad (9.11)$$

kde κ_t a V_t jsou statistiky rozdělení (κ_t se někdy nazývá počítadlo, protože uchovává počet dosud zpracovaných datových vektorů a matice V_t se nazývá rozšířená informační matice).

Matice V_t je symetrická a pozitivně definitní a často se rozkládá

- na submatice

$$V_t = \begin{bmatrix} V_y & V'_{y\psi} \\ V_{y\psi} & V_\psi \end{bmatrix}, \quad (9.12)$$

kde V_y je číslo, $V_{y\psi}$ je sloupcový vektor a V_ψ je čtvercová matice stupně o jeden menší než je V_t ,

- na faktory

$$V_t = L' D L, \quad (9.13)$$

kde L je dolní trojúhelníková matice s jedničkami na diagonále a D je diagonální matice s nezápornými prvky na diagonále. Tento rozklad se nazývá LD-rozklad a pro symetrickou pozitivně definitní matici je jednoznačný. Matice L a D se potom rozkládají na submatice

$$L = \begin{bmatrix} 1 & 0 \\ L_{y\psi} & L_\psi \end{bmatrix}, \quad D = \begin{bmatrix} D_y & 0 \\ 0 & D_\psi \end{bmatrix},$$

kde $L_{y\psi}$ je sloupcový vektor, L_ψ dolní trojúhelníková matice s jedničkami na diagonále, D_y je nezáporné číslo a D_ψ je diagonální matice s nezápornými prvky na diagonále.

Uvedené rozklady se dále využijí pro vyjádření potřebných charakteristik rozdělení.

9.9 Bodový odhad podle kvadratického kritéria

Nejprve ukážeme obecně, že bodový odhad optimální podle kvadratického kritéria je roven podmíněné střední hodnotě. Budeme odvozovat odhad $\hat{\Theta}_t$ parametru Θ s a posteriorní hp $f(\Theta|d(t))$, který je optimální podle kritéria

$$\min_{\hat{\Theta}_t} E \left[(\Theta - \hat{\Theta}_t)^2 | d(t) \right]. \quad (9.14)$$

Kritérium umocníme, aplikujeme střední hodnotu a doplníme na čtverec v proměnné $\hat{\Theta}_t$. Dostáváme

$$\begin{aligned} & \min_{\hat{\Theta}_t} E \left[\Theta^2 - 2\hat{\Theta}_t\Theta + \hat{\Theta}_t^2 | d(t) \right] = \\ & = \min_{\hat{\Theta}_t} \left\{ E \left[\Theta^2 | d(t) \right] - 2\hat{\Theta}_t E \left[\Theta | d(t) \right] + \hat{\Theta}_t^2 \right\} = *1* \end{aligned}$$

využili jsme skutečnost, že $\hat{\Theta}_t$ je číslo

$$*1* = \min_{\hat{\Theta}_t} \left\{ E \left[\Theta^2 | d(t) \right] - E \left[\Theta | d(t) \right]^2 + E \left[\Theta | d(t) \right]^2 - 2\hat{\Theta}_t E \left[\Theta | d(t) \right] + \hat{\Theta}_t^2 \right\} = *2*$$

použili jsme výpočetní vzorec pro rozptyl $D[\Theta] = E[\Theta^2] - E[\Theta]^2$

$$*2* = \min_{\hat{\Theta}_t} \left\{ D[\Theta|d(t)] + \left(\hat{\Theta}_t - E[\Theta|d(t)] \right)^2 \right\} = D[\Theta|d(t)]$$

což dá optimální odhad

$$\hat{\Theta}_t = E[\Theta|d(t)].$$

9.10 Bodové odhady parametrů spojitého modelu

Ukážeme odvození MAP (Maximum A posteriori Probability) bodového odhadu který maximalizuje aposteriorní hp parametrů.

Hledáme tedy maximum aposteriorní hp (9.11)

$$\begin{aligned} f(\Theta|d(t)) &\propto r^{-0.5\kappa} \exp \left\{ -\frac{1}{2r} [-1 \ \theta'] V \begin{bmatrix} -1 \\ \theta \end{bmatrix} \right\} = \\ &= r^{-0.5\kappa} \exp \left\{ -\frac{1}{2r} (V_y - 2\theta' V_{y\psi} + \theta' V_\psi \theta) \right\}, \end{aligned}$$

kde jsme využili dělení matice V podle (9.12).

Nejdříve budeme hledat maximum podle θ , tj. derivovat podle θ a hledat řešení pro derivaci rovnu nule.¹⁶

$$\frac{\partial f(\{\theta, r\} | d(t), r)}{\partial \theta} \propto r^{-0.5\kappa} \exp \left\{ -\frac{1}{2r} [-1 \ \theta'] V \begin{bmatrix} -1 \\ \theta \end{bmatrix} \right\} \left(\frac{-1}{2r} \right) (-2V_{y\psi} + 2V_\psi \theta) = 0.$$

Odtud pochází vztah

$$\hat{\theta} = V_\psi^{-1} V_{y\psi}. \quad (9.15)$$

Výsledek dosadíme do aposteriorní hp. V exponentu obdržíme zbytek po minimalizaci Λ

$$\begin{aligned} \Lambda &= V_y - 2\hat{\theta}' V_{y\psi} + \hat{\theta}' V_\psi \hat{\theta} = \\ &= V_y - 2V_{y\psi}' V_\psi^{-1} V_{y\psi} + V_{y\psi}' V_\psi^{-1} V_\psi V_\psi^{-1} V_{y\psi}, \end{aligned}$$

a tedy

$$\Lambda = V_y - V_{y\psi}' V_\psi^{-1} V_{y\psi}. \quad (9.16)$$

Aposterioorní hp s dosazeným bodovým odhadem (9.15) je

$$f(r|d(t)) \propto r^{-0.5\kappa} \exp \left\{ -\frac{\Lambda}{2r} \right\}.$$

¹⁶Derivujeme vektory podle vektorů. Správnost lze ověřit derivováním podle složek a zpětným sestavením do vektorového tvaru.

Derivujeme a položíme rovno nule

$$-\kappa \frac{1}{2r} + \Lambda \frac{1}{r^2} = 0,$$

a tedy dostaneme

$$\hat{r} = \frac{\Lambda}{\kappa}. \quad (9.17)$$

$\hat{\theta}$ a \hat{r} jsou bodové odhady, které tady hledáme.

9.11 Bodové odhady parametrů diskrétního modelu

Bodové odhady parametru Θ multinomiálního rozdělení (9.5) dostaneme pouhou normalizací statistiky ν_t tak, aby součty jejích prvků v řádcích jejího maticového vyjádření (jako pro Θ v odstavci (9.5)) byly rovny jedné, tj.

$$\hat{\Theta}_{y|\psi;t} = \frac{\nu_{y|\psi;t}}{\sum_{i \in y^*} \nu_{i|\psi;t}}, \quad \forall y \in y^* \text{ a } \psi \in \psi^*. \quad (9.18)$$

Tento bodový odhad jsme určili jako podmíněnou střední hodnotu parametru s rozdělením podle aposteriorní hp (9.5) - pro přehlednost vynecháme časový index t

$$\begin{aligned} \hat{\Theta}_{y|\psi} &= E[\Theta_{y|\psi}|d(t)] = \int_0^\infty \Theta_{y|\psi} f(\Theta|d(t)) d\Theta = \\ &= \frac{1}{B(\nu)} \int_0^\infty \Theta_{y|\psi} \prod_{i \in y^*} \prod_{\varphi \in \psi^*} \Theta_{i|\varphi}^{\nu_{i|\varphi}} d\Theta = *1*, \end{aligned}$$

kde jsme dosadili za aposteriorní hp z (9.5) a beta funkce B je dána v (9.6). Parametr $\Theta_{y|\varphi;t}$ formálně vyjádříme v součinném tvaru (9.4)

$$\Theta_{y|\psi} = \prod_{i \in y^*} \prod_{\varphi \in \psi^*} \Theta_{i|\varphi}^{\delta(i|\varphi, y|\psi)}$$

a dosadíme. Pokračujeme v úpravě

$$\begin{aligned} *1* &= \frac{1}{B(\nu_t)} \int_0^\infty \prod_{i \in y^*} \prod_{\varphi \in \psi^*} \Theta_{i|\varphi}^{\nu_{i|\varphi} + \delta(i|\varphi, y|\psi)} d\Theta = \\ &= \frac{1}{\prod_{\varphi \in \psi^*} B(\nu_\varphi)} \prod_{\varphi \in \psi^*} \int_0^\infty \prod_{i \in y^*} \Theta_{i|\varphi}^{\nu_{i|\varphi} + \delta(i|\varphi, y|\psi)} d\Theta_{y|\psi} = *2*, \end{aligned}$$

kde

$$B(\nu_\varphi) = \frac{\prod_{i \in y^*} \Gamma(\nu_{i|\varphi})}{\Gamma(\sum_{i \in y^*} \nu_{i|\varphi})} \text{ podle (9.6)}$$

jsme využili předpoklad o nezávislosti parametrů mezi různými konfiguracemi regresního vektoru ψ , tj nezávislosti řádků v tabulkovém vyjádření parametru v odstavci 9.5.

Dále si uvědomíme, že pro jednotlivé komponenty platí

$$\int_0^\infty \prod_{i \in y^*} \Theta_{i|\varphi}^{\nu_{i|\varphi} + \delta(i|\varphi, y|\psi)} d\Theta_{y|\psi} = \begin{cases} B(\nu_\varphi) & \text{pro } \delta = 0, \\ B(\nu_\psi + 1) & \text{pro } \delta = 1. \end{cases}$$

Potom se všechny členy s $\delta = 0$ zkrátí se stejným členem ve funkci B a zůstane jen člen s $\delta = 1$ a odpovídajícím normalizačním členem ve jmenovateli. Pokračujeme ve výpočtu

$$*2* = \frac{B(\nu_\psi + \delta(i, y))}{B(\nu_\psi)} = \frac{\prod_{i \in y^*} \Gamma(\nu_{i|\psi} + \delta(i, y))}{\Gamma(\sum_{i \in y^*} \nu_{i|\psi} + 1)} = *3* .$$

Opět $\delta(i, y)$ je nula všude kromě případu, kdy $y = i$. Všechny ostatní členy se zkrátí. Dostaneme

$$*3* = \frac{\Gamma(\nu_{y|\psi} + 1)}{\Gamma(\sum_{i \in y^*} \nu_{i|\psi} + 1)} = \frac{\nu_{y|\psi}}{\sum_{i \in y^*} \nu_{i|\psi}} = \frac{\Gamma(\nu_{y|\psi})}{\Gamma(\sum_{i \in y^*} \nu_{i|\psi})} = \frac{\nu_{y|\psi}}{\sum_{i \in y^*} \nu_{i|\psi}} .$$

V první úpravě předchozího výrazu jsme využili vlastnosti gama funkce (9.8). Tím jsme dokázali vztah (9.18).

9.12 Bodový odhad likelihoodu

Uvést likelihood ...

Vzorec (3.1) také ilustruje jeden ze základních principů bayesovství. Pokud zacházíme s objektem (tady je to model) a neznáme parametry, musíme tyto parametry odhadnout. Parametry se každopádně odhadují na základě použití Bayesova vzorce (9.3). Výsledkem odhadu je buď aposteriorní hp, která nese maximum informace o odhadu, nebo bodový odhad, který zanedbává neurčitost odhadu, ale zato práce s ním je daleko jednodušší.

Optimální způsob zavedení odhadu do objektu je využití celé aposteriorní hp, jak je to ukázáno v (3.1). Tady nepoužijeme jen jedno číslo jako odhad, ale všechny jeho možné hodnoty a výsledek načítáme (integrujeme) s vahami, které vyjadřují skutečnost, že tato hodnota odhadu je správná (váhy jsou dány aposteriorní hp).

Jednodušší způsob práce s odhady je využití bodových odhadů. Podstatou toho je aproximace aposteriorní hp pomocí Diracovy funkce δ

$$\delta(x; \xi) = \begin{cases} 0 & \forall x \neq \xi \\ \infty & \text{pro } x = \xi \end{cases}$$

a přitom platí

$$\int_{-\infty}^{\infty} \delta(x; \xi) dx = 1, \quad \forall \xi \in R$$

Pro rozumné funkce $g(x)$ platí (a je také definicí)

$$\int_{-\infty}^{\infty} \delta(x; \xi) g(x) dx = f(\xi).$$

Položíme-li tedy

$$f(\Theta|d(t-1)) = \delta(\Theta; \hat{\Theta}_{t-1})$$

kde $\hat{\Theta}_{t-1}$ je bodový odhad spočtený s pomocí $f(\Theta|d(t-1))$, dostaneme¹⁷ z (3.1)

$$f(y_t|u_t, d(t-1)) = \int_{\Theta^*} f(y_t|\psi_t, \Theta) \delta(\Theta; \hat{\Theta}_{t-1}) d\Theta = f(y_t|\psi_t, \hat{\Theta}_{t-1}) \quad (9.19)$$

Postup je tedy následující:

- paralelně s hlavní úlohou provádíme vývoj aposteriorní hp,
- z aposteriorní hp počítáme bodové odhady parametrů,
- bodové odhady parametrů dosadíme za skutečné parametry.

Výsledek jednokrokové predikce s dosazenými bodovými odhady parametrů má dobrou interpretaci:

1. Odhad budoucí hodnoty výstupu
Jak jsme se již zmínili, vzorec (9.19), který vychází z (3.1) dává popis budoucího výstupu ve formě jeho podmíněné hp.
2. Vyhodnocení odhadu parametrů vzhledem k modelu
Model je určen z měřených dat. Parametry určují vztahy mezi těmito veličinami. Hodnota hustoty pravděpodobnosti modelu s dosazenými daty a odhady parametrů vyjadřuje příslušnost parametrů k datům. Čím je větší, tím více parametry odpovídají datům.
3. Posouzení hodnoty výstupu
Chceme-li posoudit, zda měřený výstup odpovídá modelu (který je určen minulými daty a odhady parametrů), dosadíme jej jednoduše do modelu. Čím je výsledná hodnota větší, tím více hodnota výstupu odpovídá modelu.

¹⁷Obecně platí: Máme objekt $O(z)$, závislý na veličině z . Odhad a s aposteriorní hp je střední hodnota tohoto objektu. Tu přibližně nahrazujeme objektem s dosazenou střední hodnotou $z \doteq \hat{z}$

$$\int_{z^*} O(z) f(z|d(t)) dz \doteq O(\hat{z})$$

Tedy zmizí integrál i aposteriorní hp a v objektu se objeví odhad \hat{z} .

9.13 Predikce s normálním modelem

Konstrukce prediktivní hp pro víceřadovou predikci je v obecném případě tak složitá, že je prakticky nespočítatelná. Situaci budeme demonstrovat pro nejjednodušší možný případ - dvouřadová predikce s normálním regresním modelem 1. řádu bez řídicí veličiny.

Máme model

$$y_t = ay_{t-1} + e_t,$$

kde $e_t \sim N(0, r)$ a parametry a, r neznáme. Tento model definuje hp

$$f(y_t|y_{t-1}) = \frac{1}{\sqrt{2\pi r}} \exp\left\{-\frac{1}{2r}(y_t - ay_{t-1})^2\right\}.$$

Chceme určit prediktivní hp

$$f(y_{t+2}|y(t)).$$

Platí

$$f(y_{t+2}|y(t)) = \int_{y_{t+1}^*} f(y_{t+2}|y_{t+1}) f(y_{t+1}|y_t) dy_{t+1} =$$

dosadíme za model

$$= \int_{y_{t+1}^*} \frac{1}{\sqrt{2\pi r}} \exp\left\{-\frac{1}{2r}(y_{t+2} - ay_{t+1})^2\right\} \frac{1}{\sqrt{2\pi r}} \exp\left\{-\frac{1}{2r}(y_{t+1} - ay_t)^2\right\} dy_{t+1}.$$

Budeme integrovat doplněním na čtverec. Doplnění provedeme v exponentu (za vytknutým faktorem $-1/2r$).

$$\begin{aligned} & (y_{t+2} - ay_{t+1})^2 + (y_{t+1} - ay_t)^2 = \\ & = y_{t+2}^2 - 2ay_{t+2}y_{t+1} + a^2y_{t+1}^2 + y_{t+1}^2 - 2ay_{t+1}y_t + a^2y_t^2 = \end{aligned}$$

upravíme na trojčlen v y_{t+1}

$$= (a^2 + 1)y_{t+1}^2 - 2a(y_{t+2} + y_t)y_{t+1} + y_{t+2}^2 + a^2y_t^2 =$$

a doplněním na čtverec v y_{t+1} dostaneme

$$\begin{aligned} & = (a^2 + 1) \left[y_{t+1}^2 - 2\frac{a}{a^2 + 1}(y_{t+2} + y_t)y_{t+1} \right] + y_{t+2}^2 + a^2y_t^2 = \\ & = (a^2 + 1) \left[y_{t+1} - \frac{a}{a^2 + 1}(y_{t+2} + y_t) \right]^2 + \\ & \quad + y_{t+2}^2 + a^2y_t^2 - \frac{a^2}{a^2 + 1}(y_{t+2} + y_t)^2 \end{aligned}$$

první člen integrací zmizí a nás zajímá druhý člen

$$\begin{aligned} & y_{t+2}^2 + a^2 y_t^2 - \frac{a^2}{a^2 + 1} (y_{t+2}^2 + 2y_{t+2}y_t + y_t^2) = \\ &= \frac{(a^2 + 1) y_{t+2}^2 + (a^2 + 1) a^2 y_t^2 - a^2 y_{t+2}^2 - 2a^2 y_{t+2}y_t - a^2 y_t^2}{a^2 + 1} = \\ &= \frac{y_{t+2}^2 + a^4 y_t^2 - 2a^2 y_{t+2}y_t}{a^2 + 1} = \frac{(y_{t+2} + a^2 y_t)^2}{a^2 + 1} \end{aligned}$$

Vrátíme se zpět k hp a dostáváme

$$f(y_{t+2}|y(t)) = \frac{1}{\sqrt{2\pi(a^2 + 1)r}} \exp \left\{ -\frac{1}{2(a^2 + 1)r} (y_{t+2} + a^2 y_t)^2 \right\}.$$

Prediktivní hp bude opět normální, se střední hodnotou $a^2 y_t$ a rozptylem $(a^2 + 1)r$.

9.14 Logistická regrese

Derivace věrohodnostní funkce

Derivace logaritmu věrohodnostní funkce $\ln L$ s modelem (??) podle Θ je

$$\frac{\partial}{\partial \Theta} \ln L(\Theta) = \sum_{\tau=1}^t \left[y_{\tau} \psi_{\tau} - \frac{\exp(z_{\tau})}{1 + \exp(z_{\tau})} \psi_{\tau} \right] = \sum_{\tau=1}^t (y_{\tau} - p_{\tau}) \psi_{\tau},$$

kde podle (??) je $z_{\tau} = \psi_{\tau} \Theta$ a tedy $dz_{\tau}/d\Theta = \psi_{\tau}$. Dále jsme označili

$$p_{\tau} = \frac{\exp(z_{\tau})}{1 + \exp(z_{\tau})} = P(y_t = 1 | \psi_{\tau}, \Theta).$$

Druhá derivace $\ln L$ podle Θ je

$$\frac{\partial^2}{\partial \Theta^2} \ln L(\Theta) = \frac{\partial}{\partial \Theta} \sum_{\tau=1}^t (y_{\tau} - p_{\tau}) \psi_{\tau} = \sum_{\tau=1}^t \frac{\partial}{\partial \Theta} p_{\tau} \psi_{\tau} = \sum_{\tau=1}^t p_{\tau} (1 - p_{\tau}) \psi'_{\tau} \psi_{\tau},$$

protože

$$\begin{aligned} \frac{\partial}{\partial \Theta} p_{\tau} &= \frac{\partial}{\partial \Theta} \frac{\exp(z_{\tau})}{1 + \exp(z_{\tau})} = \frac{\exp(z_{\tau}) \psi'_{\tau} (1 + \exp(z_{\tau})) - \exp(z_{\tau}) \exp(z_{\tau}) \psi'_{\tau}}{(1 + \exp(z_{\tau}))^2} = \\ &= \frac{\exp(z_{\tau}) \psi'_{\tau}}{(1 + \exp(z_{\tau}))^2} = \left(\frac{\exp(z_{\tau})}{1 + \exp(z_{\tau})} \frac{1}{1 + \exp(z_{\tau})} \right) \psi'_{\tau} = p_{\tau} (1 - p_{\tau}) \psi'_{\tau}. \end{aligned}$$

Pro hledání maxima logaritmu věrohodnostní funkce $\ln L$ je výhodné použít Newtonovu metou¹⁸

¹⁸Výhodné je to zejména pro to, že se nám podařilo analyticky spočítat jak první, tak i druhou derivaci maximalizované funkce $\ln L$.

Newtonův algoritmus

Pomocí Newtonova algoritmu je možné numericky hledat extrémy¹⁹ nelineárních funkcí. Označme takovou funkci $g(x)$, kde $x = [x_1, x_2 \cdots x_n]'$. Dále označme gradient této funkce

$$g'(x) = \begin{bmatrix} \frac{\partial g}{\partial x_1} \\ \frac{\partial g}{\partial x_2} \\ \cdots \\ \frac{\partial g}{\partial x_n} \end{bmatrix}$$

a Hessovu matici

$$g''(x) = \begin{bmatrix} \frac{\partial^2 g}{\partial x_1^2} & \frac{\partial^2 g}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 g}{\partial x_1 \partial x_n} \\ \frac{\partial^2 g}{\partial x_2 \partial x_1} & \frac{\partial^2 g}{\partial x_2^2} & \cdots & \frac{\partial^2 g}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 g}{\partial x_n \partial x_1} & \frac{\partial^2 g}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 g}{\partial x_n^2} \end{bmatrix}.$$

Algoritmus začíná hledáním extrému ve zvoleném bodě $x^{(0)}$ a v dalších bodech $x^{(1)}$, $x^{(2)}$, \dots se hledá následujícím způsobem:

provedeme Taylorův rozvoj funkce g v bodě $x^{(i)}$ a vezmeme jeho první tři členy

$$g(x) \doteq g(x^{(i)}) + g'(x^{(i)})(x - x^{(i)}) + \frac{1}{2}g''(x^{(i)})(x - x^{(i)})^2.$$

Následující bod hledání $x^{(i+1)}$ položíme do maxima (minima) rozvoje - do bodu, kde je derivace nulová:

$$g'(x^{(i)}) + g''(x^{(i)})(x^{(i+1)} - x^{(i)}) = 0$$

Z toho plyne:

$$x^{(i+1)} = x^{(i)} - \frac{g'(x^{(i)})}{g''(x^{(i)})}.$$

Iterace provádíme tak dlouho, dokud dva po sobě následující body nejsou dostatečně blízko.

9.15 Generování hodnot posunutého exponenciálního rozdělení

Hodnoty z exponenciálního rozdělení se generují velice snadno pomocí normálního rozdělení přes inverzní distribuční funkci

$$f(y) = a \exp\{-ay\} \rightarrow F(y) = 1 - \exp\{-ay\}$$

¹⁹Pozor! Jde samozřejmě o lokální extrémy.

odtud

$$y = -\frac{1}{a} \ln(1 - F) = -\frac{1}{a} \ln(F)$$

kde F generujeme z rovnoměrného rozdělení (potom i hodnoty $1 - F$ jsou rovnoměrně rozděleny).

Hodnoty z posunutého exponenciálního rozdělení generujeme tak, že generujeme hodnoty $y - A$ podle předchozího návodu a hodnoty y dopočítáme. Tedy

$$y = A - \frac{1}{a} \ln(F).$$

10 Elementární úlohy

10.1 Odhad parametrů normálního regresního modelu

```

clc, clear all, close all
addpath 'funct'
% Simulation, estimation and prediction with
% SISO regression model with normal noise

nD=100; % length of time run

% parameters for simulation
% b0 a1 b1 a2 b2 ... k
th=[1 .6 -.8 .2 .1 1]; % regression coefficients
s=.1; % std. of noise
% Rem: the parameters must have the indicated order, they
% also determine the model order.

y=zeros(1,nD);
u=randn(1,nD);

nE=2; % order for estimation
nP=3; % length of prediction

% computed dimensions
nTh=length(th); % length of sim. parameters
nS=nTh/2-1; % order of sim. model
nV=2*nE+3; % order of inf. matrix
n1=max(nS,nE)+1; % beginning of time run
n2=nD-nP+1; % end of time run

% Time loop
V=.1*eye(nV); % init. inf. matrix
for t=n1:n2 % TIME LOOP beginning -----
    % simulation
    ps=genps(nS,t,y,u); % regression vector
    y(t)=th*ps'+s*randn; % simulated output
    % estimation
    Ps=[y(t) genps(nE,t,y,u)]; % regression vector
    V=V+Ps'*Ps; % information matrix
    Eth(:,t)=v2thN(V/t,1); % point estimates
    % prediction
    yi=y(1:t-1); % older data (at time t)
    for j=0:(nP-1) % loop of prediction

```



```

        ps=genps(nE,t+j,yi,u);    % regression vector
        yi(t+j)=ps*Eth(:,t);    % auxiliary prediction
    end
    yp(t+nP-1)=yi(t+nP-1);    % final prediction at t
end    % TIME LOOP end _____

```

% Results

```

s=1:nD;
set(figure(1),'position',[650 50 300 300])
plot(s,y,'b',s,yp,'r:');
title('Output and its prediction')

```

```

set(figure(2),'position',[1000 50 300 300])
plot(Eth')
title('Evolution of estimated parameters')

```

10.2 Odhad parametrů regresního modelu s rovnoměrným šumem

clc, clear all, close all

% Simulation, estimation and prediction of
 % SISO regression model with uniform noise

```

nD=200;    % length of time run

```

```

% parameters for simulation
a=-3; b=5;

```

```

y=zeros(1,nD);
u=randn(1,nD);

```

% computed dimensions

```

n1=2;    % beginning of time run
n2=nD-1;    % end of time run

```

```

Ea=1000;    % initial statistics
Eb=-1000;    % (lower and upper bound)

```

% Time loop

```

for t=n1:n2    % TIME LOOP beginning _____
    % simulation
    y(t)=(b-a)*rand+a;    % simulated output

```

```

    % estimation
    Ea(t)=min(Ea(t-1),y(t));    % update of the statistics

```

```

Eb(t)=max(Eb(t-1),y(t));    % update of the statistics

% prediction
yp(t)=(Eb(t)+Ea(t))/2;
end                            % TIME LOOP end _____

% Results
fprintf('Lower bound = %g , estimate = %g\n',a,Ea(n2))
fprintf('Upper bound = %g , estimate = %g\n',b,Eb(n2))

s=n1:n2;
set(figure(1),'position',[750 50 600 500])
plot(s,Ea(s),s,Eb(s))
axis([1 nD Ea(n2)-1 Eb(n2)+1])

set(figure(2),'position',[150 50 600 500])
plot(s,y(s),s,yp(s))

```

10.3 Odhad modelu s exponenciálním rozdělením šumu

```

clc, clear all, close all
% Simulation and estimation of
% SISO shifted exponential distribution
% f(y|a,A)=a.exp(-a(y-A)) pro y>=A, jinak nula

nD=500;                            % length of time run

% parameters for simulation
a=.6; A=2;                            % coefficients

y=zeros(1,nD);
u=randn(1,nD);

% Time loop
S=0; ka=0; my=inf;                    % init. statistics
for t=1:nD                              % TIME LOOP beginning _____
    % sumulation
    y(t)=A-log(rand)/a;                % simulated output
    % estimation
    S=S+y(t);
    ka=ka+1;
    if y(t)<my, my=y(t); end
    Ea=1/(S/ka-my);
    EA=my;
end

```

```

    Eth(:,t)=[Ea; EA];
end % TIME LOOP end -----

% Results
disp('Estimated parameters')
Ea,EA

set(figure(1),'position',[650 50 300 300])
[f,x]=hist(y,20);
bar(x,f)
title('Histogram of the output')
axis([0 max(x) 0 max(f)])

set(figure(2),'position',[1000 50 300 300])
plot(Eth')
title('Evolution of estimated parameters')
legend('Ea','EA',0)

```

10.4 Odhad diskrétního modelu s kategoričným rozdělením

```

clc, clear all, close all
addpath 'funct'
% simulation, estimation and prediction
% with discrete (categorical) model
% f(y(t)|u(t),y(t-1)), y in {0,1,2}, u in {0,1}

nd=2000;
y=zeros(1,nd);
u=rand(1,nd)<.3; % generation of u

%
% u(t) y(t-1) % parameter
th=[.01 .98 .01; % 0 0
     .01 .01 .98 % 0 1
     .97 .02 .01 % 0 2
     .02 .97 .01 % 1 0
     .97 .02 .01 % 1 1
     .01 .07 .92]; % 1 2

% Simulation
for t=2:nd
    j=3*u(t)+y(t-1)+1; % row in th
    y(t)=sum(rand>cumsum(th(j,:)));
end

```

```

% Estimation
nu=ones(6,3); % staistics
for t=2:nd
    j=3*u(t)+y(t-1)+1; % row in nu
    nu(j,y(t)+1)=nu(j,y(t)+1)+1; % update of nu
    est=fnorm(nu,2); % point estimate
    Eth(:,t)=est(:);
end

% Prediction
for t=2:nd
    j=3*u(t)+y(t-1)+1; % row in th
    Et=reshape(Eth(:,t),6,3); % parameter estimate
    yp(t)=sum(rand>cumsum(Et(j,:))); % prediction
end

% RESULTS
figure(1)
plot(Eth')
tit='Evolution of parameter estimates';
xla='time';
yla='values';
title(tit,'fontsize',22,'FontName','Times')
xlabel(xla,'fontsize',18,'FontName','Times')
ylabel(yla,'fontsize',18,'FontName','Times')

s=(nd-50+1):nd;
figure(2)
plot(s,y(s),'o',s,yp(s),'x')
axis([min(s)-1 max(s)+1 -.1 2.1])
h=legend('output','prediction',0);
set(h,'fontsize',14,'FontName','Times')
tit='Evolution of parametr estimates';
xla='time';
yla='values';
title(tit,'fontsize',22,'FontName','Times')
xlabel(xla,'fontsize',18,'FontName','Times')
ylabel(yla,'fontsize',18,'FontName','Times')

```

10.5 Off-line odhad parametrů modelu

```

clc, clear all, close all
% Simulation of RM, off-line estimation, 1st-prediction
% - influence of amplitude and type of noise to estimates and prediction

```

```

nD=1000;

b0=[1 -.8 .2;
    0  1 .3];
b1=[.1  0  0;
    0 -.2 .1];
a1=[.6 .1;
    -.3 .2];
k=[2; -1];
s=5.5*[1 .2; 0 1];

y=zeros(2,nD);
u=randn(3,nD);

Typ=5; % type of noise
for t=2:nD
    switch Typ
        case 1
            e(:,t)=randn(2,1); % normal
        case 2
            e(:,t)=rand(2,1) -.5; % uniform
        case 3
            e(:,t)=-log(rand(2,1)); % exponential
        case 4
            df=5; % deg. of freedom
            ee=randn(2,df);
            e(:,t)=diag(ee*ee'); % chi2
        case 5
            df=1; % deg. of freedom
            ee=randn(2,df);
            e(:,t)=randn(2,1)./sqrt(diag(ee*ee')./df); % Student
    end

    y(:,t)=b0*u(:,t)+a1*y(:,t-1)+b1*u(:,t-1)+k+s*e(:,t);
end

q=2:nD;
nk=length(q);
Y=y(:,q)'; X=[u(:,q)' y(:,q-1)' u(:,q-1)' ones(nk,1)];
Et=inv(X'*X)*X'*Y; Et'

yp=(X*Et)';

```

```

set (figure (1), 'position', [650 50 600 500])
subplot (311), plot (q, y (1, q), q, yp (1, :)), hold on
title ('y1 and yp1')
subplot (312), plot (q, y (2, q), q, yp (2, :))
title ('y2 and yp2')
subplot (313), plot (q, e (1, q), q, e (2, q)), hold off
title ('e1 and e2')

```

10.6 Exponenciální zapomínání v odhadu parametrů modelu

```

clc, clear all, close all
addpath 'funct'
% Estimation of  $y(t)=a \cdot y(t-1)+bu(t)+e(t)$  with forgetting

nd=1500;
phi=.995; % forgetting factor
a=.3; b=1; sd=.01; % sim. parametrs
u=sin (10*(1:nd)/nd)+.1*randn (1,nd); % input
V=zeros (3);

y=zeros (1,nd);
for t=2:nd % time loop
    % simulation
    b=b+.01*randn; % variations of b
    y (t)=a*y (t-1)+b*u (t)+sd*randn;

    % estimation
    Ps=[y (t) y (t-1) u (t)]'; % regression vector
    V=phi*V+Ps*Ps'; % statistics update
    Et=v2thN (V/t, 1); % point estimates

    % remember
    bt (t)=b;
    Ea (t)=Et (1);
    Eb (t)=Et (2);
end

% RESULTS
s=1:nd;
figure (1)
plot (s, bt (s), 'r', s, Eb (s), 'm', s, Ea (s), 'b')
axis ([min (s)-1 max (s)+1 0 1.5])
legend ('sim b', 'est b', 'est a', 0)

```

10.7 Odhad struktury regresního modelu

```

clc, clear all, close all
addpath 'funct'
% Structure determination using linear regression

nd=300; % number of prior data
a=.6; b0=1; b1=.2; c=1; sd=.1; % simulation coefficients

% Simulation
y=zeros(1,nd); % output
u=randn(1,nd); % input
v=sin(10*(1:nd)/nd)+.1*randn(1,nd); % external variable
w=.2*abs(cos(35*(1:nd)/nd+5)); % not used external
for t=2:nd
    y(t)=a*y(t-1)+b0*u(t)+b1*u(t-1)+c*v(t)+sd*randn; % model
end

% Normalization of variables
y=(y-mean(y))/std(y);
u=(u-mean(u))/std(u);
v=(v-mean(v))/std(v);
w=(w-mean(w))/std(w);

% Estimation
V=zeros(9);
for t=3:nd
    % regression vector with some wrong variables
    Ps=[y(t);y(t-1);y(t-2);u(t);u(t-1);u(t-2);v(t);w(t);1];
    V=V+Ps*Ps'; % update of statistics
end

th=v2thN(V/nd,1) % point estimates of parameters
figure, bar(th) % plot of results
% From the regression vector, the variables corresponding
% to almost zero coefficients are included. The border is
% heuristic and can be set so that the resulting model is
% reasonable :)

tit='Regeesion coefficients';
xl='order in regression vector';
yl='values';

title(tit, 'fontsize', 22, 'FontName', 'Times')

```

```
xlabel(xl,'fontsize',18,'FontName','Times')  
ylabel(yl,'fontsize',18,'FontName','Times')
```


11 Knihovna funkcí

V této kapitole budou uvedeny a popsány funkce, které s využívají ve dříve uvedených programech.

11.1 c2c.m

– výpočet bodových odhadů ukazovátka z hp $f(c_t|d(t))$ pro $t = 1, 2, \dots$

```
function [p Tn]=c2c(ct,Ect)
% cc=c2c(ct,Ect)    permutation of pointer values
% ct      simulated pointer
% Ect     estimated pointer
% np      length of prediction
%
% USAGE:
% [pt vt]=c2c(ct,Ect);
% Tct=pt(Ect);

m=max(ct);                % number of components
T=zeros(m,m);
n=min(length(ct),length(Ect));% common area for simulatio
                                % and estimation

for i=1:n
    T(ct(i),Ect(i))=T(ct(i),Ect(i))+1; % transf. matrix
end
Tn=fnorm(T,2);            % normalization of T

% computation of transformation vector
% p(sim) <-> est ...> plot(p(sim),est) is OK
p=[];
for i=1:m
    [xxx mx]=max(Tn(i,:));
    p=[p mx];
end

% Check no cluster is caught by more than one comooonent
for i=1:n
    ii=sum(i==p);
    if ii>1
```

```
    disp('*** Components are overlapping ***')
end
end
```

11.2 *catKL.m*

– Kullback_Leiblerova divergence pro kategorická rozdělení

```
function d=catKL(p,q)
% d=catKL(p,q)      Kullback-Leibler for categorical dist.

p=p(:)/sum(p);
q=q(:)/sum(q);

d=sum(p.*log(p./q));
```

11.3 *drift.m*

– generátor skoro-náhodné procházky

```
function y=drift(nd)
% clc , clear all
%
% nd=1000;

y(1)=0;
a=nd*rand;
for i=2:nd
    y(i)=.05*sin(5*i/nd-a)+.9*y(i-1)+.01*rand;
end

% fig , plot(y)
% axis([1 nd -1 1])
```

11.4 fnorm.m

– normování vektoru (matice) na součet 1

```
function fn=fnorm(f,i)
% fn=fnorm(f,i)  normalization of probabilistic table
% fn  normalized table
% f   table
% i   direction i=1 norm columns, i=2 norm rows

if nargin==1,
    sf=sum(sum(f));
    fn=f/sf;
    return ,
end
[m n]=size(f);
if i==1
    f1=sum(f,1);
    fn=f./(ones(m,1)*f1);
else
    f2=sum(f,2);
    fn=f./(f2*ones(1,n));
end
```

11.5 GaussN.m

– hp vícerozměrného normálního rozdělení

```
function [p Lp]=GaussN(x,m,R)
% [p Lp]=GaussN(x,m,R)  value of multivariate Gaussian pdf

% p      probability
% Lp     logarithm of prob.
% x      realization
% m      expectation
% R      covariance matrix

x=x(:);          % column vector
m=m(:);          % column vector
```

```

n=max(size(R));
Lp=-.5*(n*log(2*pi)+log(det(R)));
ex=(x-m)'*inv(R+1e-5*eye(size(R)))*(x-m);
Lp=Lp-.5*ex;
p=exp(Lp);

```

11.6 *genps.m*

– generování regresního vektoru pro lineární regresní vektor

```

function ps=genps(n,t,y,u)
% ps=genps(n,t,y,u)    generation of regression vector \psi
%                       i.e. without y(t)
% n    order
% t    time
% y    output
% u    input

ps=u(:,t)';
for i=1:n
    ps=[ps y(:,t-i)' u(:,t-i)'];
end
ps=[ps 1];

```

11.7 *logReg.m*

– odhad a predikce s logistickým modelem

```

function [py yp yr b]=logReg(b,y,x)
% logReg(b,y,x)    prediction for LR with multinomial Y
%
% yp    prediction in probabilities
% yr    prediction in values 0,1 (rounded yp)
% b     estimates of parameter (if performed)
% b     regression coefficients (b=[] – with estimation)
% y     regressand – independent
% x     regressor – dependant

```

```

%
if isempty(b)
    % Estimation
    opt.Display='off';
    opt.MaxFunEvals=10000;
    opt.MaxIter=10000;
    b0=1*randn(size(x,2)+1,max(y));
    [b,v0,ex,out]=fminsearch(@(b) -lrLik(b,y,x),b0,opt);
    if ex<1, fprintf('\n Wrong exit condition: ex = %d\n',ex), end
end

% Prediction
[n m]=size(x);
xp=[ones(n,1) x];
ex=exp(xp*b);
se=sum(ex,2);
dn=(se+1)*ones(1,max(y));
py=ex./dn;
py=[1-sum(py,2) py];
yr=zeros(n,1);
for i=1:n
    [mx ix]=max(py(i,:));
    yr(i)=ix-1;
end
yp=py*[0:size(py,2)-1]';

```

11.8 lrLik.m

– výpočet log-likelihoodu pro logistickou regresi

```

function logL=lrLik(b,y,x)
% logL=lrLik(b,y,x)    log-likelihood for logistic regression
% logL    logarithm of likelihood
% b        parameters of logistic model
% y        output
% x        regression vector

[n m]=size(x);          % number of data , length of x
K=max(y);               % number of values of y
x1=[ones(n,1) x];      % full regression vector
m1=m+1;                 % length of x1

```

```

ny=zeros(1,K+1);
logL=0;
% evaluation of log-likelihood
for i=1:n
    z=x1(i,:)*b;
    p=exp(z)./(1+sum(exp(z)));
    p0=1/(1+sum(exp(z)));
    j=y(i);
    if j==0
        logL=logL+log(p0);
    else
        logL=logL+log(p(j));
    end
end
end

```

11.9 normKL.m

– Kullback_Leiblerova divergence pro normální rozdělení

```

function d=normKL(m1,m2,r1,r2)
% d=normKL(m1,m2,r1,r2)      KL distance of two normal dist.
%                               d=KL(N1|N2)
% m1, m2  expectations
% r1, r2  covariances

m1=m1(:);
m2=m2(:);

n=length(m1);
d1=trace(inv(r2)*r1);
d2=(m2-m1)'*inv(r2)*(m2-m1);
d3=det(r2)/det(r1);

d=.5*(d1+d2+log(d3)-n);

```

11.10 scal.m

– škálování dat (střední hodnota 0, rozptyl 1)

```

function [y,mD,sD]=scal(x,mD,sD)
% y=scal(x)    scaling data for init
% y  scaled data
% mD mean of channels
% sD standard deviation of channels
% x  data to be filtered (in rows !!!)

[m,n]=size(x);
if nargin<2
    mD=mean(x')';
    sD=sqrt(var(x'))';
end
ii=find(sD==0);
sD(ii)=1;
mD=mD(:); sD=sD(:);
y=(x-mD*ones(1,n))./(sD*ones(1,n));

```

11.11 seed.m

– nastavení “seed” pro náhodné generátory rand a randn

```

function seed(n1,n2)
% seed(n1,n2)    sets seed for rand and randn
% n1            number for rand
% n2            number for randn

if nargin<2, n2=97531; end
if nargin<1, n1=13579; end

rand('seed',n1), randn('seed',n2);

```

11.12 th2v.m

– konstrukce apriorní informační matice z apriorních dat

```

function V=th2v(th)

```

```

ni=100;
n=length(th)
m=length(th{1});

for i=1:n
    Psi=[th{i}*ones(1,ni)+.01*randn(m,ni);
         ones(1,ni)];
    V{i}=Psi*Psi'*2/ni;
end

```

11.13 unscal.m

– odškálování dat

```

function y=unscal(x,mD,sD)
% y=unscal(x,mD,sD) uncsale the data x
% y uncsaled data
% x data to be uncsaled
% mD mean of the channels
% sD std of the channels

[m,n]=size(x);
mD=mD(:); sD=sD(:);
y=diag(sD)*x+mD*ones(1,n);

```

11.14 uut.m

– výpočet odmocniny z matice: $C=UU'$, kde U je horní trojúhelník

```

function u=uut(m)
% u=uut(m) m=uu' UL-factorization of sym. poz.def. matrix
% u upprer triangular matrix
% m symmetric poz.def. square matrix

n=length(m);
for i=n:-1:1
    s=0;

```



```

for j=i+1:n
    s=s+u(i,j)^2;
end
u(i,i)=sqrt(m(i,i)-s);
for k=i-1:-1:1
    s=0;
    for j=i+1:n
        s=s+u(i,j)*u(k,j);
    end
    u(k,i)=(m(k,i)-s)/u(i,i);
end
end

```

11.15 v2thN.m

– konstrukce bodových odhadů parametrů regresního modelu ze statistik

```

function [th,s2]=v2thN(V,m)
% [th,s2]=v2thN(V,m)
% Computation of parameter estimates
% from normalized information matrix
% m dimension of y
if nargin<2
    m=1;
end
% partitioning of information matrix
Vy=V(1:m,1:m);
Vyf=V(m+1:end,1:m);
Vf=V(m+1:end,m+1:end);
% computation of point estimates
th=inv(Vf+1e-5*eye(size(Vf)))*Vyf;
s2=(Vy-Vyf'*th);
th=th';

```

11.16 wflt.m

– filtrace signálu na symetrickém okně

```

function [df dp]=wflt(j,k,dtI)
% [df dp]=wflt(j,k,dtI) daily course construction
% df      filtered daily course
% dp      rough daily course
% j       days in a week
% k       weeks in data sample
% dtI     data sample
% Works for period 5 minutes = 288 samples per day

i0=7*288;          % numb. of samples per week
i1=[];
for i=j           % samples within a week
    i1=[i1 (i-1)*288+(1:288)];
end
ii=[];
for i=k          % samples for the whole data (8 weeks)
    ii=[ii i1+(i-1)*i0];
end
dd=dtI(ii);     % chosen data for DC
dI=[];
for i=1:length(j)*length(k) % matrix of DCses
    dI=[dI; dd(((i-1)*288+1):(i*288))];
end
dp=mean(dI);    % rough daily course
nf=10;         % filtration (by central window)
dp0=[dp(1:nf) dp dp((end-nf+1):end)]; % averaging)
for i=1:288
    df(i)=mean(dp0((i):(i+2*nf))); % filtered DC
end

```

Reference

- M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions*. Dover Publications, New York, 1972.
- A. Agresti. *Categorical data analysis*. Wiley, Hoboken, 2002.
- B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Prentice Hall, 1979.
- B.D.O. Anderson and J.B. Moore. *Optimal Control : Linear Quadratic Methods*. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.
- M. Aoki. *Optimization of Stochastic Systems: Topics in Discrete-Time Systems*. Academic Press, New York, London, 1967.
- K.J. Astrom. *Introduction to Stochastic Control*. Academic Press, New York, 1970.
- K.J. Astrom. Theory and applications of adaptive control. *Automatica*, 19(5):471–486, 1983.
- K.J. Astrom and B. Wittenmark. *Adaptive Control*. Addison-Wesley, Reading, Massachusetts, 1989.
- M. Basseville and I.V. Nikiforov. *Detection of abrupt changes: theory & applications*. Prentice Hall, Englewood Cliffs, New Jersey, 1993. ISBN 0 13 126780 9.
- R. Bellman. *Dynamic Programming*. Princeton University Press, New York, 1957.
- R. Bellman. *Introduction to the Mathematical Theory of Control Processes*. Academic Press, New York, 1967.
- J. M. Bernardo. Expected information as expected utility. *The Annals of Statistics*, 7(3):686–690, 1979.
- J.M. Bernardo and A.F.M. Smith. *Bayesian Theory*. John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore, 2 edition, 1997.
- G.J. Bierman. *Factorization Methods for Discrete Sequential Estimation*. Academic Press, New York, 1977.
- D.W. Clarke. *Advances in Model-Based Predictive Control*. Oxford University Press, Oxford, 1994.
- D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control. *Automatica*, 23(2):137–160, 1987.
- B. de Finetti. *Theory of Probability: A Critical Introductory Treatment*. J. Wiley, New York, 1970.
- M.H. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, New York, 1970.

- A. P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.
- James S. Dyer, Peter C. Fishburn, Ralph E. Steuer, Jyrki Wallenius, and Stanley Zionts. Multiple criteria decision making, multiattribute utility theory: The next ten years. *Management Sciences*, 38(5):645–654, 1992.
- R.H. Elliot, L. Assoun, and J.B. Moore. *Hidden Markov Models*. Springer-Verlag, New York, 1995.
- B. S. Everitt and D. J. Hand. *Applied Multivariate Data Analysis*. Edward Arnold, London, 1991.
- B.S. Everitt and D.J. Hand. *Finite Mixture Distributions*. Chapman and Hall, 1981.
- P. Eykhoff. *System Identification, Parameter and State Estimation*. John Wiley, New York, 1974.
- A.A. Feldbaum. Theory of dual control. *Autom. Remote Control*, 21(9), 1960.
- A.A. Feldbaum. Theory of dual control. *Autom. Remote Control*, 22(2), 1961.
- T.S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1:209–230, 1973.
- M. Fiedler. *Special matrices and their use in numerical mathematics*. SNTL, Prague, 1981. in Czech.
- W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov Chain Monte Carlo in practice*. Chapman & Hall, London, 1997. ISBN 0 412 05551 1.
- G.H. Golub and C.F. VanLoan. *Matrix Computations*. The John Hopkins University Press, Baltimore – London, 1996. 3rd edition.
- S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan, New York, 1994.
- J. Homolová and I. Nagy. Traffic model of a microregion. In P. Horáček, M. Šimandl, and P. Zitek, editors, *Preprints of the 16th World Congress of the International Federation of Automatic Control*, pages 1–6, Prague, July 2005. IFAC.
- O.L.R. Jacobs. *An Introduction to Dynamic Programming*. Chapman and Hall Ltd., London, 1967.
- A.M. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.
- F.V. Jensen. *Bayesian Networks and Decision Graphs*. Springer-Verlag, New York, 2001.

- N.L. Johnson and S. Kotz. *Continuous univariate distributions-1*. John Wiley & Sons, 1970.
- M. Kárný, P. Nedoma, T. V. Guy, J. Knížek, and I. Nagy. Automatický systém podporující operátory složitých procesu. *Automatizace*, 46(3):177–179, 2003.
- M. Kárný, J. Böhm, T. V. Guy, L. Jirsa, I. Nagy, P. Nedoma, and L. Tesař. *Optimized Bayesian Dynamic Advising: Theory and Algorithms*. Springer, London, 2005.
- D.F. Kerridge. Inaccuracy and inference. *Journal of Royal Statistical Society*, B 23: 284–294, 1961.
- J. Kratochvílová and I. Nagy. Traffic control of microregion. In J. Andryšek, M. Kárný, and J. Kracík, editors, *CMP'04: MULTIPLE PARTICIPANT DECISION MAKING, Theory, algorithms, software and applications*, pages 161 – 171, Adelaide, May 2004a. Advanced Knowledge International.
- J. Kratochvílová and I. Nagy. Model dopravní mikrooblasti. *Automatizace*, 47(12):752–758, 2004b.
- R. Kulhavý. Directional tracking of regression-type model parameters. In *Preprints of the 2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing*, pages 97–102, Lund, Sweden, 1986.
- R. Kulhavý. Restricted exponential forgetting in real-time identification. *Automatica*, 23(5):589–600, 1987.
- R. Kulhavý. Recursive nonlinear estimation: A geometric approach. *Automatica*, 26(3): 545–555, 1990.
- R. Kulhavý and M. B. Zarrop. On a general concept of forgetting. *International Journal of Control*, 58(4):905–924, 1993.
- S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–87, 1951.
- L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, London, 1987.
- M. Loeve. *Probability Theory*. van Nostrand, Princeton, New Jersey, 1962. Russian translation, Moscow 1962.
- J.S. Long and J. Freese. *Regression models for categorical dependent variables using Stata*. Stata Press, 2005. 2nd edition.
- J. J. Martin. *Bayesian Decision Problems and Markov Chains*. J. Wiley, New York, 1967.
- J.S. Meditch. *Stochastic Optimal Linear Estimation and Control*. Mc. Graw Hill, 1969.

- R.K. Mehra and D.G. Lainiotis (Eds.). *System Identification – Advances and Case Studies*. Pergamon Press, New York, 1976.
- M.H.DeGroot and M.J.Schervish. *Probability and Statistics*. Addison-Wesley, ISBN 0-321-20473-5, 2002.
- E. Mosca. *Optimal, Predictive, and Adaptive Control*. Prentice Hall, 1994.
- R. Murray-Smith and T.A. Johansen. *Multiple Model Approaches to Modelling and Control*. Taylor & Francis, London, 1997.
- I. Nagy. Stochastické systémy. <http://www.fd.cvut.cz/personal/nagyivan/SSA/StSysTexty.pdf>, 2012.
- I. Nagy and F. Dušek. Theory and implementation of paper cross-profile adaptive control. In K. Warwick, M. Kárný, and A. Halousková, editors, *Advanced Methods in Adaptive Control for Industrial Application*, volume 158. Springer Verlag, Lecture Notes, 1991.
- I. Nagy, M. Kárný, J. Novovičová, and M. Valečková. Mixture-model Identification in Traffic Control Problems. Technical Report 1973, ÚTIA AV ČR, Praha, 1999.
- I. Nagy, P. Nedoma, M. Kárný, L. Pavelková, and P. Ettlér. O bayesovském učení. *Automa*, (7):56–60, 2002.
- I. Nagy, M. Kárný, P. Nedoma, and Š. Voráčová. Bayesian estimation of traffic lane state. *International Journal of Adaptive Control and Signal Processing*, 17(1):51–65, 2003.
- I. Nagy, E. Suzdaleva, and M. Kárný. Bayesian estimation of mixtures with dynamic transitions and known component parameters. *Kybernetika*, 47(4):572–594, 2011a.
- I. Nagy, E. Suzdaleva, M. Kárný, and T. Mlynářová. Bayesian estimation of dynamic finite mixtures. *Int. Journal of Adaptive Control and Signal Processing*, 25(9):765–787, September 2011b. doi: 10.1002/acs.1239.
- V. Kumar P. Tan, M. Steinbach. *Introduction to Data Mining*. Pearson Education, Inc., ISBN 0-321-32136-7, 2006.
- R. Patton, P. Frank, and R. Clark. *Fault Diagnosis in Dynamic Systems: Theory & Applications*. Prentice Hall, 1989.
- V. Peterka. Bayesian approach to system identification. In P. Eykhoff, editor, *Trends and Progress in System Identification*, pages 239–304. Pergamon Press, Oxford, 1981.
- A. Quinn, P. Ettlér, L. Jirsa, I. Nagy, and P. Nedoma. Probabilistic advisory systems for data-intensive applications. *International Journal of Adaptive Control and Signal Processing*, 17(2):133–148, 2003.

- C.R. Rao. *Linear method of statistical inference and their applications*. Academia, Prague, 1987a. in Czech.
- M.M. Rao. *Measure Theory and Integration*. John Wiley, New York, 1987b.
- A. Renyi. *Probability theory*. Academia, Prague, 1972. in Czech.
- J. Spousta. Algorithms for Bayesian estimation of spline model structure. *Kybernetika*, 28:413–423, 1992.
- E. Suzdaleva and I. Nagy. Recursive state estimation for hybrid systems. *Applied Mathematical Modelling*, 36(4):1347–1358, 2012a. doi: 10.1016/j.apm.2011.08.042.
- E. Suzdaleva and I. Nagy. Experiments with pid controller for fuel consumption optimization. Technical Report 2326, ÚTIA AV ČR, Praha, December 2012b.
- E. Suzdaleva, I. Nagy, and L. Pavelková. Fuel consumption optimization: Early experiments. In *Preprints of the 16th IFAC Symposium on System Identification Sysid 2012*, pages 751–756, Brussels, Belgium, July 11 – 13 2012a.
- E. Suzdaleva, I. Nagy, L. Pavelková, and T. Mlynářová. Servo problem within fuel consumption optimization. In *Proceedings of IASTED International Conference on Engineering and Applied Science*, pages 100–107, Colombo, Sri Lanka, December 27-29 2012b.
- R. Taylor. *Introduction to Functional Analysis*. Academia, Prague, 1973. (Czech translation).
- D.M. Titterton, A.F.M. Smith, and U.E. Makov. *Statistical Analysis of Finite Mixtures*. John Wiley, New York, 1985.
- V.S. Vladimirov. *Generalized Functions in Mathematical Physics*. Mir Publishers, Moscow, 1979.
- J. Štecha and V. Havlena. Teorie dynamických systémů. *Je možno nalézt na webu*.
- Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. Technical Report 95-041, UNC-CH Computer Science, 1995.
- P.E. Wellstead and M.B. Zarrop. *Self-tuning Systems*. John Wiley, Chichester, 1991.
- M. Zarrop. *Experiment Design for Dynamic System Identification. Lecture Notes in Control and Information Sciences 21*. Springer, New York, 1979.
- A. Zellner. *An Introduction to Bayesian Inference in Econometrics*. J. Wiley, New York, 1976.