

Stochastické Systémy

Ivan Nagy¹

Obsah

| | |
|---|----------|
| 1 Úvod | 7 |
| 1.1 Opakování pravděpodobnosti | 7 |
| 1.1.1 Příklad [náhodná veličina] | 7 |
| 1.1.2 Příklad [pravděpodobnostní funkce] | 7 |
| 1.1.3 Příklad [pravděpodobnost, že ...] | 8 |
| 1.1.4 Poznámka | 8 |
| 1.1.5 Příklad [sdružená pravděpodobnostní funkce] | 8 |
| 1.1.6 Příklad [výpočet marginální a podmíněné hp] | 10 |
| 1.1.7 Příklad [marginální a podmíněné rozdělení pro normální hp] | 11 |
| 1.1.8 Příklad [podmíněné a marginální rozdělení pro diskrétní pf] | 12 |
| 1.1.9 Poznámka | 14 |
| 1.2 Systém a veličiny s ním spojené | 14 |
| 1.2.1 Příklad [proces a systém] | 14 |
| 1.2.2 Poznámka | 15 |
| 1.2.3 Poznámka | 15 |

¹FD ČVUT Praha, Katedra Aplikované matematiky

| | | |
|----------|---|-----------|
| 2 | Spojité model | 16 |
| 2.1 | Princip stochastického modelu | 16 |
| 2.1.1 | Příklad [stochastický model] | 17 |
| 2.2 | Lineární regresní model s normálním šumem | 18 |
| 2.3 | Diferenciální a diferenční rovnice ve spojitém modelu | 19 |
| 2.3.1 | Diskretizace diferenciální rovnice | 19 |
| 2.3.2 | Typy diferenciálních a diferenčních rovnic | 19 |
| 2.4 | Regresní model ve stavovém tvaru | 20 |
| 2.4.1 | Příklad [regresní model] | 21 |
| 3 | Diskrétní a logistický model | 22 |
| 3.1 | Diskrétní model | 22 |
| 3.1.1 | Příklad [diskrétní model] | 22 |
| 3.2 | Logistický model | 23 |
| 3.2.1 | Příklad [logistický model] | 23 |
| 4 | Odhad spojitého modelu | 24 |
| 4.1 | Bayesovské odhadování | 24 |
| 4.1.1 | Parametry z hlediska bayesovství | 24 |
| 4.1.2 | Bayesův vztah | 26 |
| 4.1.3 | Reprodukovatelné parametrické vyjádření aposteriorní hp | 26 |
| 4.1.4 | Příklad [odhadování bez reprodukce aposteriorní hp] | 27 |
| 4.1.5 | Příklad [Odhadování s reprodukcí aposteriorní hp] | 27 |
| 4.1.6 | Výsledek odhadování | 28 |
| 4.1.7 | Odhad výstupu | 28 |
| 4.1.8 | Poznámka | 28 |
| 4.1.9 | Bodový odhad výstupu | 29 |
| 4.2 | Odhad parametrů regresního modelu | 29 |
| 4.2.1 | Model | 29 |
| 4.2.2 | Rekurze pro statistiku | 30 |
| 4.2.3 | Algoritmus odhadu | 30 |
| 4.2.4 | Bodový odhad výstupu s bodovým odhadem parametru | 31 |
| 4.2.5 | Příklad [odhad regresního modelu] | 31 |
| 4.2.6 | Příklad [bodový odhad regresního modelu] | 33 |
| 4.2.7 | Poznámka | 34 |

| | | |
|----------|---|-----------|
| 5 | Odhad diskrétního a logistického modelu | 34 |
| 5.1 | Odhad parametrů diskrétního modelu | 34 |
| 5.1.1 | Model a jeho součinný tvar | 35 |
| 5.1.2 | Příklad [model falešné mince] | 35 |
| 5.1.3 | Statistika | 35 |
| 5.1.4 | Přepočet statistiky | 36 |
| 5.1.5 | Bodový odhad parametrů | 36 |
| 5.1.6 | Bodový odhad výstupu | 37 |
| 5.1.7 | Příklad [odhad s diskrétním modelem] | 37 |
| 5.2 | Odhad parametrů logistického modelu | 40 |
| 5.2.1 | Odhad logistického modelu | 40 |
| 5.2.2 | Předpověď výstupu | 40 |
| 5.2.3 | Příklad [odhad logistické regrese 1] | 41 |
| 5.2.4 | Příklad [odhad s logistickým modelem 2] | 41 |
| 5.3 | Klasifikace | 42 |
| 5.3.1 | Příklad [klasifikace s diskrétním modelem] | 43 |
| 5.3.2 | Příklad [klasifikace s logistickou regresí] | 43 |
| 6 | Předpověď modelované veličiny | 44 |
| 6.1 | Jednokroková předpověď | 44 |
| 6.1.1 | Příklad [jednokroková predikce s regresním modelem] | 45 |
| 6.1.2 | Poznámka | 45 |
| 6.2 | Víceková předpověď | 45 |
| 6.3 | Víceková bodová předpověď | 47 |
| 6.3.1 | Předpověď se spojitým modelem | 47 |
| 6.3.2 | Příklad [bodová předpověď] | 47 |
| 6.3.3 | Příklad [předpověď pro normální rozdělení] | 48 |
| 6.3.4 | Předpověď s diskrétním modelem | 48 |
| 6.3.5 | Příklad [předpověď s diskrétním modelem] | 48 |
| 6.3.6 | Poznámka | 50 |

| | |
|--|-----------|
| 7 Stav, stavový model a odhad stavu | 50 |
| 7.1 Stav | 50 |
| 7.1.1 Příklad [pojem stavu] | 50 |
| 7.2 Stavový model | 50 |
| 7.3 Odhad stavu | 51 |
| 7.3.1 Poznámka | 52 |
| 7.4 Kalmanův filtr | 52 |
| 7.4.1 Příklad [odhad stavu] | 53 |
| 7.4.2 Poznámka k příkladu | 54 |
| 8 Řízení se spojitým modelem | 54 |
| 9 Řízení s diskrétním modelem | 58 |
| 10 Přílohy | 61 |
| 10.1 Doplnění na čtverec | 61 |
| 10.2 Přirozené podmínky řízení | 61 |
| 10.3 Bayesův vzorec | 62 |
| 10.3.1 Odvození Bayesova vzorce | 62 |
| 10.3.2 Aplikace Bayesova vzorce | 63 |
| 10.4 Multinomiální rozdělení | 63 |
| 10.5 Dirichletovo rozdělení | 64 |
| 10.6 Normální rozdělení | 65 |
| 10.7 Inverzní Gauss-Wishartovo (GiW) rozdělení | 65 |
| 10.8 Bodový odhad podle kvadratického kritéria | 66 |
| 10.9 Bodové odhady parametrů spojitého modelu | 67 |
| 10.10 Bodové odhady parametrů diskrétního modelu | 68 |
| 10.11 Logistická regrese | 69 |
| 10.11.1 Derivace věrohodnostní funkce | 69 |
| 10.11.2 Newtonův algoritmus | 69 |
| 10.12 Rozšířený Kalmanův filtr | 70 |
| 10.12.1 Konstrukce rozšířeného KF | 70 |
| 10.12.2 Realizace KF | 71 |

| | |
|---|-----------|
| 11 Programy | 72 |
| 11.1 Diferenciální a Diferenční rovnice | 72 |
| 11.1.1 Diferenciální rovnice | 72 |
| 11.1.2 Diferenční rovnice | 74 |
| 11.1.3 Diskretizace | 77 |
| 11.2 Simulace | 78 |
| 11.2.1 T11simCont2 | 78 |
| 11.2.2 T12simDisc | 79 |
| 11.2.3 T13simLogReg | 81 |
| 11.2.4 T14simState | 83 |
| 11.3 Odhad | 84 |
| 11.3.1 T21estCont2 | 84 |
| 11.3.2 T22estDisc | 86 |
| 11.3.3 T23LogRegM | 88 |
| 11.4 Předpověď | 89 |
| 11.4.1 T31preCont2 | 89 |
| 11.4.2 T32preDisc | 92 |
| 11.4.3 T34statEstKF | 94 |
| 11.5 Klasifikace | 96 |
| 11.5.1 T41classMix | 96 |
| 11.6 Řízení | 100 |
| 11.6.1 T51ctrlCont | 100 |
| 11.6.2 T52ctrlDisc | 102 |
| 11.7 Složitější úlohy | 105 |
| 11.7.1 T81adaptC1 | 105 |
| 11.7.2 T82adaptC2 | 107 |
| 11.7.3 T83adaptD | 111 |
| 11.7.4 T84estS1S2 | 115 |
| 11.7.5 T85estStrR | 118 |
| 11.7.6 T86estStrS | 121 |
| 11.7.7 T87estDiLo | 125 |
| 11.8 Pomocné procedury | 128 |
| 11.8.1 dsim | 128 |
| 11.8.2 psi2row | 128 |

| | | |
|---------|--|-----|
| 11.8.3 | row2psi | 129 |
| 11.8.4 | allRows | 130 |
| 11.8.5 | Gauss2 | 130 |
| 11.8.6 | v2th | 131 |
| 11.8.7 | v2th2 | 132 |
| 11.8.8 | Kalman | 132 |
| 11.8.9 | logit | 133 |
| 11.8.10 | logit_inv | 134 |
| 11.9 | Discretization of differential equations | 135 |
| 11.9.1 | Elementary differential equations | 135 |
| 11.9.2 | Elementary difference equations | 136 |
| 11.9.3 | Discretization of a continuous model | 137 |

1 Úvod

1.1 Opakování pravděpodobnosti

- **Veličina** je pojem používaný pro kvantitativní a kvalitativní popis reality. Veličina má svou jednotku a číselnou hodnotu. Určování hodnoty veličiny (tedy její velikosti ve zvolených jednotkách) nazýváme měření. Hodnoty veličin jsou buď uspořádané nebo bez uspořádání. U veličin s uspořádanými hodnotami lze hovořit o tom, že veličina při současném měření nabyla hodnoty větší nebo menší než při měření jiném.

- **Náhodná veličina** je veličina, jejíž hodnotu neznáme a nemůžeme ji změřit, nebo jsme ji dosud nezměřili, a při jejím opakovaném měření dostáváme různé hodnoty. Změřená hodnota této veličiny je její realizace.

Náhodná veličina je buď **diskrétní** (má konečný nebo spočetný počet realizací), nebo **spojitá** (její realizace mohou nabývat jakékoli hodnoty z reálné osy).

- **Rozdělení náhodné veličiny** je funkce, která jednotlivým hodnotám nebo intervalům hodnot náhodné veličiny přiřazuje pravděpodobnosti.

1.1.1 Příklad [náhodná veličina]

Sledujeme projíždějící automobily a kontrolujeme jejich rychlost. Potom “rychlost projíždějícího automobilu” je náhodná veličina. Při každém měření dostaneme totiž odlišnou hodnotu rychlosti. Konkrétní naměřené rychlosti vybraných automobilů jsou realizace.

1.1.2 Příklad [pravděpodobnostní funkce]

Rychlost projíždějících automobilů z minulého příkladu nemusí nutně představovat spojitou náhodnou veličinu, jak by se na první pohled mohlo zdát. Rychlosti můžeme definovat např.:

| Definovaná hodnota rychlosti | Interval příslušných rychlostí [km/h] |
|------------------------------|---------------------------------------|
| 1 | (0; 30) |
| 2 | (30; 60) |
| 3 | (60; 90) |

Z dlouhodobého měření jsme pomocí statistické definice pravděpodobnosti zjistili, že pravděpodobnosti výskytu jednotlivých hodnot jsou $P(1) = 0.23$, $P(2) = 0.69$ a $P(3) = 0.08$. Jestliže je doba pozorování dostatečně dlouhá, můžeme hodnoty pravděpodobnosti považovat za ustálené a rozdělení naší náhodné veličiny, kterou označíme x , ve formě pravděpodobnostní funkce psát jako následující tabulku:

| | | | |
|--------|------|------|------|
| x | 1 | 2 | 3 |
| $f(x)$ | 0.23 | 0.69 | 0.08 |

1.1.3 Příklad [pravděpodobnost, že ...]

V případě, že budeme brát rychlost automobilů jako libovolné nezáporné reálné číslo, dostáváme spojitou náhodnou veličinu x s hustotou pravděpodobnosti $f(x)$. Pomocí této hustoty lze určit pravděpodobnost libovolného intervalu $(a, b) \subset R_0^+$ takto

$$P(X \in (a, b)) = \int_a^b f(x) dx$$

- **Náhodný vektor** je vektor náhodných veličin $X = [X_1, X_2, \dots, X_n]'$.

1.1.4 Poznámka

Vektory bereme vždy jako sloupcové vektory - apostrof označuje transpozici. Konvence o sloupcových vektorech je formální a má usnadnit orientaci v maticových vzorcích.◁

- **Sdružené rozdělení** je rozdělení náhodného vektoru definované pomocí hustoty pravděpodobnosti (zkráceně hp). Pro dvousložkový vektor $X = [X_1, X_2]'$ bude zápis spojitě náhodné veličiny vypadat následovně:

$$f(x_1, x_2) = P(X_1 \leq x_1 \wedge X_2 \leq x_2)$$

a diskrétní náhodné veličiny:

$$f(x_1, x_2) = P(X_1 = x_1 \wedge X_2 = x_2).$$

Definice pro obecný náhodný vektor je analogická.

1.1.5 Příklad [sdružená pravděpodobnostní funkce]

Uvažujme pokus, při kterém měříme dopravní stupeň X na sledované vozovce v jednom x_1 i druhém x_2 směru. Pro jednoduchost uvažujme pouze 3 hodnoty dopravního stupně (1 - volno, 2 - provoz, 3 - kongesce). Náhodný vektor "dopravní stupeň" je

$$X = [x_1, x_2]'$$

a jeho sdružená pravděpodobnostní funkce (zkráceně pf), $f(x) = f(x_1, x_2)$ je podle definice tvořena pravděpodobnostmi všech možných kombinací hodnot uvažovaných dopravních stavů x_1 a x_2 . Funkci můžeme zapsat ve formě tabulky:

| $x_1 \setminus x_2$ | 1 | 2 | 3 |
|---------------------|-----------------------------|-----------------------------|-----------------------------|
| 1 | $P(X_1 = 1 \wedge X_2 = 1)$ | $P(X_1 = 1 \wedge X_2 = 2)$ | $P(X_1 = 1 \wedge X_2 = 3)$ |
| 2 | $P(X_1 = 2 \wedge X_2 = 1)$ | $P(X_1 = 2 \wedge X_2 = 2)$ | $P(X_1 = 2 \wedge X_2 = 3)$ |
| 3 | $P(X_1 = 3 \wedge X_2 = 1)$ | $P(X_1 = 3 \wedge X_2 = 2)$ | $P(X_1 = 3 \wedge X_2 = 3)$ |

Ze 100 000 měření jsme sestavili následující tabulku, kde je v každém okénku počet pozorování, při kterém byla odpovídající kombinace hodnot dopravního stavu:

| $x_1 \setminus x_2$ | 1 | 2 | 3 |
|---------------------|-------|-------|------|
| 1 | 28329 | 10712 | 5323 |
| 2 | 11725 | 15349 | 6002 |
| 3 | 9233 | 3769 | 9558 |

Podle statistické definice pravděpodobnosti a za předpokladu, že počet měření je dostatečný pro ustálení hodnot pravděpodobností, získáme tabulku pravděpodobnostní funkce náhodného vektoru "dopravní stupeň" jednoduše vydělením každého prvku tabulky celkovým počtem pozorování, tedy číslem 100 000. Dostaneme tabulku

| $x_1 \setminus x_2$ | 1 | 2 | 3 |
|---------------------|---------|---------|---------|
| 1 | 0.28329 | 0.10712 | 0.05323 |
| 2 | 0.11725 | 0.15349 | 0.06002 |
| 3 | 0.09233 | 0.03769 | 0.09558 |

kteřá reprezentuje sdruženou pravděpodobnostní funkci sledovaného stavu na vozovce.

Poznámky

1. Pro diskrétní náhodnou veličinu musí platit, že všechny prvky tabulky reprezentující sdruženou pf jsou nezáporné a součet hodnot v tabulce musí být jedna.
2. Pro spojitou náhodnou veličinu musí platit, že všechny její hodnoty jsou nezáporné a integrál přes celý její definiční obor musí být jedna.
3. Význam (spojité) sdružené hp je následující: pravděpodobnost, že X_1 je z intervalu (a, b) a zároveň X_2 z intervalu (c, d) je

$$P(X \in (a, b) \wedge (c, d)) = \int_a^b \int_c^d f(x_1, x_2) dx_2 dx_1.$$

V tomto smyslu lze hovořit o pravděpodobnosti v okolí daného bodu.

◁

• Marginální a podmíněné rozdělení

Jestliže se podaří sdružené rozdělení $f(x_1, x_2)$ rozložit na součin dvou rozdělení tak, že první z nich nezáleží na x_1 , tedy

$$f(x_1, x_2) = f_{X_2}(x_2) f_{X_1|X_2}(x_1|x_2),$$

pak funkci $f_{X_2}(x_2)$ nazveme marginálním rozdělením a funkci $f_{X_1|X_2}(x_1|x_2)$ nazveme podmíněným rozdělením.

• **Marginální rozdělení** vypovídá o náhodné veličině x_2 , když o náhodné veličině x_1 nic nevíme. Musíme tedy počítat se všemi možnými hodnotami x_1 .

• **Podmíněné rozdělení** udává rozdělení náhodné veličiny x_1 , jestliže hodnota náhodné veličiny x_2 je známa (byla změřena).

Poznámky

1. Často se podmíněné rozdělení udává obecně jako funkce pro různé hodnoty náhodné veličiny v podmínce. Je ale třeba mít na paměti, že podmíněné rozdělení popisuje náhodnou veličinu před podmínkou (tedy veličinu, jejíž hodnotu dosud neznáme) s tím, že náhodné veličiny za podmínkou nabyly daných hodnot (realizací). Tedy $f(x_2|x_1)$ je rozdělení náhodné veličiny X_2 pro $X_1 = x_1$, kde x_1 je nějaká realizace náhodné veličiny X_1 .

2. Marginální rozdělení dostaneme ze sdruženého tak, že integrujeme přes druhou náhodnou veličinu

$$\int_{X_1^*} f(x_1, x_2) dx_1 = \int_{X_1^*} f(x_2) f(x_1|x_2) dx_1 = f(x_2) \int_{X_1^*} f(x_1|x_2) dx_1 = f(x_2)$$

Protože (i) $f(x_2)$ je vzhledem k integraci přes x_1 konstanta, můžeme ji vytknout před integrál, (ii) funkce $f(x_1|x_2)$ je hustota v proměnné x_1 a integrál přes celý její definiční obor musí dát jedničku.

3. Podmíněné rozdělení dopočítáme. Protože $f(x_1, x_2) = f(x_1|x_2) f(x_2)$ bude

$$f(x_1|x_2) = \frac{f(x_1, x_2)}{f(x_2)}.$$

◁

Pro běžný výpočet marginálního a podmíněného rozdělení se používají vzorce uvedené v předchozí poznámce:

Marginální rozdělení:

$$f_{X_1}(x_1) = \int_{X_2^*} f(x_1, x_2) dx_2 \quad (1.1)$$

Podmíněná rozdělení:

$$f_{X_2|X_1}(x_2|x_1) = \frac{f(x_1, x_2)}{f_{X_1}(x_1)}. \quad (1.2)$$

Výjimku tvoří normální rozdělení, kde je možno využít metodu doplnění na čtverec v exponentu sdružené hp. Tento postup je ukázán v následujícím příkladu.

1.1.6 Příklad [výpočet marginální a podmíněné hp]

Uvažujme dvourozměrnou hp

$$f(x_1, x_2) = \frac{12}{7} (x_1^2 + x_1 x_2), \quad x_1 \in (0, 1), \quad x_2 \in (0, 1).$$

Napište marginální a podmíněné rozdělení.

V tomto případě lze vytknout x_1 a zjednodušeně psát

$$x_1^2 + x_1 x_2 = x_1 (x_1 + x_2).$$

Funkce x_1 bude tedy tvořit základ marginálního rozdělení a $x_1 + x_2$ bude základem pro podmíněné rozdělení. Obě ale musíme normalizovat tak, aby měly jednotkový integrál. U podmíněného rozdělení, které popisuje proměnnou x_2 , je x_1 v podmínce a má funkci konstanty. Proto se může vyskytovat i v normalizační konstantě podmíněného rozdělení. Začneme tedy s normalizací podmíněné hp. Protože výsledek jednotkového integrálu je

$$\int_0^1 (x_1 + x_2) dx_2 = \left[x_1 x_2 + \frac{1}{2} x_2^2 \right]_0^1 = x_1 + \frac{1}{2},$$

bude podmíněná hp

$$f(x_2|x_1) = \frac{1}{x_1 + \frac{1}{2}} (x_2 + x_1).$$

Marginální hp dopočítáme ze sdružené tak, aby platilo $f(x_1, x_2) = f_{X_1}(x_1) f_{X_2|X_1}(x_2|x_1)$, tedy

$$f(x_1, x_2) = \frac{12}{7} (x_1^2 + x_1 x_2) = \frac{12}{7} x_1 (x_1 + x_2)$$

Po vydělení podmíněnou hp dostáváme marginální ve tvaru

$$f_{X_1}(x_1) = \frac{12}{7} x_1 \left(x_1 + \frac{1}{2} \right).$$

Její integrál již bude jedna (ověřte).

1.1.7 Příklad [marginální a podmíněné rozdělení pro normální hp]

Úkolem je určit marginální a podmíněnou hp ze sdružené hp normálního rozdělení. Tento postup je specifický pro normální rozdělení, protože to má tvar exponenciály v jejímž exponentu je kvadratická funkce. Rozdělení na součin provedeme tak, že exponent rozdělíme na dva kvadráty tak, aby jeden závisel na vybrané proměnné a druhý ne. To je postup, kterému se říká doplnění na čtverec. Vybraná veličina pak bude popsána podmíněnou hp, ta druhá marginální.

Hp sdruženého normálního rozdělení je

$$f(x_1, x_2) = \frac{1}{2\pi} \exp \left\{ -\frac{1}{2} (5x_1^2 + x_2^2 - 4x_1 x_2 - 2x_1 + 1) \right\}.$$

(1) Nejprve budeme rozkládat $f(x_1, x_2) = f_{X_1}(x_1) f_{X_2|X_1}(x_2|x_1)$. To znamená, že z kvadratické formy chceme nejdříve "vytáhnout" všechna x_2 a zbytek nechat již jen jako funkci x_1 . Doplnujeme proto v proměnné x_2 :

$$\begin{aligned} 5x_1^2 + x_2^2 - 4x_1 x_2 - 2x_1 + 1 &= x_2^2 - 4x_1 x_2 + 4x_1^2 - 4x_1^2 + 5x_1^2 - 2x_1 + 1 = \\ &= (x_2 - 2x_1)^2 + x_1^2 - 2x_1 + 1 = (x_2 - 2x_1)^2 + (x_1 - 1)^2. \end{aligned}$$

Dosadíme do sdružené hp a dostaneme

$$\begin{aligned} f(x_1, x_2) &= \frac{1}{2\pi} \exp \left\{ -\frac{1}{2} [(x_2 - 2x_1)^2 + (x_1 - 1)^2] \right\} = \\ &= \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} (x_2 - 2x_1)^2 \right\} \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} (x_1 - 1)^2 \right\} = f_{X_2|X_1}(x_2|x_1) f_{X_1}(x_1) \end{aligned}$$

(2) Rozkládat můžeme ale také opačně $f(x_1, x_2) = f_{X_2}(x_2) f_{X_1|X_2}(x_1|x_2)$. Tentokrát budeme kvadratickou formu z exponentu sdruženého rozdělení doplňovat v proměnné x_1 .

$$\begin{aligned} 5x_1^2 + x_2^2 - 4x_1 x_2 - 2x_1 + 1 &= \\ &= 5 \left(x_1^2 - 2x_1 \frac{2}{5} \left(x_2 + \frac{1}{2} \right) + \left(\frac{2}{5} \left(x_2 + \frac{1}{2} \right) \right)^2 \right) - \frac{4}{5} \left(x_2 + \frac{1}{2} \right)^2 + x_2^2 + 1 = \\ &= 5 \left(x_1 - \frac{2}{5} \left(x_2 + \frac{1}{2} \right) \right)^2 + \frac{1}{5} (x_2 - 2)^2. \end{aligned}$$

První exponent bude patřit podmíněné hp, druhý marginální. Podle vzorce pro hp normálního rozdělení bude rozptyl podmíněné hp $\frac{1}{5}$ a rozptyl marginální bude 5. Dostáváme tak

$$f_{X_1|X_2}(x_1|x_2) = \frac{1}{\sqrt{2\pi \frac{1}{5}}} \exp \left\{ -\frac{1}{2} 5 \left(x_1 - \frac{2}{5} \left(x_2 + \frac{1}{2} \right) \right)^2 \right\}$$

a

$$f_{X_2}(x_2) = \frac{1}{\sqrt{2\pi \cdot 5}} \exp \left\{ -\frac{1}{2} \cdot \frac{1}{5} (x_2 - 2)^2 \right\}.$$

1.1.8 Příklad [podmíněné a marginální rozdělení pro diskrétní pf]

Určete marginální a podmíněnou hustotu diskrétního náhodného vektoru se sdruženou pf $f(x_1, x_2)$ danou následující tabulkou

| X_1, X_2 | 1 | 2 | 3 |
|------------|-----|-----|-----|
| 1 | 0.1 | 0.3 | 0.2 |
| 2 | 0.2 | 0.1 | 0.1 |

Marginální pravděpodobnosti spočteme jednoduše tak, že prvky tabulky sečteme (1) ve sloupcích pro marginálu v x_2 a (2) v řádcích pro marginálu v x_1

$f(x_1, x_2)$

| X_1, X_2 | 1 | 2 | 3 | $f(x_1)$ |
|------------|-----|-----|-----|----------|
| 1 | 0.1 | 0.3 | 0.2 | 0.6 |
| 2 | 0.2 | 0.1 | 0.1 | 0.4 |
| $f(x_2)$ | 0.3 | 0.4 | 0.3 | |

Podmíněná pravděpodobnostní funkce je rozdělením pro náhodnou veličinu před znaménkem podmínky jestliže náhodná veličina za znaménkem podmínky nabyla určité realizace, tj. např. $f(x_1|x_2 = 5)$. Zápis $f(x_1|x_2)$ znamená pravděpodobnostní funkci náhodné veličiny X_1 a je funkcí veličiny x_2 , která reprezentuje realizaci náhodné veličiny X_2 .

Pro libovolnou hodnotu náhodné veličiny v podmínce platí, $f_{X_1|X_2} = f/f_{X_2}$, tedy že podmíněná pf je podílem sdružené a marginální. Platí tedy:

| $f(x_1 x_2)$ | | | | $f(x_2 x_1)$ | | | |
|--------------|---------------|---------------|---------------|--------------|---------------|---------------|---------------|
| X_1, X_2 | 1 | 2 | 3 | X_1, X_2 | 1 | 2 | 3 |
| 1 | $\frac{1}{3}$ | $\frac{3}{4}$ | $\frac{2}{3}$ | 1 | $\frac{1}{6}$ | $\frac{1}{2}$ | $\frac{1}{3}$ |
| 2 | $\frac{2}{3}$ | $\frac{1}{4}$ | $\frac{1}{3}$ | 2 | $\frac{1}{2}$ | $\frac{1}{4}$ | $\frac{1}{4}$ |

kde např. pro první tabulku prvky v prvním a druhém řádku a prvním sloupci představují podmíněnou pf náhodné veličiny X_1 za podmínky, že náhodná veličina X_2 nabyla realizace 1. Tedy, jestliže $X_2 = 1$, pak $P(X_1 = 1) = \frac{1}{3}$ a $P(X_1 = 2) = \frac{2}{3}$.

Pro podmíněná a marginální rozdělení platí následující vztahy:
Řetězové pravidlo (rozklad sdružené hp)

$$f(x, y|z) = f(x|y, z) f(y|z). \quad (1.3)$$

Bayesův vzorec (prohození náhodné veličiny a podmínky)

$$f(x|y, z) = \frac{1}{k} f(y|x, z) f(x|z), \quad (1.4)$$

kde normalizační konstanta $k = f(y|z) = \int_{X^*} f(y|x, z) f(x|z) dx$ a z je realizace náhodné veličiny, která je společně v podmínce všech rozdělení.

• **Charakteristiky náhodné veličiny** jsou neúplným popisem náhodné veličiny, který vystihuje její určité vlastnosti (úroveň realizací, velikost odchylek realizací atd.). Pro naše účely, vzhledem k tomu, že nás bude zajímat především normální rozdělení, jsou nejdůležitější střední hodnoty a rozptyl (kovarianční matice).

- Střední hodnota

$$E[X] = \sum_{x \in X^*} x f(x) \quad \text{diskrétní} \quad E[X] = \int_{X^*} x f(x) dx \quad \text{spojitá}$$

- Rozptyl

$$D[X] = \sum_{x \in X^*} (x - E[X])^2 f(x) \quad \text{diskrétní} \quad D[X] = \int_{X^*} (x - E[X])^2 f(x) dx \quad \text{spojitý}$$

Pro náhodný vektor $X = [X_1, X_2]'$ platí (píšeme pro spojitou náhodnou veličinu - pro diskrétní je analogický):

- Střední hodnota

$$E[X] = \int_{X_1^* \times X_2^*} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} f(x_1, x_2) dx_1 dx_2 = \begin{bmatrix} E[X_1] \\ E[X_2] \end{bmatrix},$$

- Kovarianční matice

$$C[X] = \int_{X_1^* \times X_2^*} (X - E[X]) (X - E[X])' f(X) dX = \begin{bmatrix} D[X_1] & \text{cov}[X_1, X_2] \\ \text{cov}[X_1, X_2] & D[X_2] \end{bmatrix},$$

kde $\text{cov}[X_1, X_2]$ je kovariance X_1 a X_2 daná vzorcem

$$\text{cov}[X_1, X_2] = \int_{X_1^*} \int_{X_2^*} (x_1 - E[X_1]) (x_2 - E[X_2]) f(x_1, x_2) dx_2 dx_1.$$

- Podmíněná střední hodnota je dána pro náhodný vektor vzorcem

$$E[X_1|X_2] = \int_{X_1^*} x_1 f(x_1|x_2) dx_1.$$

Výsledkem je funkce proměnné x_2 tedy veličiny nacházející se v podmínce.

Poznámka

Všimněme si, že mezi střední hodnotou a podmíněnou střední hodnotou je dosti podstatný rozdíl. Řekneme-li „střední hodnota intenzity v jednom rameni křižovatky“, nemáme žádnou další informaci např. o tom, v kterou denní nebo dokonce noční hodinu se střední hodnota uvažuje. Střední hodnota se potom bere přes všechny možné hodnoty a konkrétním hodnotám intenzity může být dosti vzdálená. Naopak střední hodnota intenzity v daném rameni za podmínky, že intenzita v druhém rameni je větší než devadesát procent jejího maxima, je určena poměrně přesně. S velkou pravděpodobností budou totiž intenzity růst a klesat v obou ramenech přibližně stejně. Víme tedy, že tato podmíněná střední hodnota bude poměrně dobře vypovídat o maximálních hodnotách intenzity ve sledovaném rameni a bude jim tedy blízko.

Podmíněná střední hodnota je modelem pro popis náhodné veličiny v závislosti na jiných náhodných veličinách nacházejících se v podmínce. Nepodmíněná střední hodnota může sloužit jako počáteční odhad pro model (viz další kapitoly).

◁

Náhodný proces je funkce času, jejíž hodnoty jsou náhodné veličiny. Jedná se prakticky o jakoukoli veličinu vyvíjející se v čase, která je pod vlivem neurčitosti.

Náhodná posloupnost je náhodný proces, který měříme v diskrétním čase. Jedná se o posloupnost náhodných veličin, kde indexem posloupnosti je diskrétní čas. Každá náhodná veličina má své charakteristiky, které se také vyvíjejí v čase. Dostáváme tak posloupnost středních hodnot nebo rozptylů. Náhodné posloupnosti budeme využívat pro popis veličin měřených v diskrétním čase.

Náhodná posloupnost se spojitými hodnotami je náhodná posloupnost spojitých náhodných veličin. Tj. v každém okamžiku měření t dostaneme spojitou náhodnou veličinu, která může nabývat jakékoli hodnoty z reálné osy.

Náhodná posloupnost s diskrétními hodnotami je posloupnost diskrétních náhodných veličin, tj. takových náhodných veličin, které mohou nabývat konečného nebo spočetného množství hodnot.

1.1.9 Poznámka

Náhodné procesy se spojitým časem také existují a používají se. V našem výkladu se ale neobjeví a tak se s nimi nebudeme zabývat. ◁

1.2 Systém a veličiny s ním spojené

Předmětem našeho zájmu je vymezená část reality, na které měříme data a kterou chceme poznávat, abychom její chování mohli předpovídat, případně ovlivňovat. Sledovanou realitu nazveme **proces**. Soubor veličin spojených s procesem a jejich vzájemné vztahy nazveme **systém**.

1.2.1 Příklad [proces a systém]

Křižovatka je křížení dvou silnic. Silnice mohou být asfaltové, dlážděné nebo prašné. Mohou být jednosměrné nebo obousměrné, s jedním nebo více pruhy. Křižovatka může být řízená, nebo neřízená, atd. Tento úsek reality daný konkrétními silnicemi (případně chodníky) s příslušným vybavením (semafony, detektory, přechody pro chodce atd.) jsme nazvali *procesem*. Na tomto procesu měříme intenzitu provozu v

obou směrech a na obou ramenech. Tyto čtyři měřené intenzity i_1, i_2, i_3, i_4 spolu s veličinou t - denní čas a d - den a jejich vzájemnými vztahy určíme jako *systém*, který budeme matematicky popisovat a dále zkoumat.

Automobil je zařízení na přepravu osob a věcí. Je většinou z plechu a spousta nejrůznějších součástí, které jsou i u většiny různých typů aut dosti podobné. Tento souhrn plechu, součástek, benzínu a dalších přísad nazýváme *procesem*. Pokud nás bude zajímat spotřeba automobilu, vezmeme do systému ty veličiny, které se spotřebou souvisí. Jsou to vlastní spotřeba, rychlost, moment motoru, otáčky motoru, plyn, brzda, rychlostní stupeň a z okolí automobilu ještě úhel stoupání (klesání). Tyto veličiny a jejich vzájemné vztahy vyjadřují chování automobilu (vzhledem k jeho spotřebě) a tvoří *systém*. Jeho popisem je model spotřeby automobilu.

Pro definici systému jsou důležité veličiny. Protože ve velké většině jsou tyto veličiny pod vlivem neurčitosti (poruchy v procesu samém nebo chyby měření), uvažujeme tyto veličiny jako náhodné a pro jejich popis využíváme pravděpodobnostní pojmy - rozdělení, odhad apod.

Podle hodnot dělíme veličiny na **diskrétní** (mohou nabýt hodnoty z konečné nebo spočetné množiny) a **spojité** (jejich hodnoty jsou z celé reálné osy nebo nějaké její podmnožiny, např. z její nezáporné poloosy). Příklady diskrétních veličin jsou např. dopravní stupeň, barva na semaforu, typ nehody nebo rychlostní stupeň. Spojité veličiny jsou čas, rychlost, intenzita dopravního proudu. Intenzita je sice definována jako počet automobilů za hodinu, což je vlastně konečná množina, ale protože je velká, je lépe ji uvažovat jako veličinu spojitou.

1.2.2 Poznámka

Některé veličiny mají zvláštní postavení, např. rychlost automobilu. To je samozřejmě veličina spojitá. Počítáme-li ale její průměrnou hodnotu v určitém časovém intervalu, bereme rychlosti všech automobilů a dělíme jejich počtem. Jestliže automobilů ubývá (například s postupem noci), průměrná rychlost je přibližně stejná nebo spíše trochu roste a na jedinou dokonce nejde spočítat, protože žádný automobil neprojel. Navíc to, že jedno auto projelo velmi rychle, vůbec neznamená, že by auto jedoucí za ním mělo jet také rychle. Z hlediska vzájemné souvislosti měřených hodnot je tedy tato veličina velmi málo spojitá.

◁

Další hledisko, podle kterého lze veličiny dělit, je **uspořádanost časového vývoje jejich hodnot**. Zde jsou veličiny uspořádané (ordinální), kde lze určit, která hodnota veličiny je větší a která menší, a veličiny bez uspořádání (nominální), kde porovnání hodnot nemá smysl. Příkladem ordinální veličiny je hustota provozu, doba jízdy nebo spotřeba automobilu. Nominálními hodnotami jsou barvy na semaforu, denní doba (den, noc) nebo typ nehody (hmotná škoda, zranění, úmrtí).

1.2.3 Poznámka

Některé nominální hodnoty lze srovnat vůči hodnotám jiné ordinální veličiny. Tou jsou většinou peníze nebo její pozitivní či negativní vliv na zdraví, životní prostředí apod. Jako příklad lze uvést veličinu typ nehody. Vzhledem k jejímu dopadu lze její uspořádání definovat tak, jak jsme jej uvedli - od hmotné škody k úmrtí.

◁

Pro nás nejdůležitějším hlediskem pro třídění veličin je jejich **význam v systému**. Z tohoto hlediska rozlišujeme:

výstup - modelovaná veličina, kterou v daném okamžiku neznáme (realizuje se až na základě jiných veličin nebo našich akcí), ale dodatečně ji můžeme změřit.

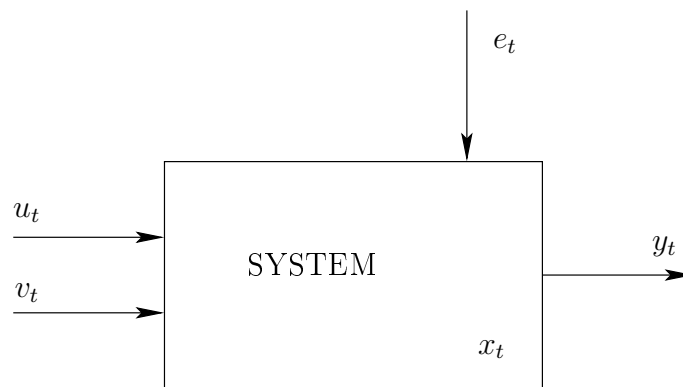
vstup (řízení) - veličina, kterou můžeme nastavovat a která má vliv na výstup.

externí vstup - veličina, kterou můžeme měřit, ale nemůžeme ji měnit a která má vliv na výstup.

stav - veličina, kterou nelze měřit a o které se dozvídáme jen prostřednictvím měřeného vstupu a výstupu.

šum - složka systému reprezentující neurčitost. Nelze ji ani měřit ani předpovídat.

Uvedené veličiny je možno znázornit na následujícím obrázku



kde y_t je výstup, u_t je vstup, v_t je externí vstup, x_t je stav a e_t je šum.

Ve schématu jsme uvedli všechny veličiny (náhodné veličiny) s indexem t , což je **diskrétní čas**. Označíme-li τ běžný spojité čas, pak vztah mezi spojitým a diskretním časem je $\tau = T \cdot t + \tau_0$, kde T je perioda vzorkování, tj. pevný časový interval, ve kterém se měří (vzorkují) všechny veličiny, τ_0 je počáteční čas a $t = 0, 1, 2, \dots$ je diskretní čas (označující pořadí periody od začátku měření). Veličiny jsou v každém časovém okamžiku náhodné a jsou indexovány časem - jsou to tedy **náhodné posloupnosti** (vektory náhodných veličin).

2 Spojitý model

Veličiny v dopravním systému jsou náhodné posloupnosti indexované diskretním časem t . V každém časovém okamžiku to jsou náhodné veličiny, po změření dostaneme realizace náhodné veličiny. Tyto náhodné veličiny mohou být buď spojité, nebo diskretní. V této kapitole se budeme zabývat případem, kdy výstupem systému je spojitá náhodná veličina indexovaná diskretním časem (náhodná posloupnost se spojitými hodnotami).

2.1 Princip stochastického modelu

Model je obrazem systému. Je to matematický popis závislosti modelované veličiny na jiných (vhodně vybraných) veličinách. Ve většině praktických případů funkci závislosti přímo neznáme,

a tak vztahy mezi veličinami popisujeme za pomoci parametrů. Hodnoty parametrů určujeme pomocí odhadu z měřených dat. Tato neznalost parametru a poruchy ve veličinách (ať už vznikající při samotném generování nebo v procesu měření) způsobují, že je takový model prakticky vždy pod vlivem neurčitosti.

2.1.1 Příklad [stochastický model]

Princip modelování lze demonstrovat na situaci, kdy byla náhle zablokována silnice a příjezdějí auta se stávají do kolony. Modelovanou veličinou y_t je délka kolony narůstající v diskrétním čase t . Tato délka závisí na intenzitě proudu I_t a na průměrné délce automobilu (včetně mezery mezi automobily) θ . Pro vývoj kolony v čase můžeme psát

$$y_t = y_{t-1} + \theta I_t, \quad y_0 = 0,$$

kde y_0 je počáteční délka kolony a 0 je okamžik vzniku blokace. Tento model lze uvažovat jako deterministický.

Jestliže jsme naměřili s periodou vzorkování 90 sec

| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------|---|---|---|---|---|---|----|---|---|----|
| I_t | 8 | 6 | 5 | 9 | 8 | 9 | 12 | 5 | 7 | 4 |

a uvažujeme-li délku vozidla 8 m, pak vývoj délky kolony v metrech bude

| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| y_t | 64 | 112 | 152 | 224 | 288 | 360 | 456 | 496 | 552 | 584 |

Je ale zřejmé, že tento deterministický model nemůže platit přesně. Počítali jsme s délkou auta 8 m, což je jen hrubý odhad, a rovněž jsme zaokrouhlovali počet automobilů v každé periodě měření intenzity I_t . Proto realističtější model je model stochastický, který můžeme vyjádřit následující rovnicí s přičteným šumem reprezentujícím neurčitosti a poruchy v systému

$$y_t = y_{t-1} + \theta I_t + e_t.$$

V tomto stochastickém modelu lze uvažovat i počáteční podmínku y_0 za náhodnou. Pokud jsme “v průměru” odhadovali dobře, bude mít šum přibližně nulovou střední hodnotu. V případě, že systém příliš nemění své stochastické vlastnosti, bude rozptyl šumu konstantní. Pokud dále “vysvětlující veličiny” modelu nesou o modelované veličině dostatek informací, budou jednotlivé složky náhodné posloupnosti navzájem nezávislé. Takové náhodné posloupnosti říkáme **bílý šum** (white noise).

Model, ke kterému jsme dospěli v minulém příkladě, obsahuje deterministickou (tady lineární) funkci počítající modelovanou veličinu y_t v závislosti na nezávislé veličině I_t , jakémsi parametru θ , který vyjadřuje konkrétní míru souvislosti mezi y_t a I_t . Stochastická část modelu je šum e_t , který (podobně jako v regresní analýze) “dorovnává” modelovanou veličinu i s poruchami proti jejím ideálním hodnotám (predikci).

Na tuto rovnici lze také pohlížet jako na transformaci náhodné veličiny e_t s rozdělením $f(e_t)$ na náhodnou veličinu y_t s podmíněným rozdělením $f(y_t|y_{t-1}, I_t, \theta)$. Abychom transformační rovnici mohli použít, musí být veličiny y_{t-1} , I_t , θ známé, tj. musí se vyskytnout v podmínce rozdělení modelované veličiny. Transformace potom představuje pouhé posunutí z nulové střední hodnoty šumu, na střední hodnotu modelované veličiny y_t , která je

$$E[y_t|y_{t-1}, I_t, \theta] = E[y_{t-1} + \theta I_t + e_t|y_{t-1}, I_t, \theta] = y_{t-1} + \theta I_t.$$

Rozptyl y_t je stejný jako rozptyl e_t . (Obojí si zkuste dokázat).

2.2 Lineární regresní model s normálním šumem

Nejčastěji používaným spojitým modelem je lineární regresní model s normálním rozdělením náhodné složky - šumu. Tento model má obecně rovnici

$$y_t = \psi_t' \Theta + e_t, \quad (2.1)$$

kde y_t je modelovaná veličina,

ψ_t je regresní vektor (vektor veličin, které vysvětlují y_t),

Θ je parametr modelu (zahrnuje regresní koeficienty θ a rozptyl šumu r),

e_t je šum s normálním rozdělením s nulovou střední hodnotou a konstantním rozptylem r .

Statický regresní model obsahuje v regresním vektoru jen vstupní veličiny (většinou měřené v současném časovém okamžiku), nikoli zpožděné hodnoty modelované veličiny. Jeho rovnici tedy můžeme psát ve tvaru

$$y_t = c_1 v_{1;t} + c_2 v_{2;t} + \dots + c_m v_{m;t} + k + e_t,$$

kde $\psi_t' = [v_{1;t}, v_{2;t}, \dots, v_{m;t}, 1]$ je regresní vektor a $\theta = [c_1, c_2, \dots, c_m, k]$ je vektor regresních koeficientů. Některé z veličin regresního vektoru mohou představovat řízení.

Typický dynamický regresní model popisující vstup a výstup systému je dán regresním vektorem ve tvaru

$$\psi_t' = [u_t, y_{t-1}, u_{t-1}, \dots, y_{t-n}, u_{t-n}, 1]$$

a jemu odpovídajícím vektorem parametrů

$$\theta' = [b_0, a_1, b_1, \dots, a_n, b_n, k].$$

Dosadíme-li do (2.1), dostaneme **model ve tvaru rovnice**

$$y_t = b_0 u_t + a_1 y_{t-1} + b_1 u_{t-1} + \dots + a_n y_{t-n} + b_n u_{t-n} + k + e_t. \quad (2.2)$$

Tato rovnice pak definuje podmíněnou hp modelované veličiny y_t takto:

Šum e_t má podle definice rozdělení

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

Podle transformační rovnice (2.2), která má Jakobián roven jedné, platí

$$e_t = y_t - \psi_t' \theta$$

a po dosazení do rozdělení šumu dostaneme

$$f(y_t | \psi_t, \Theta) = \frac{1}{\sqrt{2\pi r}} \exp \left\{ -\frac{1}{2r} (y_t - \psi_t' \theta)^2 \right\}, \quad (2.3)$$

což je hledaná podmíněná hp modelované veličiny - **model systému ve tvaru hustoty**.

Řádem regresního vektoru nazveme největší zpoždění modelované veličiny v regresním vektoru.

Statický model je model nultého řádu.

V modelu 1. řádu závisí modelovaná veličina na své minulé hodnotě. Zpoždění jiných veličin (řízení, externího vstupu) řád modelu neovlivní.

2.3 Diferenciální a diferenční rovnice ve spojitém modelu

2.3.1 Diskretizace diferenciální rovnice

Reálný čas plyne spojitě a většina dějů v reálném světě probíhá rovněž spojitě. Je proto dobře si uvědomit, že na pozadí našeho systému stojí spojitý proces a náš systém vzniká diskretizací tohoto procesu. Celou situaci budeme demonstrovat na velmi jednoduché (idealizované) soustavě popsané diferenciální rovnicí prvního řádu bez pravé strany - jedná se tedy o problém dozrívání počátečních podmínek. Spojité proměnné zde budeme značit velkými písmeny s argumentem času τ v závorce - např. $Y(\tau)$. Spojitý systém je tedy popsán rovnicí (Y' zde značí derivaci)

$$Y'(\tau) + aY(\tau) = 0, \quad Y(0) = y_0.$$

Naším úkolem je vyjádřit tuto diferenciální rovnici pomocí diferenční rovnice v diskrétním čase t s periodou vzorkování T ($\tau = tT$) tak, aby vzorky generované diferenční rovnicí ležely na funkci, která je řešením diferenciální rovnice a daných počátečních podmínek.

Postup odvození diferenční rovnice (diskretizace) je následující. Řešení diferenciální rovnice odhadneme ve známém exponenciálním tvaru a po dosazení počátečních podmínek dostaneme

$$Y(\tau) = y_0 \exp\{-a\tau\}.$$

Do řešení dosadíme diskrétní čas $\tau = tT$. Diskretizované řešení napíšeme pro čas $t + 1$

$$Y((t + 1)T) = y_0 \exp\{-a(t + 1)T\}.$$

Na levé straně je vzorek Y v čase $t + 1$, který značíme y_{t+1} . Na pravé straně upravíme exponenciálu

$$\exp\{-a(t + 1)T\} = \exp\{-aT\} \exp\{-atT\}.$$

První člen na pravé straně předchází rovnosti je vzorek Y v diskrétním čase t , druhý člen je konstanta. Dosadíme do diskretizovaného řešení a dostaneme

$$y_{t+1} = y_0 \exp\{-aT\} y_t \quad \text{nebo} \quad y_t = y_0 \exp\{-aT\} y_{t-1}$$

a tedy

$$y_t = Ay_{t-1},$$

kde $A = y_0 \exp\{-aT\}$. To je diferenční rovnice, která splňuje naše požadavky.

Podobným způsobem lze diskretizovat i diferenciální rovnice vyšších řádů, i když postup je poněkud zdouhavější.

2.3.2 Typy diferenciálních a diferenčních rovnic

Pro řešení diferenciální rovnice s konstantními koeficienty bez pravé strany je rozhodující řešení její charakteristické rovnice. Pro rovnice druhého řádu

$$y'' + a_1y' + a_0y = 0$$

je charakteristická rovnice

$$\lambda^2 + a_1\lambda + a_0 = 0.$$

Mohou nastat tři typy řešení charakteristické rovnice:

1. *dvě reálná řešení* λ_1 a λ_2 - diferenciální rovnice má řešení ve tvaru lineární kombinace dvou exponenciál

$$y = \alpha_1 \exp \{ \lambda_1 \tau \} + \alpha_2 \exp \{ \lambda_2 \tau \}$$

2. *jeden dvojnásobný kořen* λ - řešení je v tvaru

$$y = (\alpha_1 \tau + \alpha_2) \exp \{ \lambda \tau \}$$

3. *dva komplexně sdružené kořeny* $\lambda + j\omega$ a $\lambda - j\omega$ - řešení diferenciální rovnice je

$$y = \exp \{ \lambda \tau \} [\alpha_1 \sin (\omega \tau) + \alpha_2 \cos (\omega \tau)] .$$

Pro diferenciální rovnice druhého řádu s konstantními koeficienty bez pravé strany platí podobná analýza. Pro rovnici

$$y_{t+2} + a_1 y_{t+1} + a_0 y_t = 0$$

je opět rozhodující charakteristická rovnice

$$z^2 + a_1 z + a_0 = 0,$$

kde z je operátor jednokrokového předstihu $y_{t+1} = z y_t$.

Tato rovnice má podobné typy řešení (i je imaginární jednotka):

1. *dva reálné kořeny* λ_1 a λ_2 - řešení diferenciální rovnice je

$$y_t = \beta_1 \lambda_1^t + \beta_2 \lambda_2^t$$

2. *jeden dvojnásobný kořen* λ - řešení je

$$y_t = \beta_1 \lambda^t + \beta_2 t \lambda^t$$

3. *dva komplexně sdružené kořeny* $\lambda \pm i\omega$ - řešení je

$$y_t = (\lambda^2 + \omega^2)^{t/2} (\beta_1 \sin (\omega t) + \beta_2 \cos (\omega t))$$

2.4 Regresní model ve stavovém tvaru

Stav systému x_t je taková veličina (vektor veličin), který v sobě obsahuje informaci o celém dosavadním vývoji systému. Jestliže známe starý stav a aktuální řídicí veličiny, jsme schopni konstruovat pravděpodobnostní popis nového stavu. Model stavu má tedy vždy 1. řád. V lineárním případě je dán rovnicí

$$x_t = M x_{t-1} + N u_t + w_t,$$

kde M , N jsou matice odpovídajících rozměrů, w_t je bílý šum.

V řadě případů je výhodnější pracovat s modelem 1. řádu, i když mnoho-rozměrným, než s regresním modelem s řádem vyšším než jedna. Konstrukcí stavového modelu je celá řada. Takováto procedura je známa např. z přepočtu diferenciální (nebo diferenční) rovnice n -tého řádu na n rovnic, prvního řádu. Zde uvedeme tu nejjednodušší, která přímo vychází ze struktury regresního modelu.

Uvažujme obecně regresní model

$$y_t = b_0 u_t + a_1 y_{t-1} + \dots + a_n y_{t-n} + b_n u_{t-n} + k + e_t.$$

Definujeme vektor (stav) $x_t = [y_t, u_t, y_{t-1}, u_{t-1}, \dots, y_{t-n}, u_{t-n}, 1]'$ a pro něj s pomocí regresního modelu a identických rovnic sestavíme model v následujícím tvaru

$$x_t = Mx_{t-1} + Nu_t + \epsilon_t \quad (2.4)$$

se strukturou (z důvodu přehlednosti ji uvedeme pro regresní model řádu 2)

$$\underbrace{\begin{bmatrix} y_t \\ u_t \\ y_{t-1} \\ u_{t-1} \\ 1 \end{bmatrix}}_{x_t} = \underbrace{\begin{bmatrix} a_1 & b_1 & a_2 & b_2 & k \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} y_{t-1} \\ u_{t-1} \\ y_{t-2} \\ u_{t-2} \\ 1 \end{bmatrix}}_{x_{t-1}} + \underbrace{\begin{bmatrix} b_0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_B u_t + \underbrace{\begin{bmatrix} e_t \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{w_t},$$

kde první rovnice představuje regresní model a zbytek jsou identity.

Význam tohoto převodu ukážeme na příkladě.

2.4.1 Příklad [regresní model]

Pro regresní model 2. řádu se zanedbatelným šumem (tedy šumem s tak malým rozptylem, že šum lze zanedbat) určete hodnotu y_5 pro počáteční podmínky y_{-1}, y_0 a dané řízení u_{-1}, u_0, \dots, u_5 .

Teoreticky je úloha jednoduchá. Pro předpověď y_1 dosadíme do regresního vektoru regresního modelu a hodnotu výstupu prostě vypočteme při současném zanedbání šumu

$$y_1 = b_0 u_1 + a_1 y_0 + b_1 u_0 + a_2 y_{-1} + b_2 u_{-1} + k.$$

Při výpočtu y_2 použijeme známé hodnoty a také vypočtenou hodnotu výstupu y_1

$$\begin{aligned} y_2 &= b_0 u_2 + a_1 y_1 + b_1 u_1 + a_2 y_0 + b_2 u_0 + k = \\ &= b_0 u_2 + a_1 (b_0 u_1 + a_1 y_0 + b_1 u_0 + a_2 y_{-1} + b_2 u_{-1} + k) + b_1 u_1 + a_0 y_0 + b_0 u_0 + k = \\ &= b_0 u_2 + (a_1 b_0 + b_1) u_1 + (a_1 b_1 + b_0) u_0 + a_1 b_2 u_{-1} + (a_1^2 + a_0) y_0 + a_1 a_2 y_{-1} + (a_1 + 1) k. \end{aligned}$$

Tímhle způsobem lze pokračovat až do času $t = 5$. Není sice příliš složité postřehnout, jakým způsobem se vyvíjejí koeficienty u jednotlivých veličin, nicméně tato úloha je daleko snazší, použijeme-li stavový model

$$x_t = Mx_{t-1} + Nu_t,$$

kde jsme podle předpokladu zadání zanedbali šum. Platí

$$x_1 = Mx_0 + Nu_1,$$

kde $x_0 = [y_0, u_0, y_{-1}, u_{-1}, 1]'$. Dále

$$x_2 = Mx_1 + Nu_2 = M(Mx_0 + Nu_1) + Nu_2 = M^2 x_0 + MNu_1 + Nu_2,$$

$$x_3 = Mx_2 + Nu_3 = M(M^2 x_0 + MNu_1 + Nu_2) + Nu_3 = M^3 x_0 + M^2 Nu_1 + MNu_2 + Nu_3.$$

Ani nemusíme pokračovat a můžeme psát obecný vzorec:

$$x_t = M^t x_0 + \sum_{i=0}^{t-1} M^i Nu_{t-i}.$$

Požadované y_5 je prvním prvkem vektoru x_5 , tedy

$$x_5 = M^5 x_0 + \sum_{i=0}^4 M^i Nu_{5-i}, \quad \text{a} \quad y_5 = x_{5,1}.$$

3 Diskrétní a logistický model

3.1 Diskrétní model

Pokud mají všechny veličiny vstupující do modelu konečný počet hodnot, hovoříme o diskrétním modelu. Seřadíme-li tyto veličiny do vektoru (tzv. rozšířený regresní vektor $\Psi_t = [y_t, \psi_t']'$), pak realizací náhodných veličin v tomto vektoru dostaneme určitou konfiguraci jejich hodnot. Diskrétní systém má konečný počet takových konfigurací, které nazýváme módy. Příkladem je odbočení auta v T-křižovatce doprava, nebo doleva (2 módy) nebo kolona nebyla a není, nebyla a je, byla a není, byla a je (4 módy). Popis diskrétního systému proto můžeme provést velmi obecně, a to tak, že každé konfiguraci hodnot veličin v rozšířeném regresním vektoru přiřadíme její pravděpodobnost, tedy

$$f(y_t | \psi_t, \Theta) = \Theta_{y_t | \psi_t}, \quad (3.1)$$

kde multi-index (rozšířený regresní vektor s diskrétními hodnotami) $y_t | \psi_t = [y_t, \psi_{1;t}, \psi_{2;t}, \dots, \psi_{n_\psi;t}]$ je vektor indexů (hodnot diskrétních veličin) a znaménko “|” je použito jen formálně a ukazuje, které veličiny jsou modelovány a které se nachází v podmínce modelu.

Podobně jako u regresního modelu nazveme **řádem modelu** největší zpoždění modelované veličiny v regresním vektoru.

3.1.1 Příklad [diskrétní model]

Model budeme demonstrovat na jednoduchém, ale přesto dostatečně obecném příkladě tzv. “řízené koruny s pamětí”. Jedná se o systém s dvouhodnotovým výstupem s hodnotami 1 a 2 (např. rub a líc), který je závislý na minulém výstupu (např. po dopadu se koruna zašpiní a tím se rozváže pravděpodobnostní poměr jednotlivých stran) a na dvouhodnotovém řízení, rovněž s hodnotami 1 a 2 (např. způsob, jak se položí mince na ruku při hodu). Potom platí

$$y_t \in \{1, 2\}, \quad \psi_t = [u_t, y_{t-1}] \in \{1, 2\} \times \{1, 2\}, \quad \Psi_t = [y_t | u_t, y_{t-1}]$$

a model je možno zapsat ve formě tabulky

| $f(y_t u_t, y_{t-1})$ | y_t | |
|-------------------------|-------|-----|
| | 1 | 2 |
| $[u_t, y_{t-1}]$ | | |
| 1, 1 | 0.3 | 0.7 |
| 1, 2 | 0.8 | 0.2 |
| 2, 1 | 0.1 | 0.9 |
| 2, 2 | 0.2 | 0.8 |

kde $\Theta_{1|11} = 0.3$ a např. $\Theta_{2|21} = 0.9$.

Obecně pro parametry diskrétního modelu musí platit

$$\sum_{i \in y^*} \Theta_{i|jk} = 1, \quad \forall j, k.$$

Využití diskrétního modelu je možné zejména v situacích, kdy:

1. Modelovaná veličina je určitou klasifikací nějakého jevu - např. nehoda nastala nebo ne-nastala nebo nastala nehoda bez zranění, se zraněním, s úmrtím.
2. Nezávislé veličiny (v regresním vektoru) označují okolnosti, které doprovázejí naměřenou hodnotu modelované veličiny. Tyto veličiny mohou být nominální, například veličina denní doba s hodnotami: svítání, den, soumrak, noc.

Je potřeba si uvědomit, že čím více veličin obsahuje regresní vektor a čím více různých hodnot tyto veličiny mohou nabýt, tím větší je počet stavů takového systému. Abychom se o každém stavu něco dozvěděli, potřebujeme obrovský počet dat. Při tvorbě diskrétního modelu se proto snažíme do regresního vektoru vybrat jen podstatné veličiny (vzhledem k vysvětlení hodnot modelované veličiny) a u každé veličiny definovat co nejmenší počet hodnot. Pomoci může např. spojování hodnot: pro denní dobu místo jitra, den, soumrak, noc lze definovat světlo, šero, tma (pokud tomu nebrání konkrétní zadání úlohy).

3.2 Logistický model

Logistický model použijeme v případě, kdy modelovaná veličina je diskrétní (dvouhodnotová) $y_t \in \{0, 1\}$ a předpokládáme, že její hodnota stochasticky závisí na veličinách

$$\psi_t = [1, x_{1;t}, x_{2;t}, \dots, x_{n;t}]',$$

kteřé mohou být jak diskrétní, tak i spojité. Jednička v regresním vektoru představuje opět respektování konstanty modelu. V případě, kdy veličiny z vektoru ψ_t jsou všechny diskrétní, jedná se o diskrétní model, kterým jsme se zabývali v kapitole 3.1.

Logistický model má tvar

$$f(y_t | \psi_t, \Theta) = \frac{\exp(y_t z_t)}{1 + \exp(z_t)}, \quad (3.2)$$

kde je použita lineární regrese

$$z_t = \psi_t' \Theta + e_t. \quad (3.3)$$

Význam logistického modelu je v tom, že popisuje diskrétní veličinu y_t v závislosti na spojitě veličině z_t která je vypočtena na základě lineární regrese z veličin, které mohou být jak diskrétní, tak i spojité.

3.2.1 Příklad [logistický model]

Modelujeme závažnost nehody $y_t \in \{0, 1\}$ v závislosti na rychlosti automobilu $v_{1;t} \in R^+$ a osvětlení $v_{2;t} \in \{1, 2, 3\}$, kde 1 znamená světlo, 2 označuje šero a 3 představuje tmu. Lineární regrese (3.3) má rovnici

$$z_t = \theta_0 + \theta_1 v_{1;t} + \theta_2 v_{2;t} + e_t$$

a model (3.2) bude tvořen pravděpodobnostmi

$$\begin{aligned} P(y_t = 1 | \psi_t, \Theta) &= \frac{\exp(z_t)}{1 + \exp(z_t)}, \\ P(y_t = 0 | \psi_t, \Theta) &= \frac{1}{1 + \exp(z_t)}, \end{aligned}$$

kde první rovnici je pro $y_t = 1$ a druhá pro $y_t = 0$.

Princip fungování tohoto modelu je následující. Lineární regrese (3.3) mapuje regresní vektor ψ_t na veličinu z_t , které přirozeně nabývá hodnot z celé reálné osy. Funkce (3.2) potom transformuje tuto reálnou osu do intervalu $(0, 1)$, takže nakonec dostaneme hodnotu ve tvaru pravděpodobnosti. Hodnoty této pravděpodobnosti blízké 1 ukazují na hodnotu $y_t = 1$, hodnoty blízké 0 mluví pro $y_t = 0$.

Jiné vyjádření logistického modelu

Jiné možné vyjádření stejného logistického modelu je možno napsat pomocí funkce *logit*, která je definována

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right). \quad (3.4)$$

Dosadíme-li $p = P(y_t = 1|\psi_t, \Theta)$, dostaneme logistický model ve tvaru

$$\ln\left(\frac{P(y_t = 1|\psi_t, \Theta)}{1 - P(y_t = 1|\psi_t, \Theta)}\right) = \psi_t\Theta + e_t.$$

Samozřejmě pro $y_t = 0$ platí

$$P(y_t = 0|\psi_t, \Theta) = 1 - P(y_t = 1|\psi_t, \Theta)$$

Průběh funkce *logit* je vykreslen na Obrázku 3.1.

4 Odhad spojitého modelu

Model je matematickým popisem vybraných veličin sledovaného procesu. Tyto veličiny popisujeme stochasticky (pomocí hustot pravděpodobnosti) v závislosti na jiných vybraných veličinách většinou jako lineární vazby pomocí diferencních rovnic.

Při návrhu modelu řešíme dva kroky:

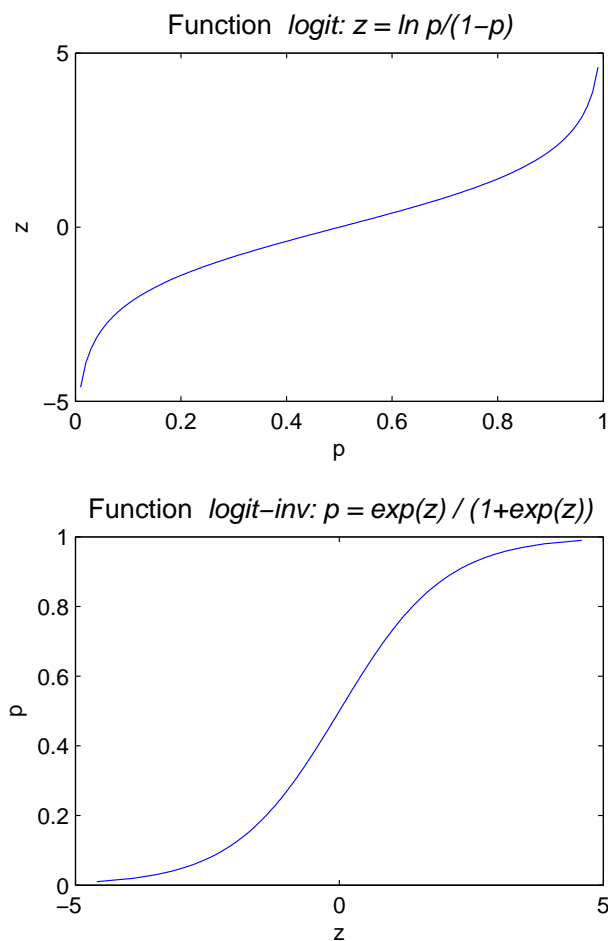
1. návrh struktury modelu, tj. výběr veličin a počtu kroků jejich zpoždění, na kterých modelovaná veličina závisí,
2. určení parametrů modelu, které vyjadřují konkrétní závislosti ve sledovaném procesu.

Zde se budeme zabývat druhým bodem - odhadem parametrů modelu.

4.1 Bayesovské odhadování

4.1.1 Parametry z hlediska bayesovství

V klasické statistice se odhadované parametry, ale i jiné odhadované veličiny, považují za náhodné veličiny. Jejich popisem je hustota pravděpodobnosti. Jestliže je pro konstrukci této hp



Obrázek 3.1: Funkce $z = \text{logit}(p)$

Funkce *logit* převádí proměnnou p , která má rozsah $(0, 1)$ na veličinu z , která má hodnoty z celé reálné osy. Toho se využije při logistické regresi tak, že hodnoty $z = \psi'\Theta \in R$ se transformují na pravděpodobnosti $p \in (0, 1)$. Transformace je patrná z obrázku: velké kladné hodnoty z se zobrazí blízko $p = 1$, hodnoty z kolem nuly padnou blízko $p = 0.5$ a velké záporné hodnoty z dávají hodnoty blízko $p = 0$.

Funkci *logit_inv* inverzní k funkci *logit* dostaneme otočením horního obrázku o 45° (viz dolní obrázek). Tato funkce převádí naopak reálná čísla na pravděpodobnosti.

použita jen předběžná (expertní) znalost, hovoříme o apriorním popisu. Jestliže jsou dále využita i měřená data, dostáváme aposteriorní popis.

V případě popisu parametrů se jedná o hp $f(\Theta)$, která se nazývá **apriorní hp** a která odráží prvotní znalosti o parametrech. V průběhu odhadování se měří data d_t v časech $t = 1, 2, \dots, T$, kde T je horizont intervalu odhadování. Informace z měřených dat se postupně využívá pro zpřesnění popisu parametrů a původní apriorní hp se vyvíjí na **aposteriorní hp**

$$f(\Theta) \xrightarrow{d_1} f(\Theta|d(1)) \xrightarrow{d_2} f(\Theta|d(2)) \xrightarrow{d_3} \dots \xrightarrow{d_T} f(\Theta|d(T))$$

Střední hodnota aposteriorní hp vypovídá o bodových odhadech parametrů (viz Přílohy 10.8), kovarianční matice o nepřesnosti odhadů. Přepočty hp popisující parametry se provádí na základě **Bayesova vztahu**.

4.1.2 Bayesův vztah

Vývoj hp parametrů, tj. postupné upřesňování hp parametrů podle informace přicházející z měřených dat d_1, d_2, \dots, d_t , se provádí podle Bayesova vzorce (viz Přílohy 1.4)

$$f(\Theta|d(t)) \propto f(y_t|\psi_t, \Theta) f(\Theta|d(t-1)), \quad (4.1)$$

který se počítá pro $t = 1, 2, \dots, T$ a startuje s tzv. apriorní hp $f(\Theta|d(0)) = f(\Theta)$, která odráží apriorní, expertní znalost.

Vztah (4.1) je možno zapsat také rovnou pro koncový čas T

$$f(\Theta|d(T)) = \prod_{t=1}^T f(y_t|\psi_t, \Theta) f(\Theta|d(0)) = L_T(\Theta) f(\Theta|d(0)),$$

kde

$$L_T(\Theta) = \prod_{t=1}^T f(y_t|\psi_t, \Theta) \quad (4.2)$$

je věrohodnostní funkce (likelihood).

Z uvedeného je patrné, že bayesovský odhad je vlastně klasický odhad maximální věrohodnosti korigovaný apriorní hp.

4.1.3 Reprodukovatelné parametrické vyjádření aposteriorní hp

Vztah (4.1) je rekurze pro funkce. Ta je prakticky nerealizovatelná, proto je třeba jednotlivé hp vyjádřit v konkrétním tvaru, který závisí na konečném počtu číselných charakteristik a tuto funkcionální rekurzi převést na rekurzi algebraickou pro charakteristiky rozdělení.²

²Např. přepočítat funkci $f(x)$ na $g(x)$ lze jenom tak, že ji přepočítáme $f(x) \rightarrow g(x)$ pro každý bod $x \in R$. Jestliže ale je $f(x) = \exp\{ax\}$ a $g(x) = \exp\{bx\}$, pak stačí přepočítat $a \rightarrow b$. Cela funkce už je dána svým předpisem.

Navíc je třeba parametrické vyjádření volit tak, aby při postupném odhadu v čase nevznikaly nové a nové charakteristiky, tj, aby se formální tvar hp parametrů reprodukoval. Jinak se tato hp velmi rychle stane tak složitou, že ji prakticky není možno počítat v rozumném čase.

Například bude-li mít model normální rozdělení, zvolíme apriorní hp také jako normální. Násobení normálních hp vede opět na normální rozdělení, které je určené svou střední hodnotou a rozptylem. Ty je možno počítat přímo ze středních hodnot a rozptylů modelu soustavy a apriorní hp. Dostáváme tak rekuzi na číslech, nikoli na funkcích. Pokud má model tuto vlastnost, řekneme, že při odhadu dostáváme **reprodukuje se aposteriorní hp**.

Obecně tento problém vyřešit nelze, ale pro případy, kterými se zde budeme zabývat, tj. pro regresní spojitý model s normálním rozdělením a pro diskrétní model s multinomiálním rozdělením, takové tzv. adjungované apriorní distribuce na parametrech existují ve formě inverzního Gauss-Wishartova a Dirichletova rozdělení. Budeme o nich podrobněji hovořit v následujících kapitolách.

4.1.4 Příklad [odhadování bez reprodukce aposteriorní hp]

V tomto příkladu budeme ilustrovat situaci, kdy odhadujeme s modelem, který nevede na reprodukuje se aposteriorní hp.

Uvažujme model $f(y|a) = ay^2 - 2ay + \frac{4a+3}{6}$ pro $y \in (0, 2)$ a parametrem $a \in (-\frac{3}{4}, \frac{3}{2})$. Protože známe rozsah parametru a a nemáme o něm žádnou apriorní informaci, budeme apriorní hp volit jako rovnoměrnou, tj. $f(a) = \frac{4}{9}$ na intervalu $a \in (-\frac{3}{4}, \frac{3}{2})$.

V prvním kroku odhadu naměříme y_1 a dostaneme

$$f(a|y_1) \propto f(y_1|a) f(a) = \left(ay_1^2 - 2ay_1 + \frac{4a+3}{6} \right) \frac{4}{9}.$$

V druhém kroku

$$f(a|y_1, y_2) \propto f(y_2|a) f(a|y_1) = \left(ay_2^2 - 2ay_2 + \frac{4a+3}{6} \right) \left(ay_1^2 - 2ay_1 + \frac{4a+3}{6} \right) \frac{4}{9}$$

a tak dále. Z toho je dobře vidět, že formální tvar aposteriorní hp je stále složitější, protože je v něm třeba si pamatovat stále složitější výrazy. Po prvním kroku pracujeme s y_1^2 a y_1 . Po druhém už to bude $y_1^2 y_2^2$, $y_1^2 y_2$, $y_1 y_2^2$, $y_1 y_2$, y_1^2 , y_2^2 , y_1 , y_2 a tak dále.

4.1.5 Příklad [Odhadování s reprodukcí aposteriorní hp]

Zde ukážeme odhad s modelem, který vede na reprodukuje se aposteriorní hp.

Vezmeme model $f(y_t|a) = \exp\{-ay_t\}$, $y_t \geq 0$, $a > 0$. Jako apriorní hp uvažujme $f(a) = \exp\{-ay_0\}$. Potom po prvním kroku odhadu máme

$$f(a|y_1) \propto f(y_1|a) f(a) = \exp\{-ay_1\} \exp\{-ay_0\} = \exp\{-a(y_1 + y_0)\},$$

po druhém

$$f(a|y_1, y_2) \propto \exp\{-a(y_2 + y_1 + y_0)\}$$

a obecně

$$f(a|y(k)) \propto \exp\left\{-a \sum_{i=0}^k y_i\right\}.$$

Je zřejmé, že když označíme statistiku odhadu $S_k = \sum_{i=0}^k y_i$, pak přepočítání statistiky v čase k je

$$S_k = S_{k-1} + y_k.$$

Formální tvar hp parametrů se nemění a vzorec pro přepočítání statistik má stále stejný tvar.

4.1.6 Výsledek odhadování

Výsledkem procedury odhadu je aposteriorní hp

$$f(\Theta|d(t)),$$

která dává úplný stochastický popis parametrů - tedy výčet všech možných hodnot a jejich pravděpodobnosti výskytu. Pokud je to možné (většinou z důvodů spočitatelnosti), mělo by se tam, kde máme neznámé parametry, počítat s touto celou distribucí.

Není-li možné využít v dalších výpočtech celou aposteriorní hp nebo potřebujeme-li jako odhady čísla, můžeme přejít k bodovým odhadům

$$\hat{\Theta}_t = E[\Theta|d(t)].$$

Obě tyto varianty využití procedury odhadování lze dobře ilustrovat v dalším odstavci Odhad výstupu systému.

4.1.7 Odhad výstupu

Předpověď výstupu je popsána prediktivní hp $f(y_t|\psi_t, d(t-1))$. Protože parametry Θ obecně neznáme, nesmí se vyskytovat v podmínce.³

Prediktivní hp dostaneme tak, že modelujeme obě neznámé veličiny y_t i Θ pomocí sdružené hp, kterou rozložíme podle řetězového pravidla a integrujeme přes Θ , abychom dostali popis jen pro výstup y_t

$$\begin{aligned} f(y_t|\psi_t, d(t-1)) &= \int_{\Theta^*} f(y_t, \Theta|\psi_t, d(t-1)) d\Theta = \\ &= \int_{\Theta^*} f(y_t|\psi_t, \Theta) f(\Theta|d(t-1)) d\Theta, \end{aligned} \quad (4.3)$$

kde chybějící veličiny v podmínkách vypadly z důvodu nezávislosti. Předpověď výstupu v čase t (jako náhodné veličiny) je tedy dána modelem $f(y_t|\psi_t, \Theta)$ a vyjádřením neznámého parametru pomocí aposteriorní hp parametrů $f(\Theta|d(t-1))$.

4.1.8 Poznámka

Dobře si všimněte, jak bayesovství zachází s neznámou veličinou (tady parametrem). Každého by napadlo: mám model a potřebuji do něho parametr. Udělám odhad a ten tam dosadím. To ale není optimální. Správné je použít úplné pravděpodobnosti tak jako v (4.3). Do modelu postupně dosazují všechny možné hodnoty parametrů a počítám vážený průměr - dosazení a sčítání dělá integrál, váhy jsou dány aposteriorní hp. <

³Jinak bychom tuto hp nemohli přímo použít - neměli bychom za Θ co dosadit. Parametry ale potřebujeme pro model. Proto musíme odhad Θ zabudovat do konstrukce prediktivní hp.

4.1.9 Bodový odhad výstupu

Bodový odhad parametru zkonstruovaný s pomocí aposteriorní hp je (viz Příloha 10.8) podmíněná střední hodnota parametru

$$\hat{\Theta}_t = E[\Theta|d(t)] = \int_{\Theta^*} \Theta f(\Theta|d(t)) d\Theta. \quad (4.4)$$

Tento bodový odhad už není úplný, ale jen částečný popis parametru (např. střední hodnota nic neříká o rozptylu). Pokud je ale aposteriorní hp dostatečně „štíhlá“, a to ona po správném odhadu je,⁴ je možno ji nahradit Diracovým impulzem $\delta(\Theta - \hat{\Theta}_t)$, kde $\delta(0)$ je jedna a jinde nula, tedy

$$f(\Theta|d(t-1)) \rightarrow \delta(\Theta - \hat{\Theta}_{t-1}). \quad (4.5)$$

Dosadíme za aposteriorní hp a dostaneme

$$f(y_t|\psi_t, d(t-1)) = \int_{\Theta^*} f(y_t|\psi_t, \Theta) \delta(\Theta - \hat{\Theta}_{t-1}) d\Theta = f(y_t|\psi_t, \hat{\Theta}_{t-1}). \quad (4.6)$$

Výsledek je právě ten, který bychom čekali. Jestliže máme model s neznámými parametry a spočteme bodové odhady parametrů, pak tyto bodové odhady použijeme místo neznámých parametrů. Nedostáváme optimální řešení, ale pro „štíhlou aposteriorní hp“ je to řešení často přijatelné.

4.2 Odhad parametrů regresního modelu

Odhad parametrů normálního regresního modelu na základě apriorní informace a dat $d(t)$ měřených průběžně na systému provedeme na základě Bayesova vzorce (4.1) s modelem (2.3) a vhodně zvolenou apriorní hp $f(\Theta|d(0)) = f(\Theta)$.

4.2.1 Model

Regresní model s regresním vektorem ψ_t , jemu odpovídajícím vektorem regresních koeficientů θ a normálním šumem s rozptylem r má tvar

$$f(y_t|\psi_t, \Theta) = \frac{1}{\sqrt{2\pi}} r^{-0.5} \exp\left\{-\frac{1}{2r} (y_t - \psi_t' \theta)^2\right\}.$$

Pro účely odhadu je výhodné tuto hp (výraz v exponentu) upravit do následujícího tvaru (viz Příloha 10.6, rovnice (10.9))

$$f(y_t|\psi_t, \Theta) = \frac{1}{\sqrt{2\pi}} r^{-0.5} \exp\left\{-\frac{1}{2r} [-1 \ \theta'] D_t \begin{bmatrix} -1 \\ \theta \end{bmatrix}\right\}, \quad (4.7)$$

kde $D_t = \begin{bmatrix} y_t \\ \psi_t \end{bmatrix} \begin{bmatrix} y_t & \psi_t' \end{bmatrix}$ je tzv. datová matice.

⁴Při odhadu získáváme informaci, tím klesá neurčitost a s ní i rozptyl odhadu.

Dále v Příloze 10.7 v rovnici (10.10) je uveden tvar aposteriorní hp, který odpovídá normálnímu regresnímu modelu (jedná se o tzv. konjugovanou aposteriorní hp k normálnímu regresnímu modelu, tj. takový popis parametrů, jehož tvar se při odhadu podle Bayesova vzorce reprodukuje)

$$f(\Theta|d(\tau)) \propto r^{-0.5\kappa_\tau} \exp\left\{-\frac{1}{2r}[-1 \ \theta'] V_\tau \begin{bmatrix} -1 \\ \theta \end{bmatrix}\right\}, \quad (4.8)$$

kde V_τ a κ_τ jsou statistiky odhadu. Pro časový okamžik t vzorec s $\tau = t$ udává aposteriorní hp, pro $\tau = t - 1$ je to apriorní hp.

4.2.2 Rekurze pro statistiku

Dosazením do Bayesova vzorce (4.1) dostaneme

$$\begin{aligned} & \underbrace{r^{-0.5\kappa_t} \exp\left\{-\frac{1}{2r}[-1 \ \theta'] V_t \begin{bmatrix} -1 \\ \theta \end{bmatrix}\right\}}_{\text{aposteriorní hp}} \propto \\ & \underbrace{r^{-0.5} \exp\left\{-\frac{1}{2r}[-1 \ \theta'] D_t \begin{bmatrix} -1 \\ \theta \end{bmatrix}\right\}}_{\text{model}} \underbrace{r^{-0.5\kappa_{t-1}} \exp\left\{-\frac{1}{2r}[-1 \ \theta'] V_{t-1} \begin{bmatrix} -1 \\ \theta \end{bmatrix}\right\}}_{\text{apriorní hp}}, \end{aligned}$$

odkud porovnáním obou stran tohoto vztahu dostaneme rovnice pro přepočítání statistik

$$V_t = V_{t-1} + D_t, \quad (4.9)$$

$$\kappa_t = \kappa_{t-1} + 1. \quad (4.10)$$

Uvedené statistiky se nazývají: V - rozšířená informační matice, κ - počítadlo vzorků a datová matice D_t . je

$$D_t = \begin{bmatrix} y_t \\ \psi_t \end{bmatrix} \begin{bmatrix} y_t & \psi_t' \end{bmatrix},$$

4.2.3 Algoritmus odhadu

Postup odhadu parametrů je následující:

1. Konstrukce apriorní hp $f(\Theta|d(0))$. Obecně to není jednoduchá záležitost. Jedná se o převod předpokladů nebo požadavků týkajících se parametrů Θ (jako třeba Θ má nezáporné prvky, nebo menší než nějaká horní hranice apod.) do apriorních statistik konstruované apriorní hp.

Výstupem jsou apriorní statistiky V_0 a κ_0 a z nich zkonstruovaná apriorní hp podle (4.8) s dosazenými apriorními statistikami.

2. Měření dat d_1, d_2, \dots kde $d = \{y, u\}$ a průběžný přepočítání statistik podle vztahů (4.9) a (4.10) tj.

$$V_t = V_{t-1} + D_t \text{ a } \kappa_t = \kappa_{t-1} + 1.$$

Výstupem jsou aposteriorní statistiky V_t a κ_t .

3. Konstrukce aposteriorní hp $f(\Theta|d(t))$ dosazením do vztahu (4.8) s $\tau = t$ a statistikami V_t, κ_t .
Výstupem je aposteriorní hp $f(\Theta|d(t))$.
4. Výpočet bodových odhadů $\hat{\Theta}_t$ (jsou-li potřeba). Bodové odhady jsou dány jako podmíněná střední hodnota (viz Příloha 10.8)

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

V Příloze 10.9, v rovnicích (10.14) a (10.16) je ukázáno, že platí

$$\hat{\Theta}_t = V_\psi^{-1} V_{y\psi}, \quad \hat{r}_t = \frac{V_y - V_{y\psi}' V_\psi^{-1} V_{y\psi}}{\kappa_t}. \quad (4.11)$$

Matice V_ψ a vektor $V_{y,\psi}$ dostaneme rozdělením informační matice V_t na submatice

$$V_t = \begin{bmatrix} V_y & V_{y\psi}' \\ V_{y\psi} & V_\psi \end{bmatrix}, \quad (4.12)$$

kde V_y je skalár, $V_{y\psi}$ je sloupcový vektor, $V_{y\psi}'$ je řádkový vektor a V_ψ je čtvercová matice a κ_t je počítadlo datových vzorků, pro které platí $\kappa_\tau = \kappa_{\tau-1} + 1$, $\tau = 1, 2 \dots t$ s počáteční hodnotou κ_0 (apriorní statistika).

Výsledkem uvedeného algoritmu je:

- konstrukce aposteriorní hp podle bodu 3,
- bodové odhady parametrů (4.11).

4.2.4 Bodový odhad výstupu s bodovým odhadem parametru

Střední hodnotu výstupu y_t (tedy jeho optimální odhad) dostaneme přímo z modelu takto

$$\hat{y}_t = E[y_t | \psi_t, d(t-1)] = E[\psi_t' \hat{\theta}_t + e_t] = \psi_t' \hat{\theta}_t.$$

To znamená, že do modelu bez šumu⁵ dosadíme data a odhady parametrů a jednoduše spočteme výstup.

Odhadování parametrů regresního modelu a jeho výstupu budeme ilustrovat na příkladech.

4.2.5 Příklad [odhad regresního modelu]

Simulujte spojitý dynamický systém popsaný regresním modelem 2. řádu s regresním vektorem

$$\psi_t = [u_t, y_{t-1}, u_{t-1}, y_{t-2}, u_{t-2}, 1]'$$

⁵Odhadem je střední hodnota. Střední hodnota šumu je nula, takže predikce se provádí jakoby s modelem bez šumu.

jemu odpovídajícími regresními koeficienty

$$\theta = [1, 0.6, 0.5, -0.2, -0.3, 0.1]'$$

a rozptylem šumu $r = 0.01$. Odhadněte parametry $\Theta = \{\theta, r\}$ tohoto systému.

Simulace se provádí s pomocí modelu ($\sigma = \sqrt{r} = 0.1$)

$$y_t = \psi_t' \theta + \sigma e_t = u_t + 0.6y_{t-1} + 0.5u_{t-1} - 0.2y_{t-2} - 0.3u_{t-2} + 0.1 + 0.1e_t,$$

kde

$e_t \sim N_{e_t}(0, 1)$ je realizace standardního normálního šumu, ψ_t je regresní vektor ze zadání úlohy, θ jsou regresní koeficienty a σ je směrodatná odchylka šumu.

Pro odhad vytváříme rozšířený regresní vektor

$$\Psi_t = \begin{bmatrix} y_t \\ \psi_t' \end{bmatrix}' \quad (4.13)$$

pro $t = 1, 2, \dots, n_t$, se kterým přepočítáváme informační matici a počítadlo

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

s počátečními (apriorními) hodnotami V_0 a κ_0 .

Bodové odhady určíme rozkladem informační matice podle (10.11)

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

a dále podle vzorců

$$\hat{\theta}_t = V_{\psi}^{-1} V_{y\psi} \quad \text{a} \quad \hat{r}_t = \frac{V_y - V_{y\psi}' V_{\psi}^{-1} V_{y\psi}}{\kappa_t}. \quad (4.14)$$

Podrobné řešení je uvedeno v následujícím programu

```

clc , clear all
// Simulation and estimation of second order
// regression model

nt=100; // number of data

// SIMULATION
th=[1 .6 .5 -.2 -.3 .1]'; // regression coefficients
r=.01; // noise variance
s=sqrt(r); // standard deviation of noise
y=zeros(1,nt); // zero initial conditions + declar.
u=ones(1,nt)+rand(1,nt); // input declaration
// time loop for simulation
for t=3:nt
    psi=[u(t) y(t-1) u(t-1) y(t-2) u(t-2) 1]';
    y(t)=psi'*th + s*randn;
end

// ESTIMATION
V=zeros(7); // initial statistics

```

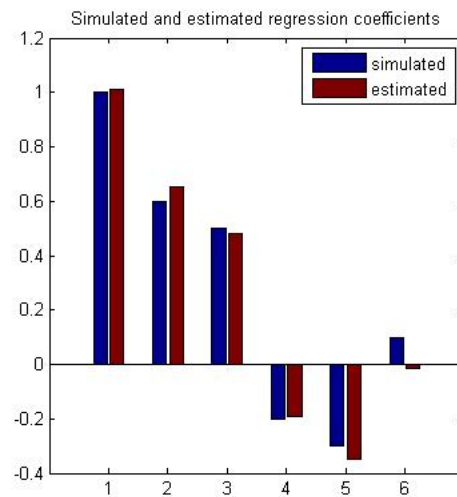
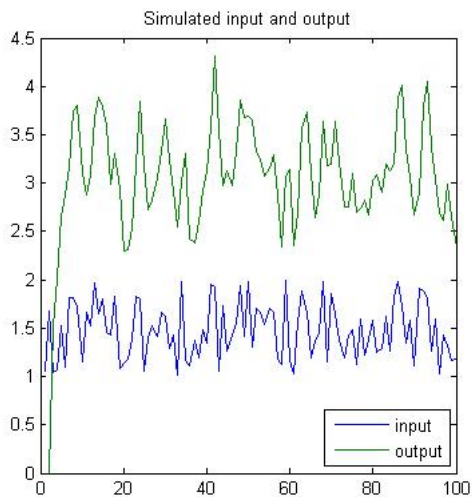
```

// time loop for estimation
for t=3:nt
    Psi=[y(t) u(t) y(t-1) u(t-1) y(t-2) u(t-2) 1]';
    V=V+Psi*Psi';
end
Vy=V(1,1);
Vyp=V(2:end,1);
Vp=V(2:end,2:end);
Eth=inv(Vp)*Vyp; // point est. of regr. coeff.
Cth=(Vy-Vyp'*inv(Vp)*Vyp)/nt; // point est. of noise var.

// print
Simulated_noise_variance=r
Estimated_noise_variance=Cth

```

Výsledky programu jsou na obrázcích



4.2.6 Příklad [bodový odhad regresního modelu]

Alternativní postup při odhadu parametrů normálního regresního modelu vycházející z metody nejmenších čtverců (který je ale s předchozím způsobem ekvivalentní) je následující.

Do rovnice regresního modelu (z předchozího příkladu)

$$y_t = b_0 u_t + a_1 y_{t-1} + b_1 u_{t-1} + a_2 y_{t-2} + b_2 u_{t-2} + k + e_t$$

postupně dosazujeme měřená data a rovnice pro $t = 1, 2, \dots, N$ píšeme pod sebe

$$\begin{aligned}
 y_1 &= b_0 u_1 + a_1 y_0 + b_1 u_0 + a_2 y_{-1} + b_2 u_{-1} + k + e_1 \\
 y_2 &= b_0 u_2 + a_1 y_1 + b_1 u_1 + a_2 y_0 + b_2 u_0 + k + e_2 \\
 &\dots
 \end{aligned}$$

$$y_N = b_0 u_N + a_1 y_{N-1} + b_1 u_{N-1} + a_2 y_{N-2} + b_2 u_{N-2} + k + e_N.$$

Vytvořenou soustavu rovnic zapíšeme maticově

$$Y = \Psi\theta + E,$$

kde Y je vektor modelované veličiny y_t , Ψ je matice regresních vektorů v řádcích, θ je vektor parametrů v pořadí odpovídajícím pořadí veličiny v regresním vektoru a E je vektor šumů.

Pro uvažovaný model bude mít soustava tvar

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix} = \begin{bmatrix} u_1 & y_0 & u_0 & y_{-1} & u_{-1} & 1 \\ u_2 & y_1 & u_1 & y_0 & u_0 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ u_N & y_{N-1} & u_{N-1} & y_{N-2} & u_{N-2} & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ a_1 \\ b_1 \\ a_2 \\ b_2 \\ k \end{bmatrix} + \begin{bmatrix} e_1 \\ e_1 \\ \dots \\ e_N \end{bmatrix}$$

Bodový odhad parametrů je

$$\hat{\theta} = (\Psi'\Psi)^{-1} \Psi'Y,$$

odhad rozptylu šumu je

$$\hat{r} = Y' (Y - \hat{\theta}\Psi) = Y'E_p,$$

kde $E_p = Y - \hat{\theta}\Psi$ je vektor chyb predikce $Y_p = \hat{\theta}\Psi$.

4.2.7 Poznámka

Z porovnání obou uvedených příkladů plyne

$$Y'Y = V_y, \quad Y'\Psi = V_{y\psi} \quad \text{a} \quad \Psi'\Psi = V_\psi.$$

<

5 Odhad diskrétního a logistického modelu

5.1 Odhad parametrů diskrétního modelu

Diskrétní model popisuje systém, v němž jsou všechny veličiny diskrétní. Model je reprezentován tabulkou pravděpodobností. Model i úlohy s ním spojené jsou velice jednoduché, ale

- práce s tabulkami může být poněkud nezvyklá,
- pokud mají veličiny větší počet různých hodnot, bude tabulka modelu neúnosně velká. V tom případě je lépe přejít na logistickou regresi.

5.1.1 Model a jeho součinnový tvar

Model (3.1) obsahuje parametry $\Theta_{y|\psi}$ (pravděpodobnosti jednotlivých konfigurací rozšířeného regresního vektoru $\Psi_t = [y_t, \psi_t']$), které jsou v obecném případě neznámé, a tak je třeba je odhadovat z měřených dat. Pro odhad použijeme Bayesův vzorec (4.1), diskrétní model (3.1) a vhodně zvolený apriorní model parametrů $f(\Theta|d(0)) = f(\Theta)$. Dle odstavce 4.1.3 o analytickém tvaru hp parametrů a o reprodukovatelnosti její struktury je třeba tuto hp parametrů zvolit v analytickém tvaru, a to takovém, aby se při postupném násobení modelem soustavy jeho analytický tvar reprodukoval. Za tímto účelem přepíšeme model soustavy (3.1) formálně do tzv. **součinnového tvaru**

$$f(y_t|\psi_t, \Theta) = \prod_{y|\psi \in \Psi^*} \Theta_{y|\psi}^{\delta(y|\psi, y_t|\psi_t)},$$

kde $y|\psi$ je multiindex (tj. vektorový index), který může nabývat všech možných konfigurací hodnot přípustných pro jednotlivé veličiny v něm obsažených; Ψ^* je množina všech takových konfigurací; symbol $\delta(y|\psi, y_t|\psi_t)$ je Kroneckerova funkce, která se rovná jedné, když platí $y|\psi = y_t|\psi_t$ (tj. $y = y_t$ a $\psi_1 = \psi_{1;t}$ až $\psi_{n_\psi} = \psi_{n_\psi;t}$), v ostatních případech je rovna nule.

5.1.2 Příklad [model falešné mince]

Pro model falešné koruny ($y = 1$ je líc, $y = 2$ je rub) lze zapsat jednotlivé pravděpodobnosti ve tvaru $f(y = 1) = p_1$ a $f(y = 2) = p_2$, kde $p_1, p_2 \geq 0$ a $p_1 + p_2 = 1$. Tento model můžeme zapsat následovně:

$$f(y) = p_1^{\delta(y,1)} p_2^{\delta(y,2)} = \prod_{i=1}^2 p_i^{\delta(y,i)}.$$

5.1.3 Statistika

Přepis do součinnového tvaru je čistě formální záležitost. Pro všechna $y|\psi \neq y_t|\psi_t$ dostáváme $\Theta^0 = 1$ a jen pro $y|\psi = y_t|\psi_t$ je $\Theta^1 = \Theta_{y_t|\psi_t}$. Nicméně je pro nás součinnový tvar návodem, jak volit apriorní model parametrů. Ten píšeme ve tvaru Dirichletova rozdělení - viz Přílohy 10.5

$$f(\Theta|d(0)) = \prod_{y|\psi \in \Psi^*} \Theta_{y|\psi}^{\nu_{y|\psi;0}},$$

kde $\nu_{y|\psi;0}$ je apriorní statistika (pro čas $t = 0$) pro odhad parametru Θ . Tato statistika je matice stejných rozměrů jako Θ . Podobně jako model ji můžeme reprezentovat pomocí tabulky. Počáteční statistika má tedy následující tvar

$$\nu_{y|\psi;0} \tag{5.1}$$

| $[u_0, y_{-1}]$ | $y_0 = 1$ | $y_0 = 2$ |
|-----------------|--------------|--------------|
| 1, 1 | $\nu_{1 11}$ | $\nu_{2 11}$ |
| 1, 2 | $\nu_{1 12}$ | $\nu_{2 12}$ |
| 2, 1 | $\nu_{1 21}$ | $\nu_{2 21}$ |
| 2, 2 | $\nu_{1 22}$ | $\nu_{2 22}$ |

5.1.4 Přepočítání statistiky

Předchozí odvozené vztahy dosadíme do Bayesova vzorce (4.1) a dostaneme pro $t = 1$

$$\begin{aligned}
 f(\Theta|d(1)) &\propto \underbrace{\prod_{y|\psi \in \Psi^*} \Theta_{y|\psi}^{\delta(y|\psi, y_1|\psi_1)}}_{\text{model}} \underbrace{\prod_{y|\psi \in \Psi^*} \Theta_{y|\psi}^{\nu_{y|\psi;0}}}_{\text{apriorní}} = \\
 &= \underbrace{\prod_{y|\psi \in \Psi^*} \Theta_{y|\psi}^{\delta(y|\psi, y_1|\psi_1) + \nu_{y|\psi;0}}}_{\text{aposteriorní}} = \prod_{y|\psi \in \Psi^*} \Theta_{y|\psi}^{\overbrace{\nu_{y|\psi;1}}^{\text{nová statistika}}},
 \end{aligned}$$

kde

$$\nu_{y|\psi;1} = \delta(y|\psi, y_1|\psi_1) + \nu_{y|\psi;0}$$

je statistika modelu parametrů $f(\Theta|d(1))$ pro čas $t = 1$.

Dále měříme data pro $t = 2, 3, \dots, N$ a přepočítáváme statistiku ν podle obecného vzorce

$$\nu_{y|\psi;t} = \delta(y|\psi, y_t|\psi_t) + \nu_{y|\psi;t-1}. \quad (5.2)$$

Význam přepočtu statistiky ne následující: Po každém změření dat přičteme jedničku do toho políčka statistiky (viz (5.1)), které odpovídá dané konfiguraci hodnot rozšířeného regresního vektoru $y_t|\psi_t$. Každé políčko tedy obsahuje počet, kolikrát příslušná konfigurace hodnot dosud nastala.

Aposteriorní hp parametrů se statistikou ν_t má tvar

$$f(\Theta|d(t)) \propto \prod_{y|\psi \in \Psi^*} \Theta_{y|\psi}^{\nu_{y|\psi;t}} \quad (5.3)$$

5.1.5 Bodový odhad parametrů

Bodový odhad parametru Θ je (viz Přílohy 10.10)

$$\hat{\Theta}_{y|\psi;t} = \frac{\nu_{y|\psi;t}}{\sum_{y \in y^*} \nu_{y|\psi;t}}, \quad (5.4)$$

což zcela odpovídá statistické definici pravděpodobnosti jevu označeného indexem $y|\psi$. Pro každou konfiguraci regresního vektoru ψ je v čitateli počet případů, kolikrát nastala daná hodnota y v rámci tohoto regresního vektoru (počet příznivých pokusů) a ve jmenovateli je celkový počet pokusů.

Tabulka bodových odhadů má stejný tvar jako samotný parametr nebo statistika

$$\hat{\Theta}_t \quad (5.5)$$

| $[u_t, y_{t-1}]$ | $y_t = 1$ | $y_t = 2$ |
|------------------|-------------------------|-------------------------|
| 1, 1 | $\hat{\Theta}_{1 11;t}$ | $\hat{\Theta}_{2 11;t}$ |
| 1, 2 | $\hat{\Theta}_{1 12;t}$ | $\hat{\Theta}_{2 12;t}$ |
| 2, 1 | $\hat{\Theta}_{1 21;t}$ | $\hat{\Theta}_{2 21;t}$ |
| 2, 2 | $\hat{\Theta}_{1 22;t}$ | $\hat{\Theta}_{2 22;t}$ |

a prvky této tabulky se počítají podle vzorce (5.4).

5.1.6 Bodový odhad výstupu

Podobně jako v případě spjitého regresního modelu i pro diskretní model s neznámými parametry je možno neznámé parametry nahradit jejich bodovými odhady - z tabulky (viz (5.1)) vezmeme řádek odpovídající kombinaci hodnot v regresním vektoru ψ_t a v něm jako odhad \hat{y}_t výstupu y_t vezmeme tu hodnotu, která má větší pravděpodobnost.

5.1.7 Příklad [odhad s diskretním modelem]

V dopravní oblasti byly sledovány nehody a byly rozlišeny podle závažnosti na lehké (jen hmotná škoda) a těžké (vážné zranění nebo smrt). Nehody jako modelovanou veličinu označíme y_t (kde t nyní označuje pořadí nehody, nikoli čas) s hodnotami $y_t = 1$ - lehká nehoda a $y_t = 2$ - těžká nehoda. Předpokládáme, že na typ nehody mají hlavní vliv následující veličiny: ψ_1 - rychlost (1 normální, 2 velká), ψ_2 - počasí (1 sucho, 2 mokro) a ψ_3 - osvětlení (1 světlo, 2 tma).

Po dobu jednoho roku jsme měřili data a získali 18 následujících záznamů.

| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|--------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| y_t | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 1 | 1 | 1 | 2 | 1 | 1 |
| $\psi_{1;t}$ | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 2 | 1 |
| $\psi_{2;t}$ | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 1 | 2 |
| $\psi_{3;t}$ | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 1 |

Model pro popis nehod podle (3.1) s využitím měřených dat bude mít tvar $f(y_t|v_t)$ s následující tabulkou

$\Theta_{y|\psi}$

| $[v_{1;t}, v_{2;t}, v_{3;t}]$ | $y_t = 1$ | $y_t = 2$ |
|-------------------------------|------------------|------------------|
| [1 1 1] | $\Theta_{1 111}$ | $\Theta_{2 111}$ |
| [1 1 2] | $\Theta_{1 112}$ | $\Theta_{2 112}$ |
| [1 2 1] | $\Theta_{1 121}$ | $\Theta_{2 121}$ |
| [1 2 2] | $\Theta_{1 122}$ | $\Theta_{2 122}$ |
| [2 1 1] | $\Theta_{1 211}$ | $\Theta_{2 211}$ |
| [2 1 2] | $\Theta_{1 212}$ | $\Theta_{2 212}$ |
| [2 2 1] | $\Theta_{1 221}$ | $\Theta_{2 221}$ |
| [2 2 2] | $\Theta_{1 222}$ | $\Theta_{2 222}$ |

Statistika odhadu podle (5.1) je reprezentována stejnou tabulkou. Aktualizace statistiky začíná s apriorní tabulkou a dále pokračuje podle vzorce (5.2), který říká: pro každé $t = 1, 2, \dots, n_t$ podle hodnot příslušných veličin y_t a $\psi_t = [v_{1;t}, v_{2;t}, v_{3;t}]'$ najděte v tabulce odpovídající políčko a k němu přičtěte jedničku.

V našem příkladě zvolíme nulovou apriorní tabulku, která odpovídá nulové apriorní informaci. S touto nulovou počáteční statistikou ν_0 dostaneme odhadovou statistiku

$$\nu_{y|\psi;0}$$

| $[v_{1;t}, v_{2;t}, v_{3;t}]$ | $y_t = 1$ | $y_t = 2$ |
|-------------------------------|-----------|-----------|
| [1 1 1] | 0 | 0 |
| [1 1 2] | 1 | 0 |
| [1 2 1] | 2 | 0 |
| [1 2 2] | 3 | 1 |
| [2 1 1] | 0 | 1 |
| [2 1 2] | 3 | 1 |
| [2 2 1] | 1 | 2 |
| [2 2 2] | 2 | 1 |

odkud (prostou normalizací řádků tabulky na součet jedna) dostaneme odhad parametrů modelu.

$$\hat{\Theta}_{y|\psi;n_t}$$

| $[v_{1;t}, v_{2;t}, v_{3;t}]$ | $y_t = 1$ | $y_t = 2$ |
|-------------------------------|-----------|-----------|
| [1 1 1] | – | – |
| [1 1 2] | 1 | 0 |
| [1 2 1] | 1 | 0 |
| [1 2 2] | 3/4 | 1/4 |
| [2 1 1] | 0 | 1 |
| [2 1 2] | 3/4 | 1/4 |
| [2 2 1] | 1/3 | 2/3 |
| [2 2 2] | 2/3 | 1/3 |

Výsledek je poměrně špatný. Parametry v prvním řádku nelze určit (zde statistika nebyla vůbec přepočtena), řádek 2,3 a 5 je deterministický (ve skutečnosti je to dáno tím, že do těchto řádků přišel jen jediný údaj) a ostatní řádky vypadají lépe, nicméně každý řádek odpovídá pokusu s hodem mincí. Umíme si představit, kolik hodů je třeba, abychom alespoň trochu objektivně posoudili pravděpodobnosti jejich stran. Podle hodnot statistiky vidíme, že do těchto řádků jsou započteny maximálně čtyři údaje. To je velmi málo a je zřejmé, že vzhledem k dimenzi tabulky statistiky máme neúnosně málo dat. Nedostatek dat v těchto případech je velmi častý a je třeba možnosti úlohy v takovém případě dobře zvážit.

Jednou z možností, jak řešit nedostatek dat, je uvažovat další informaci a tou je informace expertní. V krajním případě je dokonce možné postavit model na expertní (apriorní) informaci a změřená data použít jen k jakési korekci modelu. Tuto možnost ukážeme v další části příkladu.

Apriorní statistiku je možno sestavit takto: pro všechny řádky tabulky (tj. pro všechny možné regresní vektory)

1. určíme, jaké pravděpodobnosti bychom přiřadili hodnotám y pro konkrétní kombinaci hodnot ψ ,

2. tyto pravděpodobnosti násobíme číslem, které vyjadřuje míru důvěry k našemu přiřazení.

Např.: první regresní vektor v tabulce je $\psi = [1, 1, 1]'$, tj. rychlost = normální, počasí = sucho, světlo = dobré. V této situaci rozhodujeme o typu nehody. Můžeme říci, že jsou to ideální podmínky, a tak nehoda může být jen lehká. Volíme proto první řádek statistiky

$$\nu_{\cdot|111;0} = [9, 1].$$

To odpovídá 10 údajům o nehodách, z nichž 9 bylo lehkých a 1 těžká.

Druhý regresní vektor v tabulce je $\psi = [1, 1, 2]$ - rychlost = normální, počasí = sucho, světlo = šero. Za šera jsou někdy podmínky jízdy ošemetné zvláště pro chodce, kteří jsou špatně vidět. Proto volíme

$$\nu_{\cdot|112;0} = [1, 4],$$

což odpovídá pěti záznamům, jeden s lehkou a čtyři s těžkou nehodou.

Stejně budeme pokračovat i pro další regresní vektory, a tak získáme následující tabulku pro apriorní statistiku.

$$\nu_0$$

| $[v_1, v_2, v_3]$ | $y = 1$ | $y = 2$ |
|-------------------|---------|---------|
| [1 1 1] | 9 | 1 |
| [1 1 2] | 1 | 4 |
| [1 2 1] | 2 | 2 |
| [1 2 2] | 2 | 3 |
| [2 1 1] | 3 | 2 |
| [2 1 2] | 3 | 7 |
| [2 2 1] | 3 | 7 |
| [2 2 2] | 1 | 9 |

Odhadem z dat získáme parametry:

$$\hat{\Theta}_{y|\psi;n_t} \tag{5.6}$$

| $[v_{1;t}, v_{2;t}, v_{3;t}]$ | $y_t = 1$ | $y_t = 2$ |
|-------------------------------|-----------|-----------|
| [1 1 1] | 9/10 | 1/10 |
| [1 1 2] | 1/3 | 2/3 |
| [1 2 1] | 2/3 | 1/3 |
| [1 2 2] | 5/9 | 4/9 |
| [2 1 1] | 1/2 | 1/2 |
| [2 1 2] | 3/7 | 4/7 |
| [2 2 1] | 4/13 | 9/13 |
| [2 2 2] | 3/13 | 10/13 |

V řádku, kde nepřišla žádná data (např. 1. řádek), zůstaly apriorní odhady. Tam, kam data přišla, je apriorní informace korigována daty.

5.2 Odhad parametrů logistického modelu

V případě, kdy modelovaná veličina je diskrétní a závisí jak na diskrétních tak i na spojitých veličinách, použijeme model logistické regrese. Tento model lze použít i v případě, kdy všechny veličiny jsou diskrétní ale mají velký počet různých hodnot, takže čistě diskrétní model by měl příliš vysokou dimenzi.

Zde se budeme zabývat odhadem modelu zavedeného podle (3.2, 3.3), tedy modelem, který má tvar

$$\text{logit}(p_t) = \psi_t' \Theta + e_t,$$

kde $p_t = P(y_t = 1 | \psi_t)$ a $y_t \in \{0, 1\}$, P je pravděpodobnost a y_t je dvouhodnotový výstup. ψ_t je regresní vektor externích veličin, např. “počasí” s hodnotami sucho, mokro, námraza nebo “den v týdnu” s hodnotami všední den, víkend. Parametry modelu jsou prvky vektoru Θ . Tyto parametry se odhadují na základě změřené množiny dat $d(t) = \{y_\tau, \psi_\tau\}_{\tau=1}^t$, kde y_τ jsou skaláry a $\psi_\tau = [1, x_{1;\tau}, x_{2;\tau}, \dots, x_{n;\tau}]'$ je sloupcový vektor hodnot nezávislé proměnné. Jednička na začátku je přidána pro odhad konstanty modelu.

Lze sestavit i algoritmus pro odhad logistické regrese s více hodnotami y_t . Ten je ale složitější a my se s ním zde nebudeme zabývat.

5.2.1 Odhad logistického modelu

Tato úloha nemá reprodukovatelnou statistiku. Odhad se provádí jednorázově metodou ML (maximum likelihood) pro celý shromážděný datový vzorek. Za tímto účelem konstruujeme logaritmus věrohodnostní funkce

$$\ln L(\Theta) = \ln \prod_{\tau=1}^t f(y_\tau | \psi_\tau, \Theta)$$

jako součin modelů (3.2), kde platí (3.3). Po dosazení a drobných úpravách dostaneme

$$\ln L(\Theta) = \ln \prod_{\tau=1}^t \frac{\exp(y_\tau z_\tau)}{1 + \exp(z_\tau)} = \sum_{\tau=1}^t [y_\tau z_\tau - \ln(1 + \exp(z_\tau))],$$

kde $z_t = \psi_t' \Theta$.

Bodové odhady leží v maximu logaritmu věrohodnostní funkce. Toto maximum lze výhodně nalézt Newtonovou metodou, protože jak gradient (první derivace), tak i Hessovu matici (druhá derivace) je možno vyjádřit v analytickém tvaru. Odvození je uvedeno v Příloze 10.11.1.

5.2.2 Předpověď výstupu

Logistickou regresi nejspíše děláme proto, abychom byli schopní pro daný regresní vektor odhadnout (pro jednotnost s předchozím výkladem říkáme předpovědět) hodnotu odpovídajícího výstupu.

Toho lze dosáhnout tak, že bodové odhady $\hat{\Theta}_t$ dosadíme do modelu (3.2), (3.3) a pro libovolný regresní vektor ψ , dostáváme odhad⁶ pravděpodobnosti hodnot y

$$f(y | \psi, \hat{\Theta}_t) = \frac{\exp(y \psi' \hat{\Theta}_t)}{1 + \exp(\psi' \hat{\Theta}_t)}.$$

⁶Píšeme odhad, protože správné pravděpodobnosti bychom dostali nikoli dosazením bodových odhadů, ale násobením aposteriorní hp a integrací před parametry Θ . To je ale příliš složité.

Bodovou predikci \hat{y} pro regresní vektor ψ určíme jako podmíněnou střední hodnotu⁷

$$\hat{y} = E[y|\psi, d(t)] = \sum_{y=0}^1 yf(y|\psi, \hat{\Theta}_t) = \frac{\exp(\psi' \hat{\Theta}_t)}{1 + \exp(\psi' \hat{\Theta}_t)}.$$

Pokud požadujeme bodovou predikci pouze v přípustných hodnotách $\{0, 1\}$, získanou hodnotu \hat{y} zaokrouhlíme (pro $\hat{y} < 0.5$ na hodnotu 0 a pro $\hat{y} \geq 0.5$ na hodnotu 1).

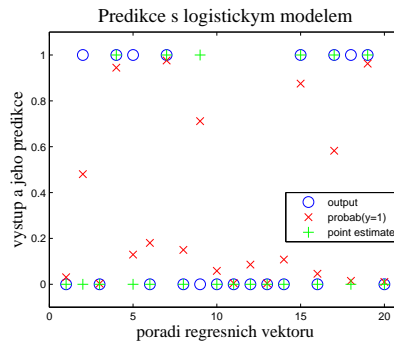
5.2.3 Příklad [odhad logistické regrese 1]

Uvažujme diskrétní veličinu y závislou na dvou veličinách regresního vektoru $\psi = [\psi_1, \psi_2]$. Tyto veličiny modelujeme jako normální náhodné veličiny $\psi_1 \sim N(-0.1, 0.25)$ a $\psi_2 \sim N(0.3, 1)$. Hodnoty y jsou přiřazeny následovně

$$y = \begin{cases} 1 & \text{pro } 2\psi_1 - \psi_2 + e > 1 \\ 0 & \text{jinde} \end{cases}, \quad (5.7)$$

kde $e \sim N(0, 1)$ je šum. Máme provést logistickou regresi těchto dat.

Řešení je dáno v programu T23LogRegM.m, který je možno nalézt v kapitole 11 Programy na str. 89. Pro odhad byl použit vzorek 50 dat. Výsledek demonstrováný na dvaceti následujících (testovacích) datech je na obrázku.



Z obrázku je vidět, že bodové odhady (+) většinou správně sedí na hodnotách výstupu (o), ale pravděpodobnosti predikcí (x) vykazují určitý stupeň neurčitosti. Ta je dána použitým heuristickým generátorem dat a šumem e , který jsme do dat přičetli při určování hodnot výstupu (5.7).

5.2.4 Příklad [odhad s logistickým modelem 2]

Uvažujme systém s výstupem $y \in \{1, 2\}$ a dalšími veličinami, které tvoří regresní vektor $\psi = [\psi_1, \psi_2, \psi_3]$, $\psi_i \in \{1, 2\}$. Dále jsme změřili data $d(t) = d(5)$, která jsou uvedena v následující tabulce.

| data | ψ | y |
|------|-----------|-----|
| 1 | [2, 2, 2] | 1 |
| 2 | [1, 2, 2] | 2 |
| 3 | [1, 1, 1] | 1 |
| 4 | [1, 1, 1] | 1 |
| 5 | [1, 1, 1] | 2 |

⁷Z následujícího vzorce je patrné, že platí $\hat{y} = f(y|\psi, \hat{\Theta}_t) = P(y = 1|\psi, \hat{\Theta}_t)$.

Data jsou pod vlivem šumu. Je vidět, že poslední tři datové položky jsou ve sporu. Třetí a čtvrtá říká, že regresnímu vektoru $[1, 1, 1]$ odpovídá hodnota jedna, zatímco u páté položky je to dvojka.

Cílem příkladu je provést odhad koeficientů logistické regrese, určit predikce pro y a ukázat, jak je provedena klasifikace.

Pro odhad a predikci je opět možno použít program z m-souboru T23LogRegM.m, který je uveden na str. 89. Odtud je možno získat výsledky:

Rovnice logistické regrese

$$\text{logit}(y) = b_0 + b_1\psi_1 + b_2\psi_2 + b_3\psi_3 + b_4\psi_4 \quad (5.8)$$

Parametry regrese

| | | | |
|-------|--------|-------|-------|
| b_0 | b_1 | b_2 | b_3 |
| 29.28 | -63.22 | 16.49 | 16.76 |

V následující tabulce jsou uvedeny měřené hodnoty regresních vektorů ψ_t , výstupu y_t , predikované hodnoty výstupu y_t - pravděpodobnosti $P(y_t = 1)$ a bodové odhady \hat{y}_t (zaokrouhlené hodnoty pravděpodobnosti).

| t | ψ | y_t | $P(y_t = 1)$ | \hat{y}_t |
|-----|-----------|-------|--------------|-------------|
| 1 | [2, 2, 2] | 1 | 0 | 1 |
| 2 | [1, 2, 2] | 2 | 1 | 2 |
| 3 | [1, 1, 1] | 1 | 0.33 | 1 |
| 4 | [1, 1, 1] | 1 | 0.33 | 1 |
| 5 | [1, 1, 1] | 2 | 0.33 | 1 |

Z tabulky je patrné, že predikce pro hodnoty regresního vektoru, při kterých se vyskytovaly sporné hodnoty výstupu, nemají pravděpodobnost nula nebo jedna. Tyto predikce říkají, že jejich hodnoty jsou nejisté, ale přiklánějí se k hodnotě nula, protože nula odpovídala dvěma případům dat, zatímco jednička pouze jednomu. Ostatní predikce jsou správné a přesné.

5.3 Klasifikace

Klasifikace je úloha, ve které jednotlivé datové položky zařazujeme do předem daných tříd označených indexem $c = 1, 2, \dots, n_c$, kde n_c je počet tříd.

Přístupů k řešení úlohy klasifikace je celá řada. Jsou to např. algoritmy K-means, DBSCAN, COBWEB a další, shrnuté v systému Weka, volně ke stažení na

http://www.cs.waikato.ac.nz/ml/weka/index_downloading.html.

Bayesovský přístup ke klasifikaci nejčastěji využívá modely ve tvaru směsi komponent, o kterých budeme mluvit v Kapitole ???. Nicméně každý model $f(y_t|\psi_t)$ s diskrétním výstupem $y_t \in \{1, 2, \dots, n_c\}$ lze využít pro klasifikaci. Hodnota předpovědi \hat{y}_t může být považována za index třídy, do které patří regresní vektor ψ_t . Model tak rozděluje prostor všech možných regresních vektorů do n_c podprostorů a každý z těchto podprostorů tvoří jednu třídu klasifikace. Využití diskrétního modelu pro úlohu klasifikace ukážeme v následujícím příkladě.

5.3.1 Příklad [klasifikace s diskretním modelem]

Pokračujeme v příkladu 5.1.7.

V našem příkladě je $y_t \in \{1, 2\}$. První hodnota odhadu výstupu $\hat{y}_t = 1$ indikuje regresní vektor (vektor veličin mající vliv na vážnost nehody) z první třídy (lehké nehody) a druhá hodnota $\hat{y}_t = 2$ zařazuje okolnosti nehody do druhé (těžké nehody).

Pro výpočet odhadu výstupu máme vzorec (4.3)

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

Jestliže přejdeme k bodovým odhadům parametru, tedy vezmeme aposteriorní hp jako Diracův impulz v bodovém odhadu (4.5) $f(\Theta|d(t-1)) = \delta(\Theta - \hat{\Theta}_{t-1})$, kde bodové odhady parametrů jsou prvky tabulky (5.6), platí (4.6)

$$f(y_t|\psi_t, d(t-1)) = \int_{\Theta^*} f(y_t|\psi_t, \Theta) \delta(\Theta - \hat{\Theta}_{t-1}) d\Theta = f(y_t|\psi_t, \hat{\Theta}_{t-1}).$$

Znamená to, že prediktivní hp je přímo dána bodovým odhadem výstupu s dosazeným bodovým odhadem parametrů.

V našem případě, daném tabulkou 5.6, množina všech regresních vektorů (z levé části tabulky) je rozdělena podle pravděpodobností hodnot y_t do dvou skupin: $\{1, 3, 4\}$ - kde větší pravděpodobnost má $y_t = 1$ a $\{2, 6, 7, 8\}$ - kde pravděpodobnější je $y_t = 0$. Regresní vektor 5 je hraniční a může být přiřazen do libovolné skupiny. Skupiny jsou charakterizovány hodnotou předpovídaného výstupu, který jsme určili jako hodnotu s největší pravděpodobností.

Data vypovídají o tom, že lehké nehody se stávají v případech, kdy rychlost byla normální, a počasí a osvětlení byly většinou dobré. Těžké nehody nastávají většinou za velké rychlosti a při počasí a osvětlení spíše špatném.

Logistický model lze rovněž využít pro úlohu klasifikace. V odstavci 5.2 jsme ukázali, jak lze na základě odhadu logistické regrese z dat $d(t) = \{y(t), \psi(t)\}$ pro libovolně zvolený regresní vektor ψ určit jemu odpovídající predikci \hat{y} . Přitom množina $\psi(t)$ nemusí zdaleka obsahovat (v praxi také neobsahuje) všechny konfigurace hodnot regresního vektoru ψ . Odhad logistické regrese a na něm založená predikce tak provádí klasifikaci v prostoru všech konfigurací regresního vektoru ψ , tj. každý regresní vektor ψ přiřadí do jedné ze dvou skupin: první skupina regresních vektorů dává predikci 1 a druhá predikci 2.

5.3.2 Příklad [klasifikace s logistickou regresí]

Navazujeme na příklad 5.2.4.

Do odhadnuté rovnice logistické regrese (5.8) můžeme za regresní vektor postupně dosadit všechny možné hodnoty regresního vektoru a získat tak predikce i pro ty regresní vektory, které ve skutečnosti nebyly naměřeny. Tím získáme optimální odpověď na otázku: "Co bychom obdrželi, kdyby nastalo ...?". Situace je v následující tabulce

| i | ψ_i | \hat{y}_i |
|-----|-----------|-------------|
| 1 | [1, 1, 1] | 1 |
| 2 | [1, 1, 2] | 2 |
| 3 | [1, 2, 1] | 2 |
| 4 | [1, 2, 2] | 2 |
| 5 | [2, 1, 1] | 1 |
| 6 | [2, 1, 2] | 1 |
| 7 | [2, 2, 1] | 1 |
| 8 | [2, 2, 2] | 1 |

Z této tabulky je vidět, že množina všech regresních vektorů (popisujících určitou situaci) se rozpadne na dvě množiny, vzhledem k hodnotě předpovídaného výstupu (určitého příznaku situace). První množina charakterizovaná hodnotou predikce 0 je množina regresních vektorů {1, 5, 6, 7, 8} a druhá množina obsahuje regresní vektory {2, 3, 4}. Tím jsme provedli klasifikaci regresních vektorů (situací) ψ podle hodnoty predikce výstupu (příznaku) y .

6 Předpověď modelované veličiny

Jestliže sledujeme vývoj modelované veličiny v čase a popisujeme ji pomocí dynamického modelu, můžeme se v čase t zajímat o její předpověď v čase $t + k$, tedy odhadovat její hodnotu y_{t+k} . Úplný popis předpovědi můžeme vyjádřit opět pomocí podmíněné hp

$$f(y_{t+k}|y(t)). \quad (6.1)$$

Částečný popis předpovědi, tedy její bodový odhad (viz Přílohy 10.8), určuje podmíněná střední hodnota

$$E[y_{t+k}|y(t)] = \int_{y^*} y_{t+k} f(y_{t+k}|y(t)) dy_{t+k}. \quad (6.2)$$

Výpočet předpovědi se liší podle (i) počtu kroků v předpovědi (jednokroková, vícekroková), (ii) parametrů modelu (známé, neznámé), (iii) formě odhadu (bayesovský odhad, odhad s bodovým odhadem parametrů, bodový odhad výstupu), (iv) typu modelu (diskrétní, spojitý).

Data pro předpověď: Pro jednoduchost budeme v této kapitole uvažovat data tvořená pouze výstupem soustavy. Nebudeme uvažovat ani řízení, ani žádnou další externí veličinu.

Časové značení: Dále budeme uvažovat situaci, kdy jsme v čase t , tj. změřili jsme hodnotu y_t , případně jsme přepočítali statistiky odhadu parametrů a máme aposteriorní hp $f(\Theta|y(t))$, případně jsme spočítali bodové odhady parametrů a nyní se zajímáme o hodnotu výstupu k kroků dopředu.

6.1 Jednokroková předpověď

Pro $k = 1$ dostáváme jednokrokovou předpověď. O té jsme již mluvili dříve (viz odstavec 4.1.9) jako o odhadu výstupu. Pokud máme **model systému se známými parametry** Θ , je situace triviální. Předpověď výstupu je dána přímo modelem $f(y_{t+1}|y(t))$ nebo jeho střední hodnotou $\hat{y}_{t+1} = E[y_{t+1}|y(t)] = \int_{y^*} y_{t+1} f(y_{t+1}|y(t)) dy_{t+1}$.

6.1.1 Příklad [jednokroková predikce s regresním modelem]

Pro regresní model $y_t = a_1 y_{t-1} + a_2 y_{t-2} + e_t$, kde $e_t \sim N(0, r)$ je jednokroková prediktivní hp dána hustotou modelu

$$f(y_{t+1}|y(t)) = \frac{1}{\sqrt{2\pi r}} \exp\left\{-\frac{1}{2r}(y_{t+1} - a_1 y_t - a_2 y_{t-1})^2\right\}$$

a bodová předpověď je

$$E[y_{t+1}|y(t)] = a_1 y_t + a_2 y_{t-1}.$$

Obecně totiž pro model $y_t = \psi_t' \theta + e_t$ platí:

$$E[y_t|\psi_t, \Theta] = E[\psi_t' \theta + e_t] = E[\psi_t' \theta] + E[e_t] = E[\psi_t' \theta] \quad (6.3)$$

$$D[y_t|\psi_t, \Theta] = D[\psi_t' \theta + e_t] = D[e_t] = r, \quad (6.4)$$

protože $\psi_t' \theta$ je konstanta⁸.

Pokud **parametry modelu neznáme**, musíme postupovat tak, jak jsme uvedli v kapitole o odhadování ve vztahu (4.3)

$$f(y_{t+1}|y(t)) = \int_{\Theta^*} f(y_{t+1}|\psi_t, \Theta) f(\Theta|y(t)) d\Theta, \quad (6.5)$$

kde jsme doplnili a ihned zase vy-integrovali parametry Θ a rozvinuli sdruženou hp podle řetězového pravidla⁹.

Hustota pravděpodobnosti predikce, tj. prediktivní hp, je dána integrálem ze součinu hp modelu a hp popisující parametry, tj. aposteriorní hp. S aposteriorní hp jsme se již setkali při odhadu a víme, že tuto hp musíme postupně vyvíjet podle Bayesova vzorce počínaje od apriorní hp. Predikce modelované veličiny s modelem s neznámými parametry v sobě tedy obsahuje implicitně obě úlohy: odhad parametrů a předpověď výstupu.

6.1.2 Poznámka

Jednokroková předpověď výstupu (nebo jiné modelované veličiny) je velice důležitá. Řada úloh odhadu i řízení je založena na výpočtu chyby predikce $\hat{e}_t = y_t - \hat{y}_t$ jako rozdílu změřené veličiny a její predikce. Další akce se provádí tak, aby se tato chyba zmenšovala. ◁

6.2 Vícekroková předpověď

Pro $k > 1$ v (6.1) hovoříme o vícekové předpovědi. Při jejím odvození budeme postupovat podobně jako při konstrukci jednokrokové predikce s modelem s neznámými parametry v (6.5). Postup budeme ilustrovat na příkladě **dvou-krokové předpovědi**

⁸Protože jak ψ_t , tak i Θ jsou v podmínce.

⁹V podmínce modelu se místo celé historie uplatní jen regresní vektor ψ_t .

$$\begin{aligned}
f(y_{t+2}|y(t)) &= \int_{\Theta^*} \int_{y_{t+1}^*} f(y_{t+2}, y_{t+1}, \Theta|y(t)) dy_{t+1} d\Theta = \\
&\quad \text{(rozklad podle řetězového pravidla)} \\
&= \int_{\Theta^*} \int_{y_{t+1}^*} f(y_{t+2}|y_{t+1}y(t), \Theta) f(y_{t+1}|y(t), \Theta) f(\Theta|y(t)) dy_{t+1} d\Theta = \\
&\quad \text{(podmínky nezávislosti)} \\
&= \int_{\Theta^*} \int_{y_{t+1}^*} f(y_{t+2}|\psi_{t+2}, \Theta) f(y_{t+1}|\psi_{t+1}, \Theta) dy_{t+1} f(\Theta|y(t)) d\Theta \tag{6.6}
\end{aligned}$$

Označíme-li

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

jako prediktor se známými parametry modelu, pak prediktivní hp (6.6) lze zapsat podobně jako v případě jednokrokové předpovědi (6.5)

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

Obecná konstrukce prediktivní hp je již jednoduchým zobecněním uvedeného příkladu. Pro k -krokovou předpověď platí

$$f(y_{t+k}|y(t)) = \int_{\Theta^*} f(y_{t+k}|y(t), \Theta) f(\Theta|y(t)) d\Theta, \tag{6.7}$$

kde

$$f(y_{t+k}|y(t), \Theta) = \int_{y_{t+k-1}} \cdots \int_{y_{t+1}} f(y_{t+k}|y(t+k-1)) \cdots f(y_{t+1}|y(t)) dy_{t+k-1} \cdots dy_{t+1} \tag{6.8}$$

je k -krokový prediktor s modelem se známými parametry.

Kritickým místem při výpočtu k -krokové předpovědi s modelem s neznámými parametry je integrál v rovnici (6.7). Prediktor daný vztahem (6.8) je dán konvolucemi modelu a často je možno jej vyjádřit analyticky. Problém je ale ve výpočtu integrálu (6.7), protože aposteriorní hp je složitá funkce a její konvoluce s prediktorem je většinou analyticky nespočítatelná. Jedním ze způsobů, jak tuto potíž řešit, může být místo aposteriorní hp uvažovat jen bodové odhady parametrů (zanedbá se neurčitost v parametrech a číselné odhady se berou jako přesné). Pro bodové odhady parametrů $\hat{\Theta}_t$ je možno formálně psát hp

$$f(\Theta|d(t)) = \delta(\Theta - \hat{\Theta}_t)$$

ve tvaru Diracovy funkce: $\delta(0)$ je jedna jinak je rovna nule.

Dosadíme do (6.7) a dostaneme

$$f(y_{t+k}|y(t)) = \int_{\Theta^*} f(y_{t+k}|y(t), \Theta) \delta(\Theta - \hat{\Theta}_t) d\Theta = f(y_{t+k}|y(t), \hat{\Theta}_t). \tag{6.9}$$

To znamená, že pro výpočet předpovědi stačí vzít prediktor (6.8) a na místo neznámých parametrů dosadit jejich bodové odhady.

6.3 Vícekroková bodová předpověď

Výpočet vícekrokové předpovědi výstupu s modelem s neznámými parametry je úloha tak složitá, že ji v praxi nelze použít. Ukážeme zde proto na příkladech některé jednodušší případy předpovědi a její řešení.

6.3.1 Předpověď se spojitým modelem

K významnému zjednodušení úlohy vícekrokové předpovědi výstupu systému dojde, jestliže přejdeme k bodovým odhadům, a to jak pro neznámé parametry modelu (6.9), tak i pro vyjádření předpovědi ve tvaru hodnoty \hat{y}_{t+k} . Bodovou předpověď výstupu pro lineární normální regresní model budeme demonstrovat v následujícím příkladě.

6.3.2 Příklad [bodová předpověď]

Uvažujme tříkrokovou bodovou predikci s modelem $y_{t+1} = a_0 y_t + a_1 y_{t-1} + e_t$ s počáteční podmínkou $y_t = \xi_0$ a $y_{t-1} = \xi_{-1}$.

Dostáváme:

První krok

$$\hat{y}_{t+1} = a_0 y_t + a_1 y_{t-1} = a_0 \xi_0 + a_1 \xi_{-1},$$

Druhý krok

$$\begin{aligned} \hat{y}_{t+2} &= a_0 y_{t+1} + a_1 y_t = a_0 \hat{y}_{t+1} + a_1 \xi_0 = \\ &= a_0 (a_0 \xi_0 + a_1 \xi_{-1}) + a_1 \xi_0 = (a_0^2 + a_1) \xi_0 + a_0 a_1 \xi_{-1}, \end{aligned}$$

Třetí krok

$$\begin{aligned} \hat{y}_{t+3} &= a_0 y_{t+2} + a_1 y_{t+1} = a_0 \hat{y}_{t+2} + a_1 \hat{y}_{t+1} = \\ &= a_0 ((a_0^2 + a_1) \xi_0 + a_0 a_1 \xi_{-1}) + a_1 (a_0 \xi_0 + a_1 \xi_{-1}) = \\ &= (a_0^3 + 2a_0 a_1) \xi_0 + (a_0^2 a_1 + a_1^2) \xi_{-1}. \end{aligned}$$

Předpověď s normálním modelem, kterou jsme se zabývali v předchozím příkladě, lze zobecnit tím, že budeme počítat nikoli bodový odhad předpovědi, ale celo-prediktivní hp (samozřejmě pro známé parametry nebo parametry dosažené jako bodové odhady). Využijeme toho, že konvoluce dvou normálních rozdělání, které při výpočtu předpovědi vznikají, mají opět normální rozdělání. Není proto potřeba počítat celou prediktivní hp (integrály), ale napočítávat jen střední hodnoty a rozptyly. Konečná prediktivní hp je jimi plně určena. Postup výpočtu je následující: místo střední hodnoty (což je deterministická část regresního modelu) dosazujeme rovnici modelu i se šumem. Z výsledné hodnoty y_{t+k} pak určíme hledanou střední hodnotu a rozptyl. Situaci ilustruje následující příklad.

6.3.3 Příklad [předpověď pro normální rozdělení]

Určete dvou-krokovou predikci s modelem $y_{t+1} = ay_t + e_{t+1}$.

$$y_{t+2} = ay_{t+1} + e_{t+2} = a(ay_t + e_{t+1}) + e_{t+2} = a^2y_t + e_{t+2} + ae_{t+1}.$$

Protože střední hodnota konstanty je konstanta a střední hodnota šumu je nula, je

$$E[y_{t+2}|y(t)] = a^2y_t + E[e_{t+2} + ae_{t+1}] = a^2y_t.$$

Protože platí $D[e_{t+2}] = D[e_{t+1}] = r$ a při vytknutí konstanty z rozptylu se tato konstanta umocní na druhou, bude

$$D[y_{t+2}|y(t)] = D[e_{t+2} + ae_{t+1}] = (1 + a^2)r.$$

Hp předpovědi má tvar

$$f(y_{t+2}|y(t)) = \mathcal{N}_{y_{t+2}}(a^2y_t, (1 + a^2)r).$$

6.3.4 Předpověď s diskretním modelem

Tak jako ve všech úlohách je i zde práce s diskretním modelem jednoduchá (nemusí se řešit žádné integrály a pouze se násobí a sčítají prvky tabulek). Operace s tabulkami jsou však poněkud nezvyklé a mohou působit problémy. Tuto práci budeme demonstrovat v následujícím příkladě. Tento příklad byl vybrán jako autoregrese prvního řádu, což je případ obzvláště jednoduchý. V obecnějším případě (např. pro řízený model) je práce s tabulkami mnohem náročnější.

6.3.5 Příklad [předpověď s diskretním modelem]

Predikci s diskretním modelem ukážeme na tom nejjednodušším příkladu, tj. s neřízeným modelem prvního řádu $f(y_t|y_{t-1})$ zadaným následující tabulkou.

| $f(y_t y_{t-1})$ | y_t | |
|------------------|----------------|----------------|
| | 1 | 2 |
| y_{t-1} | | |
| 1 | $\Theta_{1 1}$ | $\Theta_{2 1}$ |
| 2 | $\Theta_{1 2}$ | $\Theta_{2 2}$ |

Budeme chtít vyjádřit obecně k -krokovou predikci $f(y_k|y_0)$ s počáteční podmínkou $y_0 = 1$. V tomto jednoduchém případě lze výsledek vyjádřit analyticky, takže si můžeme dovolit uvažovat parametry modelu obecně.

Pro lepší přehlednost odvodíme předpověď pro $k = 3$ a potom jednoduše zobecníme.

Rozkladem prediktivní pf (podle (6.6), pro $k = 3$) dostaneme

$$f(y_3|y_0) = \sum_{y_2=1}^2 f(y_3|y_2) \underbrace{\sum_{y_1=1}^2 f(y_2|y_1) f(y_1|y_0)}_{f(y_2|y_0)}. \quad (6.10)$$

Nejprve si všimněme dvou-krokové předpovědi $f(y_2|y_0)$ označené svorkou. Jedná se o součet součinů prvků tabulky (modelu). Tabulku, která zadává model $f(y_t|y_{t-1})$, označíme M s prvky $m_{\tau,t}$ (všimněte si, že indexy u prvků tabulky jsou přehozené - první index jde svisle, druhý vodorovně - značení vychází ze zápisu podmíněné pravděpodobnosti). Dosadíme do předchozího výrazu pro předpověď a dostaneme

$$f(y_2|y_0) = \sum_{y_1=1}^2 m_{y_1,y_2} m_{y_0,y_1} = \sum_{y_1=1}^2 m'_{y_2,y_1} m'_{y_1,y_0} = M' M' = (M')^2,$$

kde apostrof značí transpozici a $(M')^2$ je kvadrát transponované matice. Součet součinů prvků tabulek modelu definuje součin matic.

Označíme-li tabulku pf předpovědi $f(y_2|y_0)$ jako M_2 bude platit

$$M'_2 = M' M' \text{ a tedy } M_2 = M^2.$$

Dosadíme do tří-krokové předpovědi (6.10) a s využitím vztahu $M_2 = M^2$ dostaneme

$$f(y_3|y_0) = \sum_{y_2=1}^2 f(y_3|y_2) f(y_2|y_0) = M' M'_2 = (M_2 M)' = (M^3)'$$

Jestliže tabulku tří-krokové předpovědi označíme M'_3 bude platit

$$M_3 = M^3.$$

Pro větší přehlednost zavedeme model číselně: $\Theta_{1|1} = 0.3$, $\Theta_{2|1} = 1 - \Theta_{1|1} = 0.7$ a $\Theta_{1|2} = 0.6$, $\Theta_{2|2} = 1 - \Theta_{1|2} = 0.4$. Tedy tabulka M' bude

$$M' = \begin{bmatrix} 0.3 & 0.7 \\ 0.6 & 0.4 \end{bmatrix}$$

a příslušná matice M (transponovaná tabulka) je

$$M = \begin{bmatrix} 0.3 & 0.6 \\ 0.7 & 0.4 \end{bmatrix}$$

Tříkroková předpověď $f(y_3|y_0)$ je reprezentována maticí

$$M_3 = M^3 = \begin{bmatrix} 0.3 & 0.6 \\ 0.7 & 0.4 \end{bmatrix} \begin{bmatrix} 0.3 & 0.6 \\ 0.7 & 0.4 \end{bmatrix} \begin{bmatrix} 0.3 & 0.6 \\ 0.7 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.51 & 0.42 \\ 0.49 & 0.58 \end{bmatrix} \begin{bmatrix} 0.3 & 0.6 \\ 0.7 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.447 & 0.474 \\ 0.553 & 0.526 \end{bmatrix}.$$

Tabulka předpovědi bude dána transpozicí této matice, tedy

| | | $f(y_3 y_0)$ | |
|-------|-----------|--------------|--|
| y_0 | $y_3 = 1$ | $y_3 = 2$ | |
| 1 | 0.447 | 0.553 | |
| 2 | 0.474 | 0.526 | |

Pro obecnou k -krokovou předpověď bude platit

$$M_k = M^k$$

a tabulka této předpovědi bude transpozicí, tedy

$$f(y_k|y_0) \rightarrow (M^k)'$$

Hodnoty tabulky se po více krocích ustálí. Pro $k \geq 9$ vypadá tabulka následovně

$$f(y_k|y_0)$$

| y_0 | $y_k = 1$ | $y_k = 2$ |
|-------|-----------|-----------|
| 1 | 0.4615 | 0.5385 |
| 2 | 0.4615 | 0.5385 |

To znamená, že bez ohledu z jaké hodnoty y_0 jsme vycházeli, pro $k \geq 9$ předpovídáme $y_k = 1$ s pravděpodobností 0.4615 a $y_k = 2$ s pravděpodobností 0.5385 (což neodporuje zdravému rozumu).

6.3.6 Poznámka

Znovu připomeňme, že jednoduchost uvedeného příkladu je dána tím, že jsme mohli využít maticového počtu. V ostatních případech to jednoduše nelze a počítání je mnohem komplikovanější. ◀

7 Stav, stavový model a odhad stavu

7.1 Stav

Stav x_t je veličina systému, kterou nelze měřit a která se (na rozdíl od parametrů) v mění čase. Aby veličina mohla být stavem, musí platit, že v každém časovém okamžiku t v sobě zahrnuje veškerou informaci ze svého minulého vývoje. Pokud známe jeho hodnotu a aktuální řízení, jsme schopni spočítat distribuci jeho budoucí hodnoty.

7.1.1 Příklad [pojem stavu]

Uvažujme systém spojený s městskou řízenou čtyř-ramennou křižovatkou. V ramenech křižovatky měříme intenzitu provozu a zaznamenáváme poměry zelené na semaforu. Zajímáme se o délky kolon v křižovatce. Tato veličina „vektor délek kolon v křižovatce“ je v této úloze stavem. Nelze ji přímo měřit a máme-li délku kolony a víme-li, jaké řízení bude použito, můžeme počítat budoucí délku kolony. Přičteme všechna auta, co přijela do kolony před semaforem, a odečteme všechna auta, co odjela na zelenou. Různé délky aut a další zaokrouhlení spojená s periodou vzorkování zahrneme do šumu. Tento šum vnáší do výpočtů neurčitost.

7.2 Stavový model

Model stavové veličiny je reprezentován dvěma vztahy: **modelem stavu** a **modelem výstupu**. Pro oba modely se uvažuje normální rozdělení

– model dynamiky stavu

$$f(x_{t+1}|x_t, u_t) = \mathcal{N}_{x_{t+1}}(Mx_t + Nu_t, r_x), \quad (7.1)$$

– model měření výstupu

$$f(y_t|x_t, u_t) = \mathcal{N}_{y_t}(Ax_t + Bu_t, r_y), \quad (7.2)$$

kde \mathcal{N}_x a \mathcal{N}_y značí normální rozdělení.

Oba modely lze vyjádřit jako rovnice

$$x_{t+1} = Mx_t + Nu_t + w_t \quad (7.3)$$

$$y_t = Ax_t + Bu_t + v_t \quad (7.4)$$

kde $w_t \sim \mathcal{N}_w(0, r_x)$ a $v_t \sim \mathcal{N}_v(0, r_y)$.

7.3 Odhad stavu

Cílem je odhad stavu x_t pro $t = 1, 2, \dots$. Odhad stavu je podobně jako odhad parametrů dán hp stavu za podmínky změřených dat $f(x_t|d(t))$. V úloze rekurzivního odhadu parametrů (4.1) existoval jen posun času při vzorkování dat. Při odhadu stavu existují posuny dva:

– posun času v datech - tzv. **filtrace**, při které se informace z nově změřených dat vkládá prostřednictvím modelu výstupu do odhadu stavu; jde o přepočítání $f(x_t|d(t-1)) \rightarrow f(x_t|d(t))$ a

– posun času ve stavu - tzv. **predikce**, při které se stav predikuje podle modelu dynamiky stavu “naslepo”, aniž by se měřila nová data; jde o přepočítání $f(x_t|d(t)) \rightarrow f(x_{t+1}|d(t))$.

Uvedené přepočty hp jsou následující

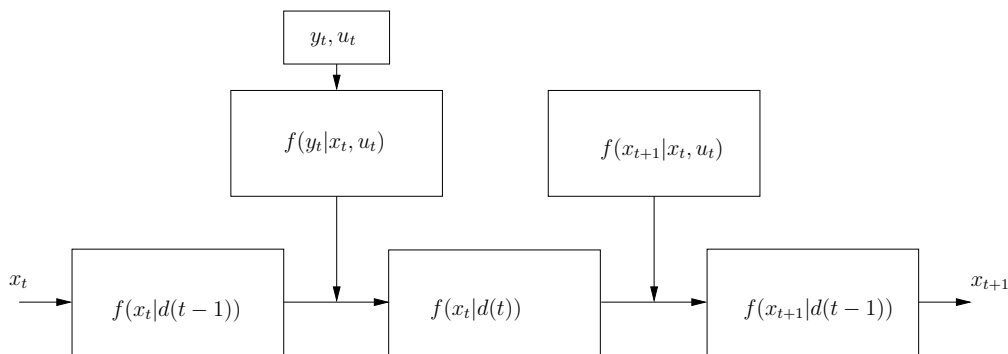
$$f(x_t|d(t)) = \frac{f(y_t|x_t, u_t) f(x_t|d(t-1))}{f(y_t|u_t, d(t-1))} \quad (7.5)$$

$$f(x_{t+1}|d(t)) = \int_{x_t^*} f(x_{t+1}|x_t, u_t) f(x_t|d(t)) dx_t, \quad (7.6)$$

kde

- $f(x_t|d(t-1)) \sim \mathcal{N}(x_{t|t-1}, R_{t|t-1})$ je popis stavu, který vstupuje do KF a který závisí na datech do času $t-1$.
 $x_{t|t-1}, R_{t|t-1}$ jsou charakteristiky tohoto popisu - střední hodnota a kovarianční matice.
- $f(x_t|d(t)) \sim \mathcal{N}(x_{t|t}, R_{t|t})$ je popis stavu po filtraci v okamžiku, kdy je odhad stavu korigován podle aktuálně změřeného výstupu y_t ; tento odhad závisí na datech $d(t) = \{y_t, d(t-1)\}$.
Charakteristiky popisu jsou $x_{t|t}, R_{t|t}$.
- $f(x_{t+1}|d(t)) \sim \mathcal{N}(x_{t+1|t}, R_{t+1|t})$ je popis stavu po predikci, kdy podle stavové rovnice modelu předpovídáme stav do příštího časového okamžiku, aniž bychom získali novou informaci z dat. Tento odhad je tedy stále závislý na datech $d(t)$ ale platí již pro příští čas $t+1$. Tento popis stavu se stává vstupním pro odhad v příští periodě KF.
Charakteristiky popisu jsou $x_{t+1|t}, R_{t+1|t}$.
- $f(y_t|x_t, u_t)$ a $f(x_{t+1}|x_t, u_t)$ jsou hp modelu stavu (7.1) a (7.2).
- $f(y_t|u_t, d(t-1)) = \int_{x_t^*} f(y_t|x_t, u_t) f(x_t|d(t-1)) dx_t$ je predikční hp výstupu (popis výstupu, nezávislý na stavu), která zde vystupuje jen jako normalizační konstanta.

Graficky lze přepočty hp znázornit takto



7.3.1 Poznámka

Odvození rekurze (7.5), (7.6) v případě, kdy stavový model obsahuje člen s řízením u_t , předpokládá existenci tzv. přirozených podmínek řízení. Ty říkají, že odhad stavu a řídicí veličina jsou podmíněně nezávislé. Předpokladem je platnost vztahů

$$f(x_t|u_t, d(t-1)) = f(x_t|d(t-1)) \Leftrightarrow f(u_t|x_t, d(t-1)) = f(u_t|d(t-1)).$$

Diskuze k těmto vztahům je uvedena v Příloze 10.2. ◁

7.4 Kalmanův filtr

Vztahy pro vývoj odhadu stavu podle (7.5) a (7.6) jsou (podobně jako Bayesův vzorec) vztahy pro funkce. S těmi nelze v počítači dobře pracovat. Jestliže všechny hp vyjádříme jako normální hp, lze rekurzi počítat pro jejich charakteristiky - střední hodnoty a rozptyly (kovarianční matice). Dostáváme následující číselnou rekurzi, která je dobře známa jako **Kalmanův filtr**

| |
|---|
| <p>Filtrace:</p> $y_p = Ax_{t t-1} + Bu_t$ <p style="text-align: center;">předpověď výstupu</p> $R_p = r_v + AR_{t t-1}A'$ <p style="text-align: center;">kovariance výstupu</p> $R_{r t} = R_{t t-1} - R_{t t-1}A'R_p^{-1}AR_{t t-1}$ <p style="text-align: center;">přepočet kovariance stavu</p> $K = R_{t t}A'r_y^{-1}$ <p style="text-align: center;">Kalmanův gain</p> $x_{t t} = x_{t t-1} + K(y_t - y_p)$ <p style="text-align: center;">datová oprava stavu</p> |
|---|

Predikce:

$$\underbrace{x_{t|t+1} = Mx_{t|t} + Nu_t}_{\text{předpověď stavu}}$$

$$\underbrace{R_{t+1|t} = r_x + MR_{t|t}M'}_{\text{přepočítání kovariance stavu}}$$

Kalmanův filtr je realizována v programu, který je uveden v části Programy na straně 133.

Použití tohoto programu budeme ilustrovat na příkladě.

7.4.1 Příklad [odhad stavu]

Uvažujme stavový model

$$\begin{bmatrix} x_{1;t+1} \\ x_{2;t+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 0.9 & 0.2 \\ 0 & 1 \end{bmatrix}}_M \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \underbrace{\begin{bmatrix} -1.5 \\ 1 \end{bmatrix}}_N u_t + 0.6 \underbrace{\begin{bmatrix} 1 & 0.3 \\ 0 & 1 \end{bmatrix}}_{sx} \begin{bmatrix} w_{1;t} \\ w_{2;t} \end{bmatrix},$$
$$y_t = \underbrace{\begin{bmatrix} 0.3 & 0.7 \end{bmatrix}}_A \begin{bmatrix} x_{1;t} \\ x_{2;t} \end{bmatrix} + \underbrace{0.1}_{sy} e_t,$$

kde

w_t a e_t jsou šумы s nulovou střední hodnotou a konstantními kovariančními maticemi;

M , N , A jsou matice stavového modelu,

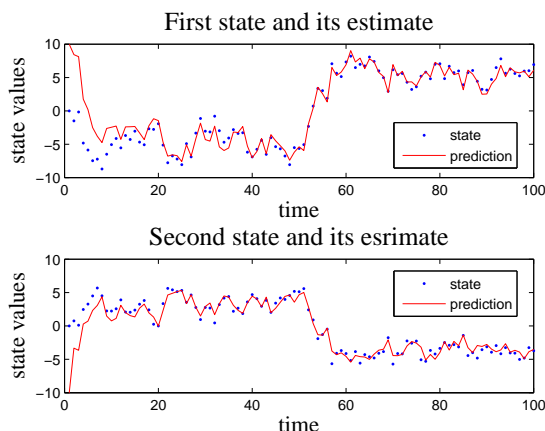
sx a sy jsou směrodatné odchylky¹⁰ kovariancí šumu.

- simulujte data z uvedeného stavového modelu,
- s pomocí Kalmanova filtru odhadněte simulovaný stav na daném časovém intervalu,
- simulovaný a odhadnutý stav porovnejte.

Řešení příkladu je uděláno v programu, který je uveden v části Programy, na straně 133.

Výsledky příkladu jsou na Obrázku 7.1. Tečkovaná křivka je průběh simulovaného stavu, plná čára je jeho bodový odhad. Horní graf ukazuje první složku stavu, spodní druhou.

¹⁰Směrodatná odchylka kovarianční matice C je definována tzv. L'DL rozkladem této matice, tj. jako horní trojúhelníková matice L' s jedničkami na diagonále, pro kterou platí $C = L'DL$, kde D je diagonální matice s nezápornými prvky na diagonále.



Obrázek 7.1: Odhad stavu pomocí Kalmanova filtru

Z grafů je patrné, že počáteční odchylka odhadu od stavu způsobená nesprávnou hodnotou počátečního odhadu rychle vymizí a odhady jsou v rámci přítomného šumu velmi přesné. Musíme mít ovšem na paměti, že stavový model má nastaveny správné parametry (parametry modelu se předpokládají známé).

◁

7.4.2 Poznámka k příkladu

Kalmanův filtr startuje s počátečním stavem $x_{1|0}$ a počáteční kovariancí $R_{1|0}$. Stav $x_{t|}$ a kovariance odhadu stavu $R_{t|}$ se během výpočtů přepočítávají. Matice r_x a r_y jsou kovariance šumů ze stavové a výstupní rovnice modelu. (Pozor!!! Nejsou to kovariance stavu a výstupu, ale kovariance šumů!!!). Ačkoli Kalmanův filtr pracuje se známým modelem, tyto kovariance většinou neznáme a musíme je nahradit jejich odhadem. A právě na správné nastavení těchto dvou kovariancí je další běh Kalmanova filtru velmi citlivý. Kovarianci $R_{1|0}$ lze nastavit jako diagonální s velkými čísly na diagonále (asi tak 10^3). Čím jsou tato čísla větší, tím méně důvěry vkládáme do našich počátečních odhadů stavu a tím rapidněji se tyto odhady na začátku mohou měnit. Ověřte si tuto skutečnost sami v předloženém příkladě. ◁

Další doporučené pokusy jsou:

- odhad z dat simulovaných bez a se šumem (nastaví se pomocí směrodatných odchylek r_w a r_v),
- odhad z jiné soustavy (změna parametrů - matic M, N, A),
- start odhadu s různými počátečními odhady stavu (volba $x_{1|0}$).

8 Řízení se spojitým modelem

V této kapitole se stručně zmíníme o úloze optimálního řízení s popisem soustavy ve formě regresního modelu. Jestliže chceme hovořit o optimálním řízení, musíme nejprve definovat kritérium optimality. Protože pracujeme s veličinami ve formě náhodných procesů, je třeba kritérium

psát jako střední hodnotu podmíněnou tím, co je nám v daném okamžiku k dispozici - apriorní znalost, tedy kritérium má tvar

$$E \left[\sum_{t=1}^N J_t | d(0) \right],$$

kde J_t je tzv. penalizační funkce, hodnotící řízení v okamžiku t . Nejčastěji se volí kvadratická penalizační funkce $J_t = y_t^2 + \omega u_t^2$, která řídí výstup na nulu a částečně pokutuje velké hodnoty řídicí veličiny.

Hodnoty tohoto kritéria na celém intervalu řízení, tj. pro $t = 1, 2, \dots, N$, nám umožní porovnávat různé způsoby řízení. Optimální řízení je takové, pro které je hodnota kritéria minimální.

Dynamické programování

Obecně lze ukázat, že optimální řídicí veličiny lze počítat rekurzivně odzadu (od konce intervalu řízení) aplikací následujících dvou kroků.

1. Výpočet střední hodnoty

$$\varphi_t = E [J_t + \varphi_{t+1}^* | u_t, d(t-1)], \quad (8.1)$$

kde J_t označuje část kritéria J odpovídající času $J_t = y_t^2 + \omega u_t^2$.

2. Minimalizace

$$\varphi_t^* = \min_{u_t} \{\varphi_t\}, \quad (8.2)$$

přičemž u_t , pro které je dosaženo minima, je optimálním řízením v kroku t .

Rekurze startuje na konci intervalu řízení, obecně pro $t = N$ s koncovou podmínkou $\varphi_{N+1}^* = 0$.

Postup syntézy řízení ukážeme v následujícím příkladě.

Příklad

Uvažujeme následující zadání úlohy: Je dáno kvadratické kritérium optimality a model soustavy.

Kritérium

$$J = E \left[\sum_{t=1}^3 (y_t^2 + \omega u_t^2) | d(0) \right].$$

Model

$$y_t = ay_{t-1} + bu_t + e_t,$$

kde a, b jsou známé parametry modelu, $e_t \sim N(0, r)$ je šum se známým rozptylem r je a y_0 je známá počáteční podmínka.

Cílem je navrhnout řídicí strategii u_1, u_2, u_3 tak, aby bylo dosaženo minima kritéria J .

V tomto příkladě, kde $N = 3$, budeme postupně provádět oba zmíněné kroky. Pro $t = 3$ je $\varphi_4^* = 0$.

⇒ Čas $t = 3$:

$$\begin{aligned}\varphi_3 &= E [y_3^2 + \omega u_3^2 + \varphi_4^* | u_3, d(2)] = E [(ay_2 + bu_3 + e_3)^2 + \omega u_3^2 | u_3, d(2)] = \\ &= (ay_2 + bu_3)^2 + r + \omega u_3^2 = (\omega + b^2) \left[u_3 + \frac{ab}{\omega + b^2} y_2 \right]^2 + \frac{\omega a^2}{\omega + b^2} y_2^2 + r,\end{aligned}$$

kde poslední úprava byla dosažena doplněním na čtverec v proměnné u_3 - viz ...

Minimalizace je velmi jednoduchá. Minima dosáhneme anulováním hranaté závorky (ta nemůže být menší než nula a výraz za ní je na u_3 nezávislý)

$$u_3 = -\frac{ab}{b^2 + \omega} y_2 = U_3 y_2,$$

kde $U_3 = -\frac{ab}{b^2 + \omega}$.

Zbytek po minimalizaci dostaneme, jestliže za hranatou závorku dosadíme nulu

$$\varphi_3^* = \frac{\omega a^2}{\omega + b^2} y_2^2 + r = S_3 y_2^2 + T_3,$$

kde $S_3 = \frac{\omega a^2}{\omega + b^2}$, $T_3 = r$.

⇒ Čas $t = 2$:

$$\begin{aligned}\varphi_2 &= E [y_2^2 + \omega u_2^2 + \varphi_3^* | u_2, d(1)] = E [y_2^2 + \omega u_2^2 + S_3 y_2^2 + T_3 | u_2, d(1)] = \\ &\quad \text{(dosazeno za } \varphi_3^*) \\ &= E [(1 + S_3) y_2^2 + \omega u_2^2 + T_3 | u_2, d(1)] = \\ &\quad \text{(spojeny členy s } y_2^2) \\ &= (1 + S_3) \left\{ (ay_1 + bu_2)^2 + r \right\} + \omega u_2^2 + T_3 = \\ &\quad \text{(středováno } y_2^2 \rightarrow (ay_1 + bu_2)^2 + r) \\ &= (1 + S_3) (a^2 y_1^2 + 2aby_1 u_2 + b^2 u_2^2) + (1 + S_3) r + \omega u_2^2 + T_3 = \\ &\quad \text{(umocněno)} \\ &= (1 + S_3) a^2 y_1^2 + 2(1 + S_3) aby_1 u_2 + (1 + S_3) b^2 u_2^2 + (1 + S_3) r + \omega u_2^2 + T_3 = \\ &\quad \text{(roznásobeno)} \\ &= \left\{ \omega + (1 + S_3) b^2 \right\} u_2^2 + 2(1 + S_3) aby_1 u_2 + (1 + S_3) a^2 y_1^2 + T_3 + (1 + S_3) r = \\ &\quad \text{(upraveno na kvadratický trojčlen v } u_2) \\ &= \left\{ \omega + (1 + S_3) b^2 \right\} \left[u_2^2 + 2 \frac{(1 + S_3) ab}{\omega + (1 + S_3) b^2} y_1 u_2 \right] + (1 + S_3) a^2 y_1^2 + T_3 + (1 + S_3) r =\end{aligned}$$

$$\begin{aligned}
& \text{(vytknut koeficient } \{\omega + (1 + S_3) b^2\}) \\
& = \{\omega + (1 + S_3) b^2\} \left[u_2^2 + 2 \frac{(1 + S_3) ab}{\omega + (1 + S_3) b^2} y_1 u_2 + \left(\frac{(1 + S_3) ab}{\omega + (1 + S_3) b^2} y_1 \right)^2 \right] - \\
& \quad - \frac{(1 + S_3)^2 a^2 b^2}{\omega + (1 + S_3) b^2} y_1^2 + (1 + S_3) a^2 y_1^2 + T_3 + (1 + S_3) r = \\
& \quad \text{(připraveno na doplnění na čtverec v } u_3) \\
& = \{\omega + (1 + S_3) b^2\} \left[u_2^2 + \frac{(1 + S_3) ab}{\omega + (1 + S_3) b^2} y_1 \right]^2 - \\
& \quad \frac{\omega (1 + S_3) a^2}{\omega + (1 + S_3) b^2} y_1^2 + T_3 + r = S_2 y_1^2 + T_2 \\
& \quad \text{(doplněno na čtverec v } u_3)
\end{aligned}$$

Odtud plyne

$$u_2 = -\frac{(1 + S_3) ab}{\omega + (1 + S_3) b^2} y_1 = U_2 y_1,$$

kde $U_2 = -\frac{(1+S_3)ab}{\omega+(1+S_3)b^2}$, a

$$\varphi_2^* = S_2 y_1^2 + T_2,$$

kde $S_2 = \frac{\omega(1+S_3)a^2}{\omega+(1+S_3)b^2}$, $T_2 = T_3 + (1 + S_3) r$.

Zbývá dopočítat řízení pro poslední čas $t = 1$. To již ale nemusíme odvozovat. Stačí, když porovnáme výsledky z obou předchozích časů. Zjistíme, že můžeme psát obecně následující algoritmus:

Algoritmus dynamického programování pro regresní model 1. řádu

$$u_t = -\frac{(1 + S_{t+1}) ab}{\omega + (1 + S_{t+1}) b^2} y_{t-1}, \quad (8.3)$$

$$\varphi_t^* = S_t y_{t-1}^2 + T_t, \quad (8.4)$$

kde

$$S_t = \frac{\omega (1 + S_{t+1}) a^2}{\omega + (1 + S_{t+1}) b^2}, \quad T_t = T_{t+1} + (1 + S_{t+1}) r. \quad (8.5)$$

Rekurze postupuje proti času a startuje v čase N (tady $N = 3$) s hodnotou $S_{N+1} = 0$, odkud již vyjde $T_{N+1} = 1$.

Dokončíme příklad podle odvozené rekurze

\Rightarrow Čas $t = 1$:

$$u_1 = -\frac{(1 + S_2) ab}{\omega + (1 + S_2) b^2} y_0 = U_1 y_0,$$

$$\varphi_1^* = S_1 y_0^2 + T_1,$$

kde

$$S_1 = \frac{\omega(1+S_2)a^2}{\omega+(1+S_2)b^2}, \quad T_1 = T_2 + (1+S_2)r$$

a S_2, T_2 byly vypočteny v kroku $t = 2$.

Tím jsme vypočítali koeficienty U_3, U_2, U_1 , ale řízení jsme zatím nemohli realizovat, protože jsme v čase $t = 0$ a hodnoty výstupu y_2 a y_1 zatím neznáme. Když jsme se dostali do času $t = 1$, závisí řízení na hodnotě y_0 , kterou už známe. Můžeme tedy začít s řízením v reálném čase. Spočteme řízení $u_1 = U_1 y_0$ a aplikujeme je. V další periodě pro $t = 2$ změříme výstup y_1 (jako odezvu na aplikované u_1) a můžeme počítat $u_2 = U_2 y_1$, které opět aplikujeme. Nakonec pro $t = 3$ změříme y_2 , vypočteme $u_3 = U_3 y_2$ a aplikujeme jej. Tím jsme splnili zadání a optimálně řídili na intervalu řízení pro $t = 1, 2, 3$.

Výsledky, které jsme odvodili, jsou obecné a platí pro libovolně dlouhý interval řízení.

Ukázka programu na toto téma je v kapitole Programy, na str. 90.

9 Řízení s diskretním modelem

Jak jsme již uvedli v Kapitole 3, obecný popis diskretního dynamického systému dostaneme tak, že každé konfiguraci hodnot jeho datového vektoru $d_t = [y_t, \psi_t]'$ přiřadíme přímo pravděpodobnost, tj.

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

V této kapitole budeme demonstrovat algoritmus dynamického programování s diskretním modelem. Pro větší přehlednost provedeme výklad pro konkrétní jednoduchý příklad.

Je dán model

$$f(y_t | [u_t, y_{t-1}], \Theta) = \Theta_{y_t | [u_t, y_{t-1}]},$$

kde $[u_t, y_{t-1}] = \psi_t$ a $y_t, u_t \in \{1, 2\}$, $\forall t$.

Kritérium pro řízení budeme definovat podobně jako u modelu pro každou konfiguraci datového vektoru zvlášť jako nezáporné číslo

$$\{J_{y_t | [u_t, y_{t-1}]} \geq 0\}_{y_t, u_t, y_{t-1} \in \{1, 2\}}.$$

Pro představu lze jak model, tak i kritérium zapsat ve formě tabulky:

Model

$$f(y_t | [u_t, y_{t-1}], \Theta)$$

| u_t, y_{t-1} | $y_t = 1$ | $y_t = 2$ |
|----------------|-----------------|-----------------|
| 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 | $\Theta_{1 22}$ | $\Theta_{2 22}$ |

Kritérium

$$J$$

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

Určete optimální strategii řízení u_1, u_2, u_3 , která minimalizuje zvolené kritérium řízení.

Pro řešení použijeme algoritmus (8.1) – (8.2) odvozený v minulé kapitole.

Zvolíme model a kritérium v konkrétním tvaru

| $f(y_t [u_t, y_{t-1}], \Theta)$ | J | |
|-----------------------------------|-----------|-----------|
| u_t, y_{t-1} | $y_t = 1$ | $y_t = 2$ |
| 1,1 | 0.7 | 0.3 |
| 1,2 | 0.2 | 0.8 |
| 2,1 | 0.9 | 0.1 |
| 2,2 | 0.4 | 0.6 |

| J | J | |
|----------------|-----------|-----------|
| u_t, y_{t-1} | $y_t = 1$ | $y_t = 2$ |
| 1,1 | 0 | 1 |
| 1,2 | 1 | 0 |
| 2,1 | 1 | 2 |
| 2,2 | 2 | 1 |

kde je v kritériu vyjádřena skutečnost, že nechceme změny ve výstupu a pro řízení preferujeme hodnotu 1. Řízení budeme opět počítat na horizontu $N = 3$.

Pro $t = N = 3$ je podle (8.1) nutné udělat středování

$$\varphi_3(u_3, y_2) = E[J|u_3, d(2)] = \sum_{y_3=1}^2 J_{y_3|u_3 y_2} \Theta_{y_3|u_3 y_2} = J_{1|u_3 y_2} \Theta_{1|u_3 y_2} + J_{2|u_3 y_2} \Theta_{2|u_3 y_2} =$$

| u_3, y_2 | φ_3 |
|------------|-------------------------------------|
| 1,1 | $0 \times 0.7 + 1 \times 0.3 = 0.3$ |
| 1,2 | $1 \times 0.2 + 0 \times 0.8 = 0.2$ |
| 2,1 | $1 \times 0.9 + 2 \times 0.1 = 1.1$ |
| 2,2 | $2 \times 0.4 + 1 \times 0.6 = 1.4$ |

a podle (8.2) minimalizace

$$\varphi_3^* = \min_{u_3} \varphi_3 = \begin{cases} 0.3 & \text{pro } y_2 = 1 \text{ (při } u_3 = 1) \\ 0.2 & \text{pro } y_2 = 2 \text{ (při } u_3 = 1) \end{cases}$$

\mathcal{S}_3 : Optimální řízení je tedy $u_3 = 1$, a to jak pro $y_2 = 1$, tak i pro $y_2 = 2$. Tím je optimalizace v kroku 3 hotova a přecházíme na krok 2.

Pro $t = N - 1 = 2$ potřebujeme zbytek φ_3^* ve formě tabulky. Regresní vektor v kroku 2 bude $[u_2, y_1]$, φ_3^* závisí jen na y_2 (tj. bude pro všechny kombinace hodnot regresního vektoru stejné), takže tabulka bude

| φ_3^* | | |
|---------------|-----------|-----------|
| u_2, y_1 | $y_2 = 1$ | $y_2 = 2$ |
| 1,1 | 0.3 | 0.2 |
| 1,2 | 0.3 | 0.2 |
| 2,1 | 0.3 | 0.2 |
| 2,2 | 0.3 | 0.2 |

Prvním krokem je opět středování pře y_2

$$\begin{aligned} \varphi_2 &= E[J + \varphi_3^* | u_2, d(1)] = \sum_{y_2=1}^2 (J_{y_2|u_2y_1} + \varphi_3^*) \Theta_{y_2|u_2y_1} = \\ &= (J_{1|u_2y_1} + \varphi_3^*) \Theta_{1|u_2y_1} + (J_{2|u_2y_1} + \varphi_3^*) \Theta_{2|u_2y_1} = \end{aligned}$$

| u_2, y_1 | φ_2 |
|------------|-------------|
| 1,1 | 0.8 |
| 1,2 | 0.7 |
| 2,1 | 1.6 |
| 2,2 | 1.9 |

$\mathcal{S}_2 : y_1 = 1$ je v řádku 1 a 3, v řádku 1 je menší hodnota kriteria \rightarrow pro $y_1 = 1$ volíme $u_2 = 1$.
Pro $y_2 = 2$ je minimum v řádku 2, a tedy volíme také $u_2 = 1$.

$$\varphi_2^* = \begin{cases} 0.8 & \text{pro } y_2 = 1, \\ 0.7 & \text{pro } y_2 = 2. \end{cases}$$

V čase $t = 1$ je

| φ_2^* | | |
|---------------|-----------|--------------|
| u_1, y_0 | $y_1 = 1$ | $y_{21} = 2$ |
| 1,1 | 0.8 | 0.7 |
| 1,2 | 0.8 | 0.7 |
| 2,1 | 0.8 | 0.7 |
| 2,2 | 0.8 | 0.7 |

a

| u_1, y_0 | φ_1 |
|------------|-------------|
| 1,1 | 1.8 |
| 1,2 | 1.7 |
| 2,1 | 2.6 |
| 2,2 | 2.9 |

$\mathcal{S}_1 : \text{odkud } u_1 = 1 \text{ jak pro } y_0 = 1 \text{ tak i pro } y_0 = 2.$

Strategie řízení je dána v bodech $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$.

Protože y_0 známe, můžeme určit u_1 , aplikovat, změřit y_1 a tak dále.

10 Přílohy

10.1 Doplnění na čtverec

Používá se pro minimalizaci kvadratického výrazu nebo pro integraci v konvoluci dvou normálních rozdělení (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.

10.2 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 popsáním 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

$$f(\Theta|u_t, d(t-1)) = f(\Theta|d(t-1)). \quad (10.1)$$

Obrácený vztah lze odvodit obdobnou úvahou, nebo jej lze pomocí Bayesova vzorce odvodit z (10.1)

$$\begin{aligned} f(u_t|d(t-1), \Theta) &= \underbrace{f(\Theta|u_t, d(t-1)) \frac{f(u_t|d(t-1))}{f(\Theta|d(t-1))}}_{\text{Bayes}} = \\ &= \underbrace{f(\Theta|d(t-1)) \frac{f(u_t|d(t-1))}{f(\Theta|d(t-1))}}_{\text{podle (10.1)}} = f(u_t|d(t-1)). \end{aligned}$$

Podobné úvahy lze místo o parametru vést i o odhadovaném stavu.

10.3 Bayesův vzorec

10.3.1 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

$$f(B|A, C) = \frac{f(A|B, C) f(B|C)}{f(A|C)}. \quad (10.2)$$

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 (10.2) 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.

10.3.2 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 (4.1) 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 **přírozené podmínky řízení** (10.1), 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 $d(t-1)$.

10.4 Multinomiální rozdělení

Multinomiální rozdělení popisuje diskretní 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

$$\begin{array}{c|cccc} y & 1 & 2 & \cdots & n_l \\ \hline f(y) & p_1 & p_2 & \cdots & p_{n_l} \end{array},$$

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 diskretního 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 svíslítkem). Proto musí platit nezápornost $\Theta_{i|jk} \geq 0, \forall i, j, k$ a dále součet parametrů pro každou konfiguraci podmínky musí dát jedničku $\sum_{i=1}^2 \Theta_{i|jk} = 1, \forall j, k$.¹¹ Pro účely odhadu je výhodné formálně tento model vyjádřit v tzv. **součinném tvaru**

$$f(y|\psi, \Theta) = \prod_{i \in y^*} \prod_{\varphi \in \psi^*} \Theta_{i|\varphi}^{\delta(i|\varphi, y|\psi)}, \quad (10.3)$$

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 součinného tvaru je skutečně formální, protože cokoli na nulu je jedna a po vynásobení zůstane jen $\Theta_{y|\psi}$.

10.5 Dirichletovo rozdělení

Konjugovaným rozdělením¹² k multinomiálnímu je rozdělení Dirichletovo

$$f(\Theta|d(t)) = \frac{1}{B(\nu_t)} \prod_{i \in y^*} \prod_{\varphi \in \psi^*} \Theta_{i|\varphi}^{\nu_{i|\varphi;t}}, \quad (10.4)$$

kde

ν_t je statistika rozdělení se stejnou strukturou jako má parametr Θ - viz tabulka v odstavci 10.4 a další značení také odpovídá značení zavedenému v tomto odstavci,

$B(\nu)$ je zobecněná beta funkce

$$B(\nu) = \prod_{\varphi \in \psi^*} \frac{\prod_{i \in y^*} \Gamma(\nu_{i|\varphi})}{\Gamma\left(\sum_{i \in y^*} \nu_{i|\varphi}\right)}, \quad (10.5)$$

kde $\Gamma(\cdot)$ je gama funkce definovaná vztahem

$$\Gamma(x) = \int_0^\infty t^{x-1} \exp(-t) dt, \quad (10.6)$$

pro kterou platí

$$\Gamma(x+1) = x\Gamma(x), \quad x \in R^+. \quad (10.7)$$

¹¹Tento požadavek je po úvaze opět zřejmý. Jestliže nastalo $u = j$ a $v = k$, má y právě dvě možnosti: $y = 1$ nebo $y = 2$.

¹²Konjugované 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.

10.6 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

$$f(y_t | \psi_t, \Theta) = \frac{1}{\sqrt{2\pi}} r^{-0.5} \exp \left\{ -\frac{1}{2r} (y_t - \psi_t' \theta)^2 \right\}. \quad (10.8)$$

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 (10.8) 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 (10.8) je s uvedenou úpravou možno zapsat takto

$$f(y_t | \psi_t, \Theta) = \frac{1}{\sqrt{2\pi}} r^{-0.5} \exp \left\{ -\frac{1}{2r} [-1 \theta'] D_t \begin{bmatrix} -1 \\ \theta \end{bmatrix} \right\}. \quad (10.9)$$

10.7 Inverzní Gauss-Wishartovo (GiW) rozdělení

Toto rozdělení vzniká jako rozdělení součinu normálních rozdělení (tj. jako rozdělení likelihoodu pro normální model). Má tvar

$$f(\Theta | d(t)) \propto r^{-0.5\kappa_t} \exp \left\{ -\frac{1}{2r} [-1 \theta'] V_t \begin{bmatrix} -1 \\ \theta \end{bmatrix} \right\}, \quad (10.10)$$

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

$$V_t = \begin{bmatrix} V_y & V_{y\psi}' \\ V_{y\psi} & V_\psi \end{bmatrix}, \quad (10.11)$$

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

$$V_t = L' D L, \quad (10.12)$$

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.

10.8 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 aposteriorní hp $f(\Theta|d(t))$, který je optimální podle kritéria

$$\min_{\hat{\Theta}_t} E \left[(\Theta - \hat{\Theta}_t)^2 | d(t) \right]. \quad (10.13)$$

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

$$\hat{\Theta}_t = E[\Theta | d(t)].$$

10.9 Bodové odhady parametrů spojitého modelu

Ukážeme odvození MAP (Maximum Aposteriori Probability) bodového odhadu který maximalizuje aposteriorní hp parametrů.

Hledáme tedy maximum aposteriorní hp (10.10)

$$\begin{aligned} f(\Theta|d(t)) &\propto r^{-0.5\kappa} \exp \left\{ -\frac{1}{2r} [-1 \theta'] V \begin{bmatrix} -1 \\ \theta \end{bmatrix} \right\} = \\ &= r^{-0.5\kappa} \exp \left\{ -\frac{1}{2r} (V_y - 2\theta' V_{y\psi} + \theta' V_\psi \theta) \right\}, \end{aligned}$$

kde jsme využili dělení matice V podle (10.11).

Nejdříve budeme hledat maximum podle θ , tj. derivovat podle θ a hledat řešení pro derivaci rovnu nule.¹³

$$\frac{\partial f(\{\theta, r\} | d(t), r)}{\partial \theta} \propto r^{-0.5\kappa} \exp \left\{ -\frac{1}{2r} [-1 \theta'] V \begin{bmatrix} -1 \\ \theta \end{bmatrix} \right\} \left(\frac{-1}{2r} \right) (-2V_{y\psi} + 2V_\psi \theta) = 0.$$

Odtud pochází vztah

$$\hat{\theta} = V_\psi^{-1} V_{y\psi}. \quad (10.14)$$

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

$$\Lambda = V_y - V_{y\psi}' V_\psi^{-1} V_{y\psi}. \quad (10.15)$$

Aposteriorní hp s dosazeným bodovým odhadem (10.14) je

$$f(r|d(t)) \propto r^{-0.5\kappa} \exp \left\{ -\frac{\Lambda}{2r} \right\}.$$

Derivujeme a položíme rovno nule

$$-\kappa \frac{1}{2r} + \Lambda \frac{1}{r^2} = 0,$$

a tedy dostaneme

$$\hat{r} = \frac{\Lambda}{\kappa}. \quad (10.16)$$

$\hat{\theta}$ a \hat{r} jsou bodové odhady, které tady hledáme.

¹³Derivujeme vektory podle vektorů. Správnost lze ověřit derivováním podle složek a zpětným sestavením do vektorového tvaru.

10.10 Bodové odhady parametrů diskrétního modelu

Bodové odhady parametru Θ multinomiálního rozdělení (??) dostaneme pouhou normalizací statistiky ν_t tak, aby součty jejích prvků v řádcích jejího maticového vyjádření (jako pro Θ v odstavci (??)) byly rovny jedné, tj.

$$\hat{\Theta}_{y|\psi;t} = \frac{\nu_{y|\psi;t}}{\sum_{i \in y^*} \nu_{i|\psi;t}}, \quad \forall y \in y^* \text{ a } \psi \in \psi^*. \quad (10.17)$$

Tento bodový odhad jsme určili jako podmíněnou střední hodnotu parametru s rozdělením podle aposteriori hp (10.4) - 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 aposteriori hp z (10.4) a beta funkce B je dána v (10.5). Parametr $\Theta_{y|\varphi;t}$ formálně vyjádříme v součinném tvaru (10.3)

$$\Theta_{y|\psi} = \prod_{i \in y^*} \prod_{\varphi \in \psi^*} \Theta_{i|\varphi}^{\delta(i|\varphi, y|\psi)}$$

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

$$B(\nu_\varphi) = \frac{\prod_{i \in y^*} \Gamma(\nu_{i|\varphi})}{\Gamma(\sum_{i \in y^*} \nu_{i|\varphi})} \text{ podle (10.5)}$$

jsme využili předpoklad o nezávislosti parametrů mezi různými konfiguracemi regresního vektoru ψ , tj nezávislosti řádků v tabulkovém vyjádření parametru v odstavci 10.4.

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 (10.7). Tím jsme dokázali vztah (10.17).

10.11 Logistická regrese

10.11.1 Derivace věrohodnostní funkce

Derivace logaritmu věrohodnostní funkce $\ln L$ s modelem (3.2) 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 (3.3) 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 metodu¹⁴

10.11.2 Newtonův algoritmus

Pomocí Newtonova algoritmu je možné numericky hledat extrém¹⁵ nelineárních funkcí. Označme takovou funkci $g(x)$, kde $x = [x_1, x_2 \cdots x_n]'$. Dále označme gradient této funkce

$$g'(x) = \begin{bmatrix} \frac{\partial g}{\partial x_1} \\ \frac{\partial g}{\partial x_2} \\ \dots \\ \frac{\partial g}{\partial x_n} \end{bmatrix}$$

¹⁴Výhodné je to zejména pro to, že se nám podařilo analyticky spočítat jak první, tak i druhou derivaci maximalizované funkce $\ln L$.

¹⁵Pozor! Jde samozřejmě o lokální extrém.

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.

10.12 Rozšířený Kalmanův filtr

10.12.1 Konstrukce rozšířeného KF

Máme nelineární stavový model

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

Poslední bodový odhad stavu x_t označíme \hat{x}_t .

Linearizaci funkcí g a h z modelu provedeme rozvojem do Taylorovy řady a zanedbáním členů druhého a vyšších řádů. Dostaneme

$$\begin{aligned} g(x_t, u_t) &\doteq g(\hat{x}_t, u_t) + g'(\hat{x}_t, u_t)(x_t - \hat{x}_t) = \\ &= g'(\hat{x}_t, u_t)x_t + [g(\hat{x}_t, u_t) - g'(\hat{x}_t, u_t)\hat{x}_t], \end{aligned} \quad (10.18)$$

$$\begin{aligned} h(x_t, u_t) &\doteq h(\hat{x}_t, u_t) + h'(\hat{x}_t, u_t)(x_t - \hat{x}_t) = \\ &= h'(\hat{x}_t, u_t)x_t + [h(\hat{x}_t, u_t) - h'(\hat{x}_t, u_t)\hat{x}_t]. \end{aligned} \quad (10.19)$$

Dosazením do modelu dostaneme jeho linearizovanou verzi

$$\begin{aligned} x_{t+1} &\doteq g'(\hat{x}_t, u_t)x_t + [g(\hat{x}_t, u_t) - g'(\hat{x}_t, u_t)\hat{x}_t] + w_t, \\ y_t &\doteq h'(\hat{x}_t, u_t)x_t + [h(\hat{x}_t, u_t) - h'(\hat{x}_t, u_t)\hat{x}_t] + v_t \end{aligned}$$

a s označením

$$Q_x = [g(\hat{x}_t, u_t) - g'(\hat{x}_t, u_t)\hat{x}_t], \quad Q_y = [h(\hat{x}_t, u_t) - h'(\hat{x}_t, u_t)\hat{x}_t],$$

dostaneme lineární tvar modelu

$$x_{t+1} = g'(\hat{x}_t, u_t)x_t + Q_x + w_t, \quad (10.20)$$

$$y_t = h'(\hat{x}_t, u_t)x_t + Q_y + v_t. \quad (10.21)$$

Pro tento model již lze použít obyčejný Kalmanův filtr. Je ale potřeba si uvědomit, že bodový odhad \hat{x}_t bude v každé rovnici jiný. V daném kroku KF se nejprve provádí filtrace, které odpovídá druhá (výstupní) rovnice. Pro ni bude posledním odhadem $x_{t|t-1}$, který pochází z minulého kroku z predikce. Pro první (stavovou) rovnici bude posledním odhadem $\hat{x}_t = x_{t|t}$, který jsme obdrželi v současném kroku KF z filtrace. Odtud je patrné, že proceduru realizující KF je pro nelineární filtraci třeba rozdělit do dvou částí - do filtrace a predikce - a obě části realizovat samostatně. Po filtraci, která končí uprostřed algoritmu, je totiž třeba přepočítat funkce g' a Q_x ze stavové rovnice pro poslední odhad stavu $x_{t|t}$.

10.12.2 Realizace KF

Obecně jsme konstatovali, že metoda rozšířeného KF převede nelineární stavový model na lineární tak, že nelineární funkce na pravých stranách modelů rozvine do nultého a prvního členu Taylorova rozvoje. Výsledkem je linearizovaný tvar modelu (10.20), (10.21). Pro tento model je již možno použít proceduru standardního KF, ovšem pro model s konstantou. Ve skutečnosti je ale možno počítat takto:

1. Střední hodnoty odhadu stavu a predikce výstupu je možno počítat z původního nerozšířeného modelu. Důvod je tento: z rovnic (10.18) a (10.19) je vidět, že derivace g' a h' jsou násobeny přírůstkem stavu $(\xi_t - \hat{\xi}_t)$, kde $\hat{\xi}_t$ je bod, v němž provádíme Taylorův rozvoj, což je poslední bodový odhad stavu. Stejný bodový odhad je ale dosazen za ξ_t a pomocí něho se počítá příští bodový odhad v algoritmu KF. Přírůstek tedy bude $(\hat{\xi}_t - \hat{\xi}_t) = 0$. Proto se při výpočtu střední hodnoty původního (nerozšířeného) stavu uplatní jen absolutní člen rozvoje, což je pravá strana původního stavového modelu. Část rozšířeného stavu, která obsahuje neznámé parametry, zůstává při výpočtu střední hodnoty stejná.
2. Konstanta Q_x nebo Q_y se při výpočtu kovariancí neuplatní. Důvod ukážeme ve zjednodušeném případě na výpočtu rozptylu z náhodné veličiny X plus konstanta a

$$\begin{aligned} D[X + a] &= E[(X + a - E[X + a])^2] = E[(X + a - E[X] - a)^2] = \\ &= E[(X - E[X])^2] = D[X]. \end{aligned}$$

Rozptyl náhodné veličiny $X + a$ a náhodné veličiny X je stejný. Podobně to v obecném případě platí i pro kovariance.

Realizace rozšířeného KF je tedy následující:

1. Z minulého kroku máme "poslední" odhad stavu $\hat{\xi}_t = \xi_{t|t-1}$.
2. Vezmeme tu část odhadu, která odpovídá původnímu stavu, a podle původní výstupní rovnice

$$y_t = Ax_t$$

spočteme predikci výstupu \hat{y}_t .

3. Kovariance počítáme podle matic rozšířené výstupní rovnice

$$y_t = \tilde{A}^d \xi_t$$

s vynechanou konstantou Q_y .

4. Přepočteme (filtrace) $\xi_{t|t-1} \rightarrow \xi_{t|t}$
Pozn.: K vývoji odhadovaných parametrů dochází jen v této části algoritmu.

5. Za poslední odhad stavu dosadíme $\hat{\xi}_t = \xi_{t|t}$.

6. Kovariance přepočteme s maticemi rozšířeného stavového modelu bez konstanty

$$\xi_{t+1} = \tilde{M}^d \xi_t.$$

7. Přepočet (predikce) stavu provedeme podle původní stavové rovnice

$$x_{t+1} = Mx_t,$$

přičemž část stavu ξ_t odpovídající neznámým parametrům zůstává nezměněna.

11 Programy

11.1 Diferenciální a Diferenční rovnice

11.1.1 Diferenciální rovnice

Při modelování stejně jako i v dalších úlohách se pohybujeme výhradně v diskrétním čase. Důvodem k tomu je především použití počítačů, které realizují odvozené algoritmy přirozeně v opakovaných časových okamžicích s danou periodou. Nicméně čas v reálném světě se tváří spjitě a diskrétní čas vzniká uměle - vzorkováním.

Spojité děje jsou modelovány diferenciálními rovnicemi. Nejčastěji se snažíme pro modelování použít rovnice s konstantními koeficienty. Pokud jde o popis neřízeného systému, např. odezvy na počáteční podmínku, použijeme homogenní rovnici. Řízení se objeví jako pravá strana rovnice. Řešení, odezva systému, je pak založeno na řešení homogenní rovnice a lze použít např. metodu variace konstanty. Základem je ale řešení homogenní rovnice, kterého si blíže povšimneme pro rovnice 1. a 2. řádu.

V první fázi se řeší charakteristická rovnice a řešení diferenciální rovnice se odhadne ve vhodném (exponenciálním) tvaru.

V druhé fázi se dopočítají volné koeficienty tak, aby řešení vyhovovalo zadaným počátečním podmínkám. Pro druhý řád diferenciální rovnice je charakteristická rovnice kvadratická a podle typu jejího řešení dostáváme určitý typ řešení diferenciální rovnice.

Toto jsou všechno známé věci. Nicméně, pro více podrobností je možno nahlédnout do kapitoly [11.9](#).

Analytické i numerické řešení a jejich porovnání je dáno v programech T01dif2p1.m, T02dif2p2.m a T03dif2p3.m.

```

// Differential equation of the 2nd order
// and its solution (two real roots)
exec("ScIntro.sce",-1)

a1=5; a0=6; // coefficients y''+a1.y'+a0.y
y0=5; y1=5; // initial value at times 0 and -1
nt=10; // number of steps
h=.001; // numerical step

b1=2-a1*h; // numerical realization
b2=-1+a1*h-a0*h^2; // derivatives replaced by differences

// analytical solution (for one multiple root)
tt=h:h:nt; // dense time for numerics
z=15*exp(-2*tt)-10*exp(-3*tt); // analytical response

// numerical solution (for one multiple root)
y=[y0 y1]; // initial conditions
j=2; // counter of vector items
for t=tt // loop of numerical computation
    j=j+1;
    y(j)=b1*y(j-1)+b2*y(j-2); // numerically discretized equation
end

// results
plot(tt,y(3:$),tt,z)
set(gcf(),"position",[700 100 600 500])
legend('num. sol.','anal.sol.')

```

```

// Differential equation of the 2nd order
// and its solution (one multiple root)
exec("ScIntro.sce",-1)

a1=2; a0=1; // coefficients y''+a1.y'+a0.y
y0=5; y1=5; // initial value at times 0 and -1
nt=10; // number of steps
h=.001; // numerical step

b1=2-a1*h; // numerical realization
b2=-1+a1*h-a0*h^2; // derivatives replaced by differences

// analytical solution (for one multiple root)
tt=h:h:nt; // dense time for numerics
z=(5+5*tt).*exp(-tt); // analytical response

// numerical solution (for one multiple root)
y=[y0 y1]; // initial conditions

```

```

j=2; // counter of vector items
for t=tt // loop of numerical computation
    j=j+1;
    y(j)=b1*y(j-1)+b2*y(j-2); // numerically discretized equation
end

// results
plot(tt,y(3:$),tt,z)
set(gcf(),"position",[700 100 600 500])
legend('num. sol.','anal.sol.')

// Differential equation of the 2nd order
// and its solution (two complex roots)
exec("ScIntro.sce",-1)

a1=4; a0=13; // coefficients y''+a1.y'+a0.y
y0=5; y1=5; // initial value at times 0 and -1
nt=10; // number of steps
h=.001; // numerical step

b1=2-a1*h; // numerical realization
b2=-1+a1*h-a0*h^2; // derivatives replaced by differences

// analytical solution (for one multiple root)
tt=h:h:nt; // dense time for numerics
z=exp(-2*tt).*(5*cos(3*tt)+10/3*sin(3*tt)); // analytical response

// numerical solution (for one multiple root)
y=[y0 y1]; // initial conditions
j=2; // counter of vector items
for t=tt // loop of numerical computation
    j=j+1;
    y(j)=b1*y(j-1)+b2*y(j-2); // numerically discretized equation
end

// results
plot(tt,y(3:$),tt,z)
set(gcf(),"position",[700 100 600 500])
legend('num. sol.','anal.sol.')

```

11.1.2 Diferenční rovnice

Stejně jako u diferenciálních rovnic tak i tady si všimneme homogenních diferenčních rovnic 1. a 2. řádu. Pro ně opět odhadneme řešení obecné rovnice a dopočítáme volné parametry podle počátečních podmínek. Pro konkrétní řešení je možno nahlédnout do kapitoly [11.9](#).

Analytické i numerické řešení a jejich porovnání je dáno v programech T04dfc2p1.m T05dfc2p2.m a T06dfc2p3.m.

```
// Difference equation of the 2nd order
// and its solution (two real roots)
exec("ScIntro.sce",-1)

lam1=.2; lam2=.3; // solution to the characteristic equation
y0=5; y1=5; // initial value at times 0 and -1
nt=10; // number of steps

a1=-(lam1+lam2); // coefficients of the dfce equation
a0=lam1*lam2; // y(t+2)+a1.y(t+1)+a0.y(t)

// ANALYTICAL SOLUTION (FOR TWO DIFFERENT ROOTS)
tau=1:nt; // discrete time for simulation
oo=ones(tau);
c1=(y1-lam2*y0)/(lam1-lam2);
c2=(y1-lam1*y0)/(lam2-lam1);
z=c1*(lam1*oo).^tau+c2*(lam2*oo).^tau; // analytical response

// NUMERICAL (RECURSIVE) SOLUTION
y=[y0 y1]; // initial conditions
for t=1:nt // loop of numerical computation
    y(t+2)=-a1*y(t+1)-a0*y(t); // numerically discretized equation
end

// RESULTS
s=2:nt+1;
plot(s,y(s),s,z,'o')
set(gcf(),"position",[700 100 600 500])
legend('num. sol.','anal.sol.')

// Difference equation of the 2nd order
// and its solution (two real roots)
exec("ScIntro.sce",-1)

lam1=.2; lam2=.2; // solution to the characteristic equation
y0=5; y1=5; // initial value at times 0 and -1
nt=10; // number of steps

a1=-(lam1+lam2); // coefficients of the dfce equation
a0=lam1*lam2; // y(t+2)+a1.y(t+1)+a0.y(t)

// ANALYTICAL SOLUTION (FOR TWO DIFFERENT ROOTS)
```

```

tau=1:nt; // discrete time for simulation
oo=ones(tau);
c1=y0;
c2=y1/lam1-y0;
z=c1*(lam1*oo).^tau+c2*tau.*(lam2*oo).^tau; // analytical response

// NUMERICAL (RECURSIVE) SOLUTION
y=[y0 y1]; // initial conditions
for t=1:nt // loop of numerical computation
    y(t+2)=-a1*y(t+1)-a0*y(t); // numerically discretized equation
end

// RESULTS
s=2:nt+1;
plot(s,y(s),s,z,'o')
set(gcf(),"position",[700 100 600 500])
legend('num. sol.','anal.sol.')

// Difference equation of the 2nd order
// and its solution (two real roots)
exec("ScIntro.sce",-1)

setting=1; // type 1 = set coefficients, 2 = set roots of char. eq.

if setting==2
    // setting of the coefficients of dfc equation
    a1=-.4; a0=.29; // coefficients of the dfce equation
    if a1^2-4*a0>=0, // check if roots are complex
        disp('The roots are not complex')
    return
end
re=-a1/2; im=sqrt(-a1^2+4*a0)/2; // real and imaginair parts of the roots
lam1=re-im*i; lam2=re+im*i; // solution to the characteristic equation
else
    // setting of roots of dfc equation
    re=-.2; im=.5;
    lam1=-re-im*i; lam2=-re+im*i; // solution to char. equation
    a1=-(lam1+lam2); // coefficients of the dfce equation
    a0=lam1*lam2; // y(t+2)+a1.y(t+1)+a0.y(t)
end

y0=5; y1=5; // initial value at times 0 and -1
nt=10; // number of steps

r=sqrt(re^2+im^2); // radius of the root
th=atan(im/re); // angle of the root

```

```

// ANALYTICAL SOLUTION (FOR TWO DIFFERENT ROOTS)
tau=1:nt; // discrete time for simulation
oo=ones(tau);
c1=y0; // constant
c2=(y1/r-y0*cos(th))/sin(th); // constant
A=(r*ones(1,nt)).^tau; // constant
z=A.*(c1*cos(th*tau)+c2*sin(th*tau)); // analytical response

// NUMERICAL (RECURSIVE) SOLUTION
y=[y0 y1]; // initial conditions
for t=1:nt // loop of numerical computation
    y(t+2)=-a1*y(t+1)-a0*y(t); // numerically discretized equation
end

// RESULTS
s=2:nt+1;
plot(s,y(s),s,z,'o')
set(gcf(),'position',[700 100 600 500])
legend('num. sol.', 'anal. sol.')

```

11.1.3 Diskretizace

Diferenční rovnice dávají řešení ve tvaru diskrétní funkce (posloupnosti čísel indexovaných diskrétním časem). Tato diskrétní veličina je tvořena vzorky spojité odezvy, generované příslušnou diferenciální rovnicí. Tedy na diferenciální rovnici lze pohlížet jako na diskretizaci diferenciální rovnice ve smyslu shody odezvy v okamžicích vzorkování. Otázka zní: jak dostaneme diferenciální rovnici která je diskretizací dané diferenciální rovnice. Odpověď pro homogenní rovnice s konstantními koeficienty lze nalézt v kapitole [11.9](#).

Poznámka:

Pro diskretizaci te také samozřejmě možno použít Laplaceovu a Z-transformaci, stejně jako pro nalezení řešení těchto rovnic.

11.2 Simulace

11.2.1 T11simCont2

Simulace regresního modelu druhého řádu

Teorie

Simulace je úloha, při které z modelu se zadanými parametry generujeme data. Využití simulace je v podstatě dvojí

- generování dat pro jiné experimenty,
- jednokroková predikce výstupu.

Predikci lze využít např. pro vyhodnocení odhadu, kdy původní data ze kterých jsme prováděli odhad srovnáme s daty predikovanými ze stejného modelu, který jsme použili pro simulaci, ale s dosaženými bodovými odhady parametrů.

Predikci lze provést snadno tak, že model s odhadnutými parametry použijeme jako při simulaci, ale s nulovým šumem (bez šumu). Druhá alternativa pro predikci je použít model i se šumem s odhadnutou směrodatnou odchylkou.

První postup odpovídá teorii - predikce je podmíněná střední hodnota, druhá varianta kromě střední hodnoty ukazuje i neurčitost predikce.

Program

... je čitelný.

Přepínač *ui* volí typ vstupního signálu *ut*. Možno dopsat i svůj vlastní vstupní signál.

V časové smyčce se generuje výstup *yt*. Simuluje se regresní model druhého řádu s absolutním členem.

Na konci programu se uloží data *datT11* pro využití v dalších úlohách.

Doporučené pokusy

1. Zvolte svoje parametry *a*, *b*, *k* a *s* a ověřte jejich význam.
2. Zkuste různé typy vstupu - parametr *ui*.
3. Zkuste zadat parametry modelu tak, aby výstupem byla sinusovka a dále tlumená sinusovka.

Návod: Nastavte regresní model tak, aby představoval diferenční rovnici popisující sinusovku nebo tlumenou sinusovku.

```

// Simulation of the second order regression model
// -----
exec("ScIntro.sce",-1)

// PARAMETERS OF THE SIMULATION
nd=100; // length of data
iu=1; // type of control (see below)
a=[.4 .2]; // parameters at yt
b=[1 .2 -.5]; // parameters at ut
k=0; // constant (model absolute term)
s=.1; // noise variance

yt=zeros(1,nd);
yt(1)=1; yt(2)=3; // initial conditions for output
// selection of the type of control
select iu
  case 1, ut=rand(1,nd,'norm');
  case 2, ut=ones(1,nd);
  case 3, ut=zeros(1,nd);
  case 4, ut=[ones(1,fix(nd/2)) -1*ones(1,nd-fix(nd/2))];
  otherwise
    // write your own control
end

// TIME LOOP OF THE SIMULATION
for t=3:nd
  // regression model
  yt(t)=a*yt(t-1:-1:t-2)+b*ut(t:-1:t-2)+k+s*rand(1,1,'norm');
end

save datT11 yt ut

// RESULTS OF THE SIMULATION
subplot(211),plot(1:nd,ut),title('Control')
set(gcf(),'position',[700 100 600 500])
subplot(212),plot(1:nd,yt),title('Output')

```

11.2.2 T12simDisc

Simulace diskrétního modelu

Teorie

Diskrétní model má všechny veličiny s konečným počtem hodnot. Tyto veličiny často vyjadřují určité okolnosti při kterých sledujeme modelovanou veličinu. Může to být “denní doba” s hodnotami 1=den, 2=jitro,večer, 3=noc. Některé veličiny mohou být také diskretizovány, např. “rychlost” s hodnotami 1=normální, 2=velká.

Diskrétní model $f(y_t|y_{t-1}, u_t)$ je zadán parametrem ve formě tabulky, např.

| u_t, y_{t-1} | $y_t = 1$ | $y_t = 2$ |
|----------------|-----------|-----------|
| 1, 1 | 0.2 | 0.8 |
| 1, 2 | 0.6 | 0.4 |
| 2, 1 | 0.1 | 0.9 |
| 2, 2 | 0.3 | 0.7 |

Čísle v řádcích tabulky jsou pravděpodobnosti - každý řádek má součet jedna. Např., když zvolím $u_t = 2$ a bylo $y_{t-1} = 1$ je pravděpodobnost, že $y_t = 1$ rovna 0.1 a pro $y_t = 2$ je to doplněk, tedy $1-0.1=0.9$, protože už nic jiného nemůže nastat.

Tak například parametr

| u_t, y_{t-1} | $y_t = 1$ | $y_t = 2$ |
|----------------|-----------|-----------|
| 1, 1 | 0 | 1 |
| 1, 2 | 1 | 0 |
| 2, 1 | 0 | 1 |
| 2, 2 | 1 | 0 |

vyjadřuje deterministický systém, který bez ohledu na řízení bude stále na výstupu střídat hodnoty 1 a 2.

Program

... používá některé programovací triky:

- Příkaz $r=\text{rand}(1,\text{nd})<.3$ generuje vektor r nul a jedniček. Tam, kde je $\text{rand}<.3$ bude 1 jinak bude 0. Nula je tedy s pravděpodobností 0.7 a 1 s pravděpodobností 0.3 (podle délek intervalů .3-1 a 0-.3).

$2 - r$ převádí 0 na 2 a 1 na 1; tedy $P(1) = 0.3$ a $P(2) = 0.7$.

- Příkaz $i=2*(u-1)+y$ generuje číslo řádku v tabulce parametru - $1, 1 \rightarrow 1$, $1, 2 \rightarrow 2$, $2, 1 \rightarrow 3$, $2, 2 \rightarrow 4$.

Řízení u_t se generuje tak, že $P(u_t = 1) = 0.3$ a $P(u_t = 2) = 0.7$.

Simulace se provádí v časové smyčce, a to tak, že podle hodnot $u_t(t)$ a $y_t(t-1)$ se určí pořadí řádku v tabulce parametru a podle pravděpodobností v tomto řádku se generuje hodnota $y_t(t)$.

Na konci programu se uloží data `datT12` pro využití v dalších úlohách.

Doporučené pokusy

1. Začněte simulovat data s maximální neurčitostí (jak ve výstupu tak i ve vlivu vstupu na výstup).
2. Ubírejte neurčitost na výstupu. Pak přejděte na systém deterministický.
3. Formulujte nějakou funkci modelu a pak ji realizujte volbou parametru.

```

// Simulation of discrete model
// -----
// (multinomial model - controlled coin with memory)
// f( yt(t) | ut(t), yt(t-1) ), yt, ut=1,2
exec("ScIntro.sce",-1)

// PARAMETERS OF THE SIMULATION
// model parameter
//yt(t)=1 =2      ut(t) yt(t-1)
th= [.2 .8      // 1    1
     .6 .4      // 1    2
     .9 .1      // 2    1
     .3 .7];    // 2    2
// E.g.  if ut(t)=2 and yt(t-1)=1 then P(yt(t)=1)=.9 etc.

nd=200;          // number of steps
yt=zeros(1,nd); // declaration
ut=2-(rand(1,nd,'unif')<.3); // control variable P(ut=1)=.3, P(ut=2)=.7
yt(1)=1;        // initial condition for output

// TIME LOOP OF THE SIMULATION
for t=2:nd
    i=2*(ut(t)-1)+yt(t-1); // row in the model parameter
    yt(t)=2-(rand(1,1,'unif')<th(i,1)); // generation of the output
end

save datT12 yt ut

// RESULTS OF THE SIMULATION
subplot(211),plot(1:nd,ut,'g:.' )
set(gcf(),"position",[700 100 600 500])
title('Input')
set(gca(),'data_bounds',[0 nd+1 .9 2.1])
subplot(212),plot(1:nd,yt,'b:.' )
title('Output')
set(gca(),'data_bounds',[0 nd+1 .9 2.1])

```

11.2.3 T13simLogReg

Simulace logistického modelu

Teorie

Logistický model použijeme pro popis diskretního výstupu, který závisí jak na diskretních tak i na spojitých veličinách. Základním trikem logistického modelu je, že regresí nemodelujeme hodnoty výstupu, ale pravděpodobnosti těchto hodnot. Pro jednoduchost uvažujeme jen hodnoty 0 a 1.

Model má tvar

$$\log \frac{p_t}{1-p_t} = \psi_t' \Theta + e_t,$$

kde $p_t = P(y_t = 1 | \psi_t, \Theta)$, $1 - p_t = P(y_t = 0 | \psi_t, \Theta)$ a $\psi_t' \Theta$ je lineární regrese s regresním vektorem $\psi_t = [1, \psi_1, \psi_2, \dots, \psi_n]'$ a parametrem $\Theta = [\Theta_0, \Theta_1, \Theta_2, \dots, \Theta_n]$.

Funkce $\text{logit}(p) = \log(p/(1-p)) : p \rightarrow \text{logit}(p)$ je transformací intervalu $(0, 1)$ na celou reálnou osu.

Model funguje takto: regrese $\psi_t' \Theta + e_t$ má reálnou hodnotu. Ta se transformuje funkcí $(\text{logit})^{-1}$ na hodnotu z intervalu $(0, 1)$, což je pravděpodobnost $P(y_t = 1 | \dots)$. Je-li $\psi_t' \Theta$ velké, je $P(y_t = 1 | \dots)$ velká a bude $y_t = 1$. A naopak.

Program

... je čitelný. Používá se funkce `logit_inv`, která realizuje inverzi zmíněného vzorce, která je

$$\text{logit_inv} = \exp\{z\} / (1 + \exp(z))$$

Na konci programu se uloží data `datT13` pro využití v dalších úlohách.

Doporučené pokusy

1. Zkuste zvolit své nezávisle proměnné x_1 a x_2 a parametry th - v některých případech můžeme dostat stále stejné y_t . Co to znamená?
2. Jak dosáhneme změny neurčitosti v modelu?¹⁶

```
// Simulation of discrete output depending on mixed variables
// -----
exec("ScIntro.sce",-1)

// PARAMETERS OF THE SIMULATION
nd=100; // number of data
th=[1 .5 -2]'; // reg. coefficients (with const. at beg.)
sd=.1; // noise std.

// SIMULATION
yt=zeros(1,nd);
xt=zeros(3,nd);
// time loop
for t=1:nd
    x1=rand(1,1,'unif')+1; // continuous input
    x2=rand(1,1,'unif')>.4; // discrete input
    xt(:,t)=[1 x1 x2]'; // regression vector (with 1 at the beg.)
    z = xt(:,t)'*th +sd*rand(1,1,'norm'); // output of the regression
    py(t)=logit_inv(z); // transformation to probability
    yt(t)=round(py(t)); // output in admissible values
```

¹⁶Malý rozptyl šumu + velké hodnoty parametrů = malá neurčitost, a naopak.

```

end

save datT13 yt xt

// RESULTS OF THE SIMULATION
subplot(311), plot(1:nd, yt, 'r. ')
set(gcf(), "position", [700 100 600 500])
set(gca(), 'data_bounds', [0 nd+1 -.1 1.1])
title('Output')
subplot(312), plot(1:nd, xt(2,:), '. ')
title('Continuous input')
subplot(313), plot(1:nd, xt(3,:), '. ')
title('Discrete input')

```

11.2.4 T14simState

Simulace se stavovým modelem

Teorie

Stavový model popisuje veličinu, kterou nemůžeme přímo měřit, a o které se dozvídáme jen prostřednictvím měřeného vstupu a výstupu. Model má dvě rovnice. První popisuje dynamický vývoj stavu, druhá ukazuje, jak se stav projeví ve výstupu

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

Simulace začíná z počáteční hodnoty x_1 . V každém kroku spočte budoucí stav x_{t+1} a také výstup y_t , který spolu s vstupem u_t nese informaci o stavu x_t .

Program

... je velice jednoduchý.

Na konci programu se uloží data `datT14` pro využití v dalších úlohách.

Doporučené pokusy

1. Zkuste zvolit své parametry a počáteční podmínku.
2. Sledujte odezvy modelu bez šumu.
3. Kdy bude systém stabilní?¹⁷

¹⁷Je stabilní, když všechna vlastní čísla matice M , tj. kořeny rovnice $\det(M - \lambda I) = 0$, jsou v absolutní hodnotě menší než 1.

```

// Simulation with a 2-dimensional state-space model
// -----
exec("ScIntro.sce",-1)

// PARAMETERS OF THE SIMULATION
nd=100; // number of data

// model parameterers x(t+1)=Mx(t)+Nu(t), y(t)=Ax(t)+Bu(t)
M=[.6 -.3
   .2 .1];
N=[3 -.1]';
A=[.1 2];
B=1;
Rw=eye(2);
Rv=.1;

xt=zeros(2,nd);
ut=rand(1,nd,'unif');
yt=zeros(1,nd);
x=zeros(2,1); // initial conditions for state

// TIME LOOP OF SIMULATION
for t=2:nd
    x=M*x+N*ut(t)+sqrt(Rw)*rand(2,1,'norm');
    yt(t)=A*x+B*ut(t)+sqrt(Rv)*rand(1,1,'norm');
    xt(:,t)=x;
end

save datT14 xt yt ut M N A B

// RESULTS OF SIMULATION
subplot(211),plot(xt')
set(gcf(),"position",[700 100 600 500])
title('State variables')
subplot(212),plot(1:nd,ut,1:nd,yt)
legend('input','output')

```

11.3 Odhad

11.3.1 T21estCont2

Odhad parametrů regresního modelu 2. řádu

Teorie

Regresní model má rovnici

$$y_t = \psi_t' \theta + e_t,$$

kde $\psi_t = [y_{t-1}, y_{t-2}, u_t, u_{t-1}, u_{t-2}, 1]'$ a $\theta = [a_1, a_2, b_0, b_1, b_2, k]$.

Algoritmus odhadu:

V_0 počáteční statistika,

y_0 počáteční podmínka,

$t = 1$

1. $\Psi = [y_t \psi'_t]'$ rozšířený regresní vektor
2. $V_{t+1} = V_t + \Psi\Psi'$ přepočítání informační matice
3. $t = t + 1$, je-li $t \leq nd$ jdi na 1.

$V \rightarrow V_y, V_{y\psi}, V_\psi. \quad th = inv(V_\psi) V_{y\psi}, \quad r = (V_y - V_{y\psi}th) / nd$

Program

... sleduje uvedený algoritmus.

Počáteční informační matice je $V = \text{zeros}(7) + 1e-8 * \text{eye}(7)$. To je nulová matice s nepatrnou diagonálou, kterou přidáváme proto, aby matice neztratila vlastnost pozitivní definitnosti.

Touto maticí vyjadřujeme úplnou počáteční neznalost. Pokud bychom chtěli tuto zadat s více informací, můžeme postupovat takto. Generujeme počáteční data - tj, pro několik náhodně zvolených regresních vektorů dosadíme do modelu a spočteme příslušné výstupy. Tato data použijeme pro konstrukci počáteční informační matice podle stejného algoritmu, jako při vlastním odhadování. Tuto počáteční simulaci můžeme provést ručně.

Simulovaná data se otevřou z disku příkazem `load datT11`. Tato data byla vytvořena a uložena programem `T11simCont2.m`

Doporučené pokusy

1. Simulujte a použijte k odhadu data s různým množstvím neurčitosti. Všimněte si přesnosti odhadu.
2. Proveďte odhad s různě velkým vybuzením systému (velikost šumu a řídicí veličiny)
3. Odhadujte pro různé typy řízení (skok, sinusovka, skoky, šum). Porovnejte přesnost odhadu.
4. Zkuste odhadovat s různě velkou počáteční informací.¹⁸

```
// Estimation of 2nd order regression model (uses T11)
// -----
exec("ScIntro.sce",-1)
```

```
load datT11 yt ut
nd=length(yt);
```

¹⁸Čím více počátečních dat se použije, tím je informace silnější. Počáteční data se mohou i opakovat.

```

V=zeros(7)+1e-8*eye(7);
th=zeros(6,nd);
r=zeros(1,nd);
for t=3:nd
    psi=[yt(t-1:t-2) ut(t-1:t-2) 1];
    V=V+psi'*psi;
    Vy=V(1,1); Vyp=V(2:$,1); Vp=V(2:$,2:$);
    th(:,t)=inv(Vp+1e-8*eye(Vp))*Vyp;
    r(t)=(Vy-Vyp'*inv(Vp+1e-8*eye(Vp))*Vyp)/t;
end

set(figure(1),'position',[500 60 600 500])
set(gcf(),"background",8)
for i=1:6
    subplot(6,1,i),plot(th(i,:))
    if i==1, title('Regression coefficients'), end
end
set(figure(2),'position',[50 60 400 200])
set(gcf(),"background",8)
plot(r)
title('Noise variance')

```

11.3.2 T22estDisc

Odhad parametrů diskrétního modelu

Teorie

Uvažujeme diskrétní model ve formě tabulky

| u_t, y_{t-1} | $y_t = 1$ | $y_t = 2$ |
|----------------|-----------------|-----------------|
| 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 | $\Theta_{1 22}$ | $\Theta_{2 22}$ |

Jedná se o model typu “řízená koruna s pamětí”. Můžeme si představit, že máme korunu, kterou házíme, a že to, co padne, závisí jednak na tom, jak si minci položíme při házení na dlaň (řízení) a také na tom, co padlo v předchozím pokusu - třeba že na stranu, která minule padla dolů, se něco přilepilo a minci rozvážilo. Pravděpodobnost toho, co padne, tedy je

$$f(y_t | u_t, y_{t-1}).$$

To znamená: jestliže znám $u_t = i$ a $y_{t-1} = j$, bude pravděpodobnost $y_t = 1$ rovna $\Theta_{1|ij}$ a pravděpodobnost $y_t = 2$ bude $\Theta_{2|ij}$ (podle tabulky).

Z toho plyne, že na model lze pohlížet jako na 4 koruny, každá je zadána svým regresním vektorem $[u_t, y_{t-1}]$ a má své parametry - příslušný řádek tabulky. Odtud také plyne, že součet čísel v každém řádku tabulky musí být roven jedné.

Odhad provádíme podle algoritmu:

ν_0 počáteční statistika (matice 4×2) nulová nebo s počátečními hodnotami,

zvolíme počáteční podmínku y_0 ,

$t = 1$

1. máme u_t a y_{t-1}
2. změříme y_t
3. určíme prvek statistiky s indexem $y_t|u_t, y_{t-1}$ a přičteme k němu jedničku
4. $t = t + 1$, jestliže $t \leq nd$, jdi na 1.

Bodové odhady parametru dostaneme tak, že každý řádek tabulky se statistikou dělíme součtem jejích prvků (tabulku normalizujeme).

Program

... je velmi jednoduchý a sleduje uvedený algoritmus. Používá se výpočet pořadí řádku v tabulce podle hodnot regresního vektoru $[u_t, y_{t-1}]$

$$i = 2 * (ut(t) - 1) + yt(t - 1),$$

který přiřazuje $1, 1 \rightarrow 1, 1, 2 \rightarrow 2, 2, 1 \rightarrow 3$ a $2, 2 \rightarrow 4$.

Simulovaná data se otevřou z disku příkazem `load datT12`. Tato data byla vytvořena a uložena programem `T12simDisc.m`

Doporučené pokusy

1. Simulujte a použijte k odhadu data s různým množstvím neurčitosti. Všimněte si přesnosti odhadu.
2. Proveďte odhad s různě velikým vybuzením systému (velikost šumu a řídicí veličiny)
3. Zkuste odhadovat s různě velkou počáteční informací.¹⁹

```
// Estimation of discrete model (uses T12)
// -----
exec("ScIntro.sce",-1)
```

```
load datT12 yt ut
nd=length(yt);
```

```
V=zeros(4,2);
th=zeros(4,nd);
for t=2:nd
```

¹⁹Čím více počátečních dat se použije, tím je informace silnější. Počáteční data se mohou i opakovat.

```

    i=2*(ut(t)-1)+yt(t-1);
    j=yt(t);
    V(i,j)=V(i,j)+1;
    thp=V./(sum(V,2)*ones(1,2));
    th(:,t)=thp(:,1);
end

set(figure(1),'position',[500 60 600 500])
set(gcf(),"background",8)
for i=1:4
    subplot(4,1,i),plot(th(i,:))
    set(gca(),'data_bounds',[0 nd -1 1.1])
    if i==1, title('Model coefficients'), end
end

```

11.3.3 T23LogRegM

Odhad parametrů logistické regrese

Teorie

Odhad logistické regrese je složitější, protože vyžaduje numerickou maximalizaci věrohodnostní funkce. Není až tak velký problém napsat vzorec pro věrohodnostní funkci pro naměřená data (průběžný přepočítání statistik jako v případě regresního a diskrétního modelu tady není možný). Dokonce lze i analyticky spočítat gradient a Hessián věrohodnostní funkce což umožní použít jednoduchou Newtonovu metodu optimalizace. I program pro Newtonovu metodu je jednoduchý, ale dohromady je toho zkrátka dost. Proto použijeme existujících funkcí Matlabu, které úlohu odhadu logistické regrese řeší.

Jsou to funkce:

- `[p,dev,stats] = mnrfit(xt,yt)`; která počítá odhad regresních koeficientů p
 - p je odhad regresních koeficientů
 - xt a yt jsou data ve sloupcích (čas běží po řádcích); xt je bez jedničky na začátku.
 - zbytek: `help mnrfit`
- `py = mnrvl(p,xt)`; která počítá pravděpodobnosti hodnot výstupů při zadaných parametrech p a regresních vektorech xt
 - py jsou pravděpodobnosti hodnot výstupu
 - p jsou odhadnuté parametry
 - xt jsou regresní vektory (bez jedničky na začátku)
 - další: `help mnrvl`

Program

... je v podstatě tvořen voláním zmíněných procedur.

Simulovaná data se otevřou z disku příkazem `load datT13`. Tato data byla vytvořena a uložena programem `T13simLogReg.m`

Doporučené pokusy

1. Odhadujte při různé neurčitosti dat.²⁰ Porovnejte výsledky podle chyby předpovědi.
2. Simulujte čistě diskrétní data a odhadněte je (i) pomocí diskrétního modelu (ii) pomocí logistické regrese. Jaký je v tom významový rozdíl?²¹

```
// Logistic regression and its verification (uses T13)
// -----
exec("ScIntro.sce",-1)

load datT13 yt xt
nd=length(yt); // length of data

// ESTIMATION and PREDICTION
xt=xt(2:3,:); // regression vectors (without 1)
yt=yt'; // output (must start from 1)
[py, yp, yr, b]=lrEst([],yt,xt);

// RESULTS
plot(1:nd,yt,'bo',1:nd,yp,'rx',1:nd,round(yp),'g+')
set(gcf(),"position",[700 100 600 500])
legend('output','prob.predic','rounded pred')
title('Logistic regression')
set(gca(),'data_bounds',[0 nd+1 -1 1.1])
```

11.4 Předpověď

11.4.1 T31preCont2

Předpověď s regresním modelem 2. řádu

Teorie

Úkolem je předpovídat výstup soustavy np kroků dopředu na základě odhadnutého regresního modelu 2. řádu, tj.

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + b_0 u_t + b_1 u_{t-1} + b_2 u_{t-2} + k + e_t$$

²⁰O neurčitosti při simulaci se mluvilo u simulace. Neurčitost odhadu je vyjádřena velikostí odhadnutých regresních koeficientů. Čím jsou hodnoty koeficientů větší, tím méně neurčitosti bylo v datech.

²¹Diskrétní model odhaduje pravděpodobnosti všech stavů systému (stavů ve smyslu různých konfigurací rozšířeného regresního vektoru). Logistická regrese dělá nejdříve určitý druh klasifikace a pak odhaduje na třídách této klasifikace. Logistická regrese nevede na tak velké dimenze jako diskrétní model. *Pozn. Vše je dosti volně řečeno.*

Konstruujeme bodový odhad výstupu s modelem s dosazenými bodovými odhady parametrů. Prakticky to znamená, že dosazujeme do modelu s odhadnutými hodnotami parametrů a bez šumu a počítáme výstup. Pokud máme data (na začátku předpovědi) dosazujeme tato data. Kde (později) data chybí, dosazujeme bodové předpovědi z minulých kroků.

Program

... je trochu složitější, protože spojuje tři úlohy - simulaci, odhad a předpověď.

Simulace a akumulace informace pro odhad se provádí v prvním cyklu. Po jeho skončení se počítají bodové odhady parametrů.

Další cyklus počítá np -krokovou předpověď výstupu. Vnější cyklus běží v čase. Pro každé t se

- sestaví vektor dat y_i , která jsou v tomto čase k dispozici (tedy do času $t - 1$),
- na základě těchto dat se provede předpověď \hat{y}_t a zařadí se do vektoru y_i ,
- dále se předpovídá pro časy $i = t + 1, t + 2, \dots, t + mp - 1$; přitom se používají data / předpovědi z vektoru y_i .

Na závěr programu se srovnávají předpovědi se skutečnými daty.

Doporučené pokusy

1. Měňte neurčitost simulovaných dat a porovnejte přesnost předpovědi.
2. Srovnajte výsledky předpovědi na 1, 3, 5 a 10 kroků.
3. Ověření předpovědi na stejných datech, která byla použita pro odhad není úplně fér. Upravte program tak, aby se odhadovalo a předpovídalo na různých množinách dat.
4. Zkuste upravit program tak, aby se průběžně odhadovalo i předpovídalo. Ve stávající verzi programu se nejdříve odhaduje a konečné výsledky odhadu se použijí pro předpověď.

```
// Prediction with 2nd order regression model
// -----
exec("ScIntro.sce",-1)
function z=psi(t,y,u), z=[y(t-1:-1:t-2) u(t-1:t-2) 1]; endfunction
// zajĀmavĀŠ kt = sin(0:pi/2) a sin(0:pi/4)

nd=200; // number of data (both estim. and predic.)
np=3; // number of steps pfor prediction

a=[.4 .2]; // parameters at yt
b=.1*[1 .2 -.5]; // parameters at ut
k=0; // constant (model absolute term)
select 2 // set 0 for kt not used
case 1 then
    kt=2*sin(%pi*(1:nd)/nd); // time varying model constant :D
case 2 then
```

```

    kt=sign(100*sin(5*%pi*(1:nd)/nd));
else kt=zeros(1,nd); end
s=.01;                                // noise variance
ths=[a b k]';                          // parameters in one vector
yt=zeros(1,nd);
ut=rand(1,nd,'norm');

// SIMULATION + ESTIMATION
V=1e-8*eye(7);                          // initial statistics
for t=3:nd,
    yt(t)=psi(t,yt,ut)*ths+kt(t)+s*rand(1,1,'norm'); // simulation
    Psi=[yt(t) psi(t,yt,ut)]';          // extended regression vector
    V=V+Psi*Psi';                       // update of statistics
end
Vy=V(1,1); Vyp=V(2:$,1); Vp=V(2:$,2:$); // partitioning of statistics
th=inv(Vp)*Vyp;                          // point estimates of reg. coef.
r=(Vy-Vyp'*inv(Vp)*Vyp)/nd;             // noise variance
printf('\nEstimated parameters\n')
printf('(variance nad reg. coefficients)\n')
disp(th,r)

// MULTI-STEP PREDICTION
for t=3:(nd-np+1)
    yi=yt(1:t-1);                        // older data (at time t)
    for j=0:(np-1)
        yi(t+j)=psi(t+j,yi,ut)*th;      // prediction on interval 1:np
    end
    yp(t+np-1)=yi(t+np-1);              // works on predictions (not data)
end

// RESULTS
plot(1:nd,yt,1:nd,yp,'r:','markersize',5)
set(gcf(),"position",[700 100 600 500])
legend('output','prediction')

// Remark 1
// It is possible to transfer this m-file to
// current estimation. It is necessary to store current
// estimates and to use them later.

// Remark 2
// The troubles with prediction can be demonstrated
// using intensities with daily courses as predicted data

```

11.4.2 T32preDisc

Předpověď s diskretním modelem

Teorie

Budeme předpovídat výstup soustavy np kroků dopředu na základě odhadnutého diskretního modelu ve formě tabulky

| y_{t-1} | $y_t = 1$ | $y_t = 2$ |
|-----------|----------------------|----------------------|
| 1 | $\hat{\Theta}_{1 1}$ | $\hat{\Theta}_{2 1}$ |
| 2 | $\hat{\Theta}_{1 2}$ | $\hat{\Theta}_{2 2}$ |

kde za neznámý parametr Θ dosadíme jeho bodový odhad $\hat{\Theta}$, který získáme předem odhadem z dat.

Budeme uvažovat neřízený model (bez u_t) a prvního řádu. V tomto případě lze použít běžnou maticovou symboliku a celý algoritmus více krokové předpovědi je velice jednoduchý. np -krokovou předpověď dostaneme tak, že počáteční podmínku (y_{t-1}) násobíme np -tou mocninou matice, která se rovná transponované tabulce představující parametr modelu.

Program

... opět obsahuje několik úloh:

První cyklus generuje data a ihned také přepočítává odhadovou statistiku. Po ukončení cyklu se ze statistiky počítají bodové odhady parametru. Pro simulace se používá předpis

$$yt(t)=(rand<th(yt(t-1),1))+1;$$

ve kterém se podle hodnoty minulého výstupu y_{t-1} vybere řádek tabulky parametru a příkazem $rand < p$ se vyrobí 1 pro $rand < p$ a 0 pro $rand \geq p$. Protože interval $(0, p)$ má délku p , bude $P(0) = p$ a $P(1) = 1 - p$. Přičtená jednička převede $0,1 \rightarrow 1,2$. Přepočet statistiky se provádí tak, že k prvku statistiky s indexem $y_t|y_{t-1}$ se přičte jednička.

Druhý cyklus provádí v každém čase t předpověď np kroků dopředu. Počáteční podmínka je $f(y_{t-1})$ pravděpodobnostní funkce pro y_{t-1} . Pro $y_{t-1} = 1$ je $f(y_{t-2}) = [1, 0]^t$ (tedy $P(1) = 1$ a $P(2) = 0$), pro $y_{t-1} = 2$ je $f(y_{t-2}) = [0, 1]^t$. Pato počáteční podmínka se násobí maticí Θ^{np} a dostaneme pravděpodobnostní funkci $f(y_{t+np-1}|y_{t-1})$. Bodový odhad určíme jako tu hodnotu výstupu, která má větší pravděpodobnost.

Na závěr programu se srovnávají předpovědi se skutečnými daty.

Doporučené pokusy

1. Měňte neurčitost simulovaných dat a porovnejte přesnost předpovědi.
2. Srovnejte výsledky předpovědi na 1, 3, 5 a 10 kroků.
3. Upravte program tak, aby se odhadovalo a předpovídalo na různých množinách dat.

4. Zkuste upravit program tak, aby se průběžně odhadovalo i předpovídalo.

```
// Prediction with discrete model
exec("ScIntro.sce",-1)

// TASK PARAMETERS
nd=100; // number of data
np=1; // length of prediction
// np=1 means prediction t-1 -> t

// model parameter
//yt(t)=1 =2 yt(t-1)
th= [.2 .8 // 1
     .6 .4]; // 2

// SIMULATION + ESTIMATION
yt=zeros(1,nd);
yt(1)=2; // initial output
V=1e-8*eye(2,2); // initial statistics
// loop for simulation
for t=2:nd
    yt(t)=(rand(1,1,'unif')<th(yt(t-1),1))+1; // output generation
    V(yt(t-1),yt(t))=V(yt(t-1),yt(t))+1; // statistics recomputation
end
th=V./(sum(V,2)*ones(1,2)); // point estimates of model parameter

// PREDICTION
yp=ones(1,nd);
for t=2:nd-np
    if yt(t-1)==1, y0=[1 0]'; // prob. of output at time t
    else y0=[0 1]'; end
    py=th^np*y0; // prob. of predicted output
    [xxx yp(t+np)]=max(py,'r'); // point estimate of output
end

// RESULTS
plot(1:nd,yt,'x',1:nd,yp,'ro')
set(gcf(),"position",[700 100 600 500])
title('Prediction with discrete model')
legend('output','prediction')
set(gca(),'data_bounds',[1 nd .9 2.1])
```

11.4.3 T34statEstKF

Odhad stavu (Kalmanův filtr)

Teorie

Odhad stavu je úloha, při které odhadujeme neměřenou veličinu x_t z měřených dat - vstupu u_t a výstupu y_t . Lineární stavový model je

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

kde stav x_t je sloupcový vektor, y_t , u_t uvažujeme skalární, M , N , A , B jsou matice známých parametrů odpovídajících rozměrů. Šumy w_t a v_t uvažujeme jako normální s nulovou střední hodnotou a kovarianční maticí (rozptylem) R_w a R_v . Odhad se provádí jako rekurze pro střední hodnoty a kovarianční matice rozdělení.

Odhad stavu probíhá ve dvou stupních:

- změří se nová data a provede se přepočítání minulého odhadu stavu podle chyby predikce, tj. podle rozdílu mezi předpovězeným výstupem z modelu a výstupem naměřeným - filtrace
- filtrovaný odhad stavu se předpovídá do budoucího času (už bez nově měřené informace) podle modelu - predikce.

Tak dochází k přepočtům

$$\hat{x}_{t|t-1} \rightarrow \hat{x}_{t|t} \rightarrow \hat{x}_{t+1|t}$$

kde stříška značí odhad, první index označuje čas kam odhad patří, druhý index označuje čas, do kterého byla zahrnuta měřená data.

Program

... obsahuje vlastně jen volání procedury `Kalman`, která provádí vlastní rekurzivní odhad stavu

- `[x,Rx,yp(t)]=Kalman(x,yt(t),ut(t),M,N,A,B,Rw,Rv,Rx)`; popis procedury `Kalman` je ve vlastním programu.
 - x je stav (je to vstup - výstupní veličina, dochází k přepočtu)
 - Rx je kovarianční matice odhadu stavu (rovněž se přepočítá)
 - yp je predikce výstupu
 - yt , ut jsou data
 - M , N , A , B jsou matice stavového modelu
 - Rw je kovarianční matice šumu w_t
 - Rv je rozptyl šumu v_t

Kovarianční matice je třeba na začátku odhadu zadat.

Matici Rx zadáme jako diagonální, s dosti velkými čísly na diagonále. Tato matice vyjadřuje naši nedůvěru k počátečnímu odhadu stavu. Čím jsou čísla na diagonále větší, tím je větší naše nedůvěra.

Matice Rw a číslo Rv zadáváme podle apriorní znalosti dat. Z modelu je

$$w_t = x_{t+1} - Mx_t - Nu_t \text{ a } v_t = y_t - Ax_t - Bu_t.$$

Odtud se snažíme získat odhady Rw a Rv . Na jejich dobrém odhadu dosti závisí celý proces odhadu stavu. Toto místo je často pro odhad kritické.

Na závěr programu se srovnává odhad stavu se stavem simulovaným.

Doporučené pokusy

1. Testujte závislost průběžného odhadu stavu na volbě jeho počáteční hodnoty.
2. Testujte kvalitu odhadu stavu v závislosti na volbě Rx .
3. Testujte kvalitu odhadu stavu v závislosti na volbě Rw a Rv .

```
// State estimation (Kalman filter) – uses T14
// _____
exec("ScIntro.sce",-1)

load datT14 xt yt ut M N A B
nd=length(yt);           // number of data

// TASK PARAMETERS
Rw=1*eye(2,2);           // state noise covariance
Rv=.1;                   // output noise covariance
Rx=1000*eye(2,2);       // estimated state covariance

// ESTIMATION
yp=zeros(1,nd);
xp=zeros(2,nd);
x=zeros(2,1);           // initial state
// loop for state estimation
for t=2:nd
    [x,xf,Rx,yp(t)]=Kalman(x,yt(t),ut(t),M,N,A,B,Rw,Rv,Rx);
    xp(:,t)=xf;
end

// RESULTS
subplot(311),plot(1:nd,xt(1,:),1:nd,xp(1,:))
set(gcf(),"position",[700 100 600 500])
title('First state entry')
legend('state','estimate')
subplot(312),plot(1:nd,xt(2,:),1:nd,xp(2,:))
```

```

title('Second state entry')
legend('state','estimate')
subplot(313), plot(1:nd, yt(1,:), 1:nd, yp(1,:))
title('Output')
legend('output','estimate')

```

11.5 Klasifikace

11.5.1 T41classMix

Odhad modelu směsi komponent, klasifikace

Teorie

Předpokládáme, že data jsou generována z konečného počtu různých zdrojů, z nichž každý má svůj samostatný model - komponentu. Výsledný model (model směsi komponent) je konvexní kombinací těchto komponent²². Při odhadu směšového modelu se určují parametry komponent a jejich váhy. Pro odhad parametrů komponent je třeba průběžně odhadovat tzv. ukazovátka. To je náhodný proces, který v každém čase ukazuje na komponentu, která je právě aktivní, tj. k níž patří generovaná data. Toto ukazovátka třídí přicházející data do jednotlivých komponent, tj. provádí klasifikaci dat. Predikce ze směšového modelu a klasifikace dat jsou dvě nejdůležitější úlohy, které lze s modelem směsi realizovat.

Odhad modelu směsi komponent je po teoretické stránce dosti složitá úloha, která vyžaduje aproximaci. Tento model nedává rekurzivní odhad s reprodukcujícím se Bayesovým vzorcem. Abychom tuto úlohu pokud možno zjednodušili, přijmeme následující předpoklady:

1. Uvažujeme aproximaci, která vůbec dovolí úlohu realizovat. Ta spočívá v tom, že ukazovátka ve tvaru Kroneckerovy delta funkce ($\delta(c, c_t) = 1$ pro $c = c_t$ a jinak je nula - tedy jedničkou ukazuje na právě aktivní komponentu c_t) nahradíme jeho střední hodnotou, tj.

$$\begin{aligned} \delta(c, c_t) &\rightarrow E[\delta(c, c_t) | d(t)] = \sum_{c=1}^{n_c} \delta(c, c_t) f(c_t | d(t)) = \\ &= [P(c_t = 1 | d(t)), P(c_t = 2 | d(t)), \dots, P(c_t = n_c | d(t))]'. \end{aligned}$$

Tedy připouštíme, že neexistuje jen jedna aktivní komponenta, ale aktivitu přebírají všechny komponenty, každá se svou aktuální pravděpodobností $P(c_t = c | d(t))$, $c = 1, 2, \dots, n_c$.

2. Komponenty budeme předpokládat statické - tj., v datovém prostoru mají pevné středy (centra) a data se kumulují (klastují) kolem nich. V dvourozměrném příkladu je datový prostor rovina a klastry jsou shluky dat kolem pevných bodů.
3. Pro výpočet datové předpovědi, podle které se určují pravděpodobnosti aktivity jednotlivých komponent, použijeme jako odhad parametrů komponenty místo aposteriorní hp (jak nakazuje bayesovství) jen jejich bodové odhady. Pravděpodobnosti aktivity komponent pak získáme tak, že změřený datový vzorek dosadíme do všech komponent s dosazenými bodovými odhady parametrů a získaný vektor hodnot znormalizujeme na součet jedna. Tento postup je názorný, jednoduchý a chyba, které se dopustíme tím, že nepoužijeme celou aposteriorní hp není velká.

²²Konvexní kombinace má za váhy pravděpodobnosti. Ty jsou nezáporné a v součtu dají jedničku.

Postup odhadu je pak následující:

Určíme počet komponent a jejich počáteční umístění (střední hodnoty komponent) a šířku (kovarianční matice komponent). Určíme počáteční odhad pravděpodobností se kterými jsou komponenty aktivovány - to je α .

1. Změříme nová data.
2. Určíme datové předpovědi pro všechny komponenty.
3. Určíme pravděpodobnosti komponent α (normalizace statistiky)
4. Z datových predikcí a pravděpodobností komponent určíme aktuální váhy komponent w_t .
5. Aktualizujeme statistiky parametrů (data vážíme pomocí w_t)
6. Aktualizujeme statistiku pro odhad α (opět s vahami w_t)
7. Konec? ne - jdeme na 1.

Program

... má několik částí.

Nejdříve se simulují data. Podle zadaných pravděpodobností se vybere aktivní komponenta a z ní se generují data.

V další části programu se provádí odhad, kombinovaný s klasifikací tak, jak jsme uvedli v minulém odstavci. Nejprve probíhá inicializace odhadu (počáteční statistiky). Volbou parametru `ini` lze vybrat jednu z předvolených variant. První je trochu nefér, protože staví na skutečných centrech komponent, druhá se opírá jen o počáteční data. Význam dobrého počátečního odhadu center je velmi důležitý.

Dále se realizuje výše popsany algoritmus. Funkce `Gauss2` počítá hodnotu gaussovského dvou-rozměrného modelu po dosazení dat. Vzhledem k numerice, funkce počítá logaritmus hodnot jednotlivých komponent. Vektor těchto hodnot se pak předběžně normuje aby maximální hodnota byla nula. Pak se prvky vektoru "exponencializují" a znovu se provádí normalizace na jednotkový součet. Předběžná normalizace převede hodnoty na "rozumná" čísla, ze kterých je možno počítat exponenciálu.

Funkce `v2th2` počítá bodové odhady parametrů komponent.

Na konci se prezentují výsledky:

- Graf 1 ukazuje vývoj odhadu regresních koeficientů - kroužky jsou skutečné polohy center, body vyznačují postupné upřesňování odhadů.
- Graf 2 zobrazuje
 - průběh vah w_t během odhadu - pravděpodobnosti přeskakují z jedné komponenty na druhou,
 - aktuální pravděpodobnosti aktivity komponent (ze současných dat),
 - pravděpodobnosti aktivity jednotlivých komponent (z minulých dat),
- Graf 3 ukazuje simulované datové klastry, použité pro odhad.

Doporučené pokusy

1. Testujte schopnosti klasifikace pro různé datové vzorky (různé tvary klastrů a jejich vzájemné vzdálenosti).
2. Testujte, jak daleko od skutečných center komponent mohou ležet počáteční centra tak, aby odhad ještě proběhl úspěšně.
3. Simulujte negaussovské klastry a zhodnoťte výsledky klasifikace.
4. Pokuste se navrhnout nový, lepší způsob inicializace odhadového algoritmu.

```
// Mixture model estimation and classification
// -----
exec(" ScIntro.sce",-1)

nd=200; // number of data for estimation
ni=10; // number of data for initialization
ini=1; // type of initialization (see row 32)
sca=3; // scattering of the data centre

sd=.4;
fi=.995; // forgetting (of older data in estimation)
thS(1).m=[-1 -1];
thS(2).m=[2 -2];
thS(3).m=[1 3]; // simulated centres
thS(1).s=sd*[1 1;0 1];
thS(2).s=sd*[2 -1;0 1];
thS(3).s=sd*[1 1;0 3]; // simulated st.deviation
alS=[.3 .5 .2]; // simulated comp. weights

// SIMULATION
c=zeros(1,nd);
yt=zeros(2,nd);
for t=1:(nd+ni)
    c(t)=sum(rand(1,1,'unif')>cumsum(alS))+1; // selection of component
    yt(:,t)=thS(c(t)).m'+thS(c(t)).s*rand(2,1,'norm'); // output generation
end
yi=yt(:,(nd+1):nd+ni); // data for initialization
mi=mean(yi,'c'); // center of initial data

// ESTIATION
// initialization
fc=zeros(3,1);
Lp=zeros(3,1);
ga=ones(3,1);
st(1).V=1e-8*eye(3);
st(2).V=1e-8*eye(3);
st(3).V=1e-8*eye(3); // initial component statistics
for t=1:ni // initial centres and corresponding statistics
    for i=1:3;
```

```

    select ini
        case 1, cent=thS(i).m'; // real component centres
        case 2, cent=mi; // centre of initial data
    end
    d=[cent;1]+[sca*rand(2,1,'norm');0]; // initial reg. vetor
    st(i).V=fi*st(i).V+d*d'; // accumulation of initial statistics
end
end
for i=1:3
    [tp cp]=v2th2(st(i).V,ni); // initial stat. -> initial parameters
    thE(i).m=tp'; // initial regression coefficients
    thE(i).cv=cp/100; // initial covariances (better slim)
end
// approximative estimation
for t=1:nd
    for i=1:3
        [xxx Lp(i,1)]=Gauss2(yt(:,t),thE(i).m,thE(i).cv); // weights from
        // components (from numerical reasons, logarithm are computed)
    end
    Lpn=Lp-max(Lp); // normalization to reasonable values
    p=exp(Lpn); // taking exponentials from logarithms
    if sum(abs(p))<1e-8
        p=ones(3,1); // if all p are zero, take uniform
    end
    wp=p.*ga; // weight w is proportional to f(y|c)*f(c)
    w=wp/sum(wp); // normalization to sum=1
    // update of statistics for the pointer
    ga=ga+w;
    // update of statistics for the components
    for i=1:3
        d=[yt(:,t);1];
        st(i).V=st(i).V+w(i)*d*d'; // update
        [thE(i).m thE(i).cv]=v2th2(st(i).V,t); // point estimates
    end
    // storing current results
    fmt(:,t)=p/sum(p); // f(y|c)
    fct(:,t)=ga/sum(ga,'r'); // f(c)
    wt(:,t)=w; // weight w
    fyt(t)=[1 2 3]*w; // output prediction
    th1(:,t)=thE(1).m'; // 1st component centre
    th2(:,t)=thE(2).m'; // 2nd component centre
    th3(:,t)=thE(3).m'; // 3rd component centre
end

// evolution of the component centres
set(figure(1),'position',[600 60 600 600])
set(gcf(),"background",8)
plot(th1(1,:),th1(2,:),'r.:')
plot(th2(1,:),th2(2,:),'b.:')

```

```

plot(th3(1,:),th3(2,:), 'g.:')
plot(thE(1).m(1),thE(1).m(2), 'ro', 'markersize',12,'linewidth',3)
plot(thE(2).m(1),thE(2).m(2), 'bo', 'markersize',12,'linewidth',3)
plot(thE(3).m(1),thE(3).m(2), 'go', 'markersize',12,'linewidth',3)
title('Evolution of component centres')
set(gca(),'data_bounds',[-2 4 -4 4])

// probabilities in time
if 0 // zatím blbne na subplot(312)
set(figure(2),'position',[80 30 500 600])
set(gcf(),"background",8)
s=1:nd;
subplot(311)
plot(s,wt(:,s),'.:')
title('Weights wt=f(mt)*f(ct)')
set(gca(),'data_bounds',[min(s)-1 max(s)+1 -1 1.1])
subplot(312)
plot(s,fyt(s),'.')
title('Probability f(yt)')
set(gca(),'data_bounds',[min(s)-1 max(s)+1 1.1 3.1])
subplot(313)
plot(s,fct(:,s))
title('Probability f(alpha)')
set(gca(),'data_bounds',[min(s)-1 max(s)+1 -1 1.1])
legend('C1','C2','C3');
end

// data clusters
set(figure(3),'position',[1030 350 300 300])
set(gcf(),"background",8)
plot(yt(1,1:nd),yt(2,1:nd),'.')
title('Data')

```

11.6 Řízení

11.6.1 T51ctrlCont

Řízení se spojitým modelem

Teorie

Dynamické programování je poměrně složitá záležitost. Proto budeme uvažovat jen model prvního řádu a nejjednodušší kvadratické kritérium s penalizační funkcí $y_t^2 + \lambda u_t^2$. Jedná se v podstatě o postupnou minimalizaci kritéria na celém intervalu řízení. Abychom dostali kauzální regulátor, musíme minimalizaci provádět od konce intervalu proti směru času. V každém kroku provedeme nejdříve středování přes neznámý výstup a výsledek, který je funkcí u_t a minulých dat, pak minimalizujeme podle u_t . Tím dostaneme předpis pro optimální řízení. Jsme ale někde uvnitř intervalu řízení a tak nemáme k dispozici veličiny, které se mají do řídicího předpisu

dosadit. Musím proto jen schovávat předpisy řízení, až dojdeme na začátek intervalu. Tady už data máme a můžeme začít vlastní řízení (aplikaci řídicího zákona). Jsme v čase t . Do řízení dosadíme data($t - 1$) a dostaneme u_t . To aplikujeme a změříme y_t . Tím máme data(t) spočteme u_{t+1} aplikujeme a změříme y_{t+1} . A tak dále až na konec horizontu.

Program

... nejdříve provede simulaci s nulovým řízením. Tím dostaneme neřízený výstup pro porovnání s řízeným.

Dále se provádí výpočet řídicího zákona, který je zde reprezentován koeficientem S_t . Výpočet běží proti směru času.

Nakonec se provádí stejná simulace, jako na začátku, ale s optimálním řízením, podle spočteného řídicího předpisu. Řízení má tvar

$$u_t = -S_t y_{t-1}$$

Jeho tvar je dán tvarem použitého modelu (tedy modelu 1. řádu $y_t = a_1 y_{t-1} + b_0 u_t + e_t$).

Podle použitého kritéria řídíme výstup na nulu. Přitom nechceme příliš velké řídicí zásahy, proto je trochu penalizován i vstup. Penalizace vstupu by ale neměla příliš kazit vlastní řízení.

Výsledný graf ukazuje neřízený výstup, k němu pro srovnání řízený výstup a také řízení, které bylo použito.

Doporučené pokusy

1. Zkuste řízení s různou velikostí penalizace řídicí veličiny. Určete nalézt nejlepší hodnotu penalizace, která dá rozumnou řídicí veličinu a přitom příliš nepokazí vlastní řízení.
2. Zkuste měnit simulovanou soustavu a sledujte výsledky řízení.
3. Zkuste řídit nestabilní soustavu - např. s koeficientem $a_1 = 1.1$.

```
// Dynamic programming for the first order rgression model
// -----
exec("ScIntro.sce",-1)

a1=.8; b0=1; // regression coefficients
se=.01; // model noise standard deviation
y0=5; // initial condition for output
la=.1; // penalization of control variable
nd=30; // length of control horizon

// SIMULATION
y=zeros(1,nd); y(1)=y0;
for t=2:nd
    u(t)=0; // no control
    y(t)=a1*y(t-1)+b0*u(t)+se*rand(1,1,'norm'); // without control
end
```

```

// CONTROL OPTIMIZATION
R=0; // initial condition for Bellman function
for t=nd:-1:2 // runs against the time
    L=1a/(1+R);
    S(t).ent=a1*b0/(b0^2+L); // coefficient at u
    R=L*a1^2/(b0^2+L); // partial reminder
end

// CONTROL APPLICATION
yr=zeros(1,nd); yr(1)=y0; // initial condition for output
ur=zeros(1,nd);
for t=2:nd
    ur(t)=-S(t).ent*yr(t-1); // optimal control
    yr(t)=a1*yr(t-1)+b0*ur(t)+se*rand(1,1,'norm'); // with opt. control
end

// RESULTS
plot(1:nd,y,'b.:',1:nd,yr,'r.:',1:nd,ur,'g.:')
set(gcf(),'position',[700 100 600 500])
title('Control to zero - dynamic programming, 1st order model')
legend('y - without control','yr - with control','ur - control')

```

11.6.2 T52ctrlDisc

Řízení s diskretním modelem

Teorie

Pro konstrukci řízení s diskretním modelem platí obecně stejná Bellmanova rovnice jako pro spojitý případ. Je to

$$\varphi_t = E[\omega + \varphi_{t+1}^* | u_t, d(t-1)] \quad (11.1)$$

$$\varphi_t^* = \min_{u_t} \varphi_t, \quad (11.2)$$

Počítá se od konce intervalu řízení, tedy od času $t = N$, s počáteční podmínkou $\varphi_{N+1}^* = 0$. V diskretním případě se ale počítá s tabulkami

| $[u_t, y_{t-1}]$ | $y_t = 1$ | $y_t = 2$ |
|------------------|-----------|-----------|
| 1, 1 | x | x |
| 1, 2 | x | x |
| 2, 1 | x | x |
| 2, 2 | x | x |

kde místo x stojí: penalizace, parametry modelu nebo prvky Bellmanovy funkce.

Penalizace: tvoří tabulku, jejíž prvky penalizují jednotlivé stavy $y_t | u_t, y_{t-1}$.

Model: je zadán tabulkou parametrů.

Bellmanova funkce φ : závisí jen na y_{t-1} - bude to řádkový vektor s dvěma prvky.

Součet ve střední hodnotě (11.1): matice penalizací + matice 4x2 jejíž řádky tvoří φ_{t+1}^* , které závisí jen na y_t (rozmyslet).

Program

... začíná zadáním tabulky parametrů modelu a obdobné tabulky s penalizacemi.

V prvním cyklu programu se provádí optimalizace. První krok je středování v (11.1). Matici penalizací násobíme tabulkou modelu (prvek po prvku) a ve výsledku sečteme sloupce. Dostaneme jeden sloupec, který máme minimalizovat přes u_t . Struktura sloupce je

| $[u_t, y_{t-1}]$ | $E[\cdot \cdot]$ |
|------------------|------------------|
| 1, 1 | a |
| 1, 2 | b |
| 2, 1 | c |
| 2, 2 | d |

Jestliže hodnota minulého výstupu je $y_{t-1} = 1$ pak volíme mezi prvky a a c . Prvek a zvolíme výběrem $u_t = 1$, prvek c volbou $u_t = 2$. Pro případ, že $y_t = 2$ rozhodujeme podobně mezi prvky b a d . Volíme podle toho, který z dvojice prvků a, c nebo b, d je menší. Protože ale v okamžiku této optimalizace nevíme, jaké bude y_{t-1} , musíme si jen zapamatovat obě volby - to je ten řídicí zákon, podle kterého budeme později volit řízení. Tento řídicí zákon schováváme v proměnné us .

V dalším cyklu se pak již jen dosazuje do schovaného řídicího zákona us a vypočtené řízení se aplikuje.

Jako výsledek se zobrazují hodnoty vstupu a výstupu.

Doporučené pokusy

1. Zkuste volit různé modely pro simulaci. Jestliže jsou prvky blízké 0 a 1, je model skoro deterministický (měl by dobře poslouchat řízení). Hodnoty blízké 0.5 definují model s velkým obsahem neurčitosti.
2. Volte různá kritéria řízení (rozhodněte se, jaké cíle chcete sledovat, realizujte je pomocí penalizací a pak sledujte, jak se cíle plní)²³. Cíle kombinujte s různým zadáním počáteční hodnoty výstupu.

```
// Control with discrete dynamic model
// -----
// fp = omega + fs (penalty function, old criterion)
// f = E[fp|d(t)] - expectation over y(t+1)
// fs = min f => u(t+1) - optimal u
exec("ScIntro.sce",-1)

// VARIABLES TO BE SET
nh=30; // length of control interval
y0=1; // initial condition for output

// y 1 2      u y1 = criterion
om=[1 2      // 1 1
```

²³Cíle mohou být např.: ať je výstup stejný jako vstup nebo ať se výstup pokud možno nemění.

```

    2 3      // 1 2
    2 3      // 2 1
    3 4];    // 2 2

// y 1 2      u y1 = system model
th=[.3 .7    // 1 1
    .1 .9    // 1 2
    .4 .6    // 2 1
    .2 .8];  // 2 2

// computed variables and initializations
// us=zeros(nh,2);
fs=zeros(1,2);

// CONTROL LAW COMPUTATION
for t=nh:-1:1
    fp=om+ones(4,1)*fs;          // penalty + reminder from last step
    // expectation
    f=sum((fp.*th),'c');         // expectation over y
    // minimization
    if f(1)<f(3),                 // for y(t-1)=1
        us(t,1)=1; fs(1)=f(1);  // optimal control, minimum of criterion
    else
        us(t,1)=2; fs(1)=f(3);  // optimal control, minimum of criterion
    end
    if f(2)<f(4),                 // for y(t-1)=2
        us(t,2)=1; fs(2)=f(2);  // optimal control, minimum of criterion
    else
        us(t,2)=2; fs(2)=f(4);  // optimal control, minimum of criterion
    end
end
J=fs(y0);                        // final value of criterion

// CONTROL APPLICATION
y(1)=y0;
for t=1:nh
    u(t+1)=us(t,y(t));           // optimal control
    y(t+1)=dsim(u(t+1),y(t),th); // simulation
end

// RESULTS
plot(1:nh+1,y,'ro',1:nh+1,u,'g+')
set(gcf(),'position',[700 100 600 500])
legend('output','input')
set(gca(),'data_bounds',[.8 nh+.2, .8 2.2])
title('Optimal control with discrete model')

printf('\n Minimal value of the criterion: %g\n',J)

```

11.7 Složitější úlohy

11.7.1 T81adaptC1

Adaptivní řízení s regresním modelem 1. řádu

- **metoda ustupujícího horizontu:** V každém kroku t je provedena syntéza řízení na celém horizontu $t + nh, t + nh - 1, \dots, t + 1$ s parametry, odhadnutými k časovému okamžiku t . Z navrženého řízení se realizuje jen jeden krok pro čas t . Nová data se přidávají do datového vzorku a opakuje se odhad parametrů.
- **bodové odhady parametrů:** Optimálním výsledkem odhadu je aposteriorní hp. Ta by měla být použita v syntéze řízení. V tomto příkladě se, pro jednoduchost, počítají bodové odhady parametrů (tj. jejich střední hodnota) a ty jsou dále použity pro syntézu na celém intervalu řízení.

Teorie

Pro řízení je použito dynamické programování. Při něm se počítá optimální řízení s kvadratickým kriteriem na celém intervalu budoucích vstupů a výstupů. Optimalizace začíná na konci intervalu a postupuje dopředu s tím, že v každém kroku se počítá střední hodnota příslušné části kriteriá a její minimalizací se získá předpis pro řízení (výpočet optimálního vstupu). Protože jsme v budoucích časových okamžicích, nelze řízení zatím realizovat, protože budoucí data jejichž je řízení funkcí zatím neznáme. Jak dojdeme do času $t + 1$ dostaneme předpis pro u_{t+1} , které je funkcí dat v čase t a případně starších. Ty už známe a řízení můžeme realizovat. Tím získáme data v čase $t + 1$ a můžeme realizovat další řízení. A tak dále.

Metoda ustupujícího horizontu kombinuje řízení s odhadem. Z napočítaného řízení na časovém intervalu realizuje jen jeden, aktuální krok, a po změřených nových datech aktualizuje odhady parametrů.

Pro simulaci i jako model soustavy je zvolen regresní model 1. řádu bez konstanty, tj.

$$y_t = a_1 y_{t-1} + b_0 u_t + e_t,$$

s normálním šumem.

Program

... má následující strukturu:

- nejprve se zadají parametry úlohy a provede se inicializace veličin,
- potom následuje smyčka diskretního času ve které se periodicky opakují následující úlohy
 - simulace, kdy se aplikuje řízení a měří se hodnoty nového výstupu,
 - odhad parametrů, kdy se přepočte statistika a počítají se bodové odhady
 - optimalizace řízení na řídicím horizontu, kdy počítá předpis pro řízení na horizontu,
 - aplikace řízení, kdy se použije poslední předpis pro výpočet řídicí veličiny.

Pomocí přepínače `estim` je možno zvolit buď řízení se známými parametry nebo adaptivní řízení, kdy se parametry modelu použité v optimalizaci řízení průběžně odhadují.

Jako výsledek se zobrazují dva grafy. První graf zobrazuje řízený výstup soustavy - řídí se na nulu a druhý graf ukazuje vývoj odhadu parametrů modelu - horní obrázek ukazuje parametr a_1 u výstupní veličiny (rovná čára je jeho skutečná hodnota) a dolní obrázek totéž pro b_0 , koeficient u vstupu.

Doporučené pokusy

1. Pro předvolenou soustavu i pro další soustavy s jinými parametry porovnejte adaptivní řízení a řízení se známými parametry.
2. Porovnejte kvalitu řízení (jak adaptivního tak i s pevnými parametry) pro různě dlouhý horizont řízení.
Poznámka: Vývoj řídicího zákona na horizontu je možno sledovat tak, že zastavíte program (myší klik na čítku před číslem příslušného řádku programu) a zobrazíte proměnnou S . Čísla by v okamžiku použití (tj. na konci) měla být ustálená.
3. Pokuste se upravit program tak, aby používal tzv. iterace rozložené v čase, tj. aby vývoj proměnné S (tj. Bellmanovy funkce) nepočítal vždy znovu z nulové hodnoty, ale z hodnoty, se kterou končil v minulém kroku. Délka horizontu se pak může zvolit mnohem kratší.

```
// Adaptive control (receding horizon with
//      point estimates of parameters)
exec("ScIntro.sce",-1)

nd=20;           // length of the adaptive control
nh=3;           // length of control horizon
estim=1;        // estimation 1=yes, 2=no (true par. are used)

a1=.99; b0=.5; s=.1; // system parameters
la=.1;         // penalization of control
y(1)=5; u(2)=1; // initial conditions for output and input
V=eye(3)*1e-8; // initial statistics
for t=2:nd
    // SIMULATION
    y(t)=a1*y(t-1)+b0*u(t)+s*rand(1,1,'norm');

    // ESTIMATION
    psi=[y(t) y(t-1) u(t)]';
    V=V+psi*psi';
    Vy=V(1,1); Vyp=V(2:$,1); Vp=V(2:$,2:$);
    th(:,t)=inv(Vp)*Vyp;
    r(t)=(Vy-Vyp'*inv(Vp)*Vyp)/t;
    if estim==1
        a1E=th(1,t); b0E=th(2,t);
    else
        a1E=a1; b0E=b0;
    end
end
```

```

// CONTROL OPTIMIZATION
R=0; // initial condition for Bellman function
for i=nh:-1:1 // runs against the time
    L=1a/(1+R);
    R=L*a1E^2/(b0E^2+L); // partial reminder
end
S=a1E*b0E/(b0E^2+L); // coefficient at u
// CONTROL APPLICATION
u(t+1)=-S*y(t); // optimal control
end

// RESULTS
set (figure(1), 'position',[100 100 600 500])
set (gcf(), "background",8)
plot (1:nd,y,1:nd,u(1:nd))
title ('Controlled output and control variable')
legend ('output', 'input')

set (figure(2), 'position',[700 100 600 500])
set (gcf(), "background",8)
subplot (211), plot (1:nd, th(1,:),[1 nd],[a1 a1], 'r')
title ('Coefficient at output')
set (gca(), 'data_bounds',[1 nd -2 2])
legend ('estimated', 'true')
subplot (212), plot (1:nd, th(2,:),[1 nd],[b0 b0], 'r')
title ('Coefficient at input')
set (gca(), 'data_bounds',[1 nd -1 1])
legend ('estimated', 'true')

```

11.7.2 T82adaptC2

Adaptivní řízení s regresním modelem 2. řádu

- **metoda ustupujícího horizontu:** V každém kroku t je provedena syntéza řízení na celém horizontu $t + nh, t + nh - 1, \dots, t + 1$ s parametry, odhadnutými k časovému okamžiku t . Z navrženého řízení se realizuje jen jeden krok pro čas t . Nová data se přidávají do datového vzorku a opakuje se odhad parametrů.
- **bodové odhady parametrů:** Optimálním výsledkem odhadu je aposteriorní hp. Ta by měla být použita v syntéze řízení. V tomto příkladě se, pro jednoduchost, počítají bodové odhady parametrů (tj. jejich střední hodnota) a ty jsou dále použity pro syntézu řízení.
- **řízení na žádanou hodnotu (setpoint):** Řízení minimalizující kritérium $\sum y_t^2$ řídí tak, aby výstup byl pokud možno nulový. Jestliže použijeme kritérium $\sum (y_t - w_t)^2$, řídíme tak, aby $y_t - w_t$ bylo co nejbližší nule, a tedy aby bylo $y_t = w_t$. Průběh w_t se nazývá žádaná hodnota pro výstup (setpoint) a ten budeme uvažovat jako průběžně známý, tedy v každém okamžiku t známe $w(t)$.

Poznámka: V programu bude formálně zadán na celém intervalu řízení, ale my počítáme s tím, jako by dopředu nebyl znám.

- **regresní model jako stavový model:** V rekurzivních výpočtech, jako je např. výpočet Bellmanovy funkce v dynamickém programování, je výhodné pracovat s regresním modelem ve stavovém tvaru. Převod regresního modelu na stavový tvar je uveden v odstavci 2.4.

Teorie

Pro řízení je použito dynamické programování. Při něm se počítá optimální řízení s kvadratickým kriteriem na celém intervalu budoucích vstupů a výstupů. Optimalizace začíná na konci intervalu a postupuje dopředu s tím, že v každém kroku se počítá střední hodnota příslušné části kriteriální funkce a její minimalizací se získá předpis pro řízení (výpočet optimálního vstupu). Protože jsme v budoucích časových okamžicích, nelze řízení zatím realizovat, protože budoucí data jejichž je řízení funkcí zatím neznáme. Jak dojdeme do času $t + 1$ dostaneme předpis pro u_{t+1} , které je funkcí dat v čase t a případně starších. Ty už známe a řízení můžeme realizovat. Tím získáme data v čase $t + 1$ a můžeme realizovat další řízení. A tak dále.

Metoda ustupujícího horizontu kombinuje řízení s odhadem. Z napočítaného řízení na časovém intervalu realizuje jen jeden, aktuální krok, a po změřených nových datech aktualizuje odhady parametrů.

Pro simulaci i jako model soustavy je zvolen regresní model 2. řádu s konstantou, tj.

$$y_t = a_1 y_{t-1} + a_2 x_{t-2} + b_0 u_t + b_1 u_{t-1} + b_2 u_{t-2} + k + e_t,$$

s normálním šumem.

Kriterium je ve tvaru

$$\sum (y_t - w_t)^2 + \lambda u_t^2.$$

Program

... má následující strukturu:

- nejprve se zadají parametry úlohy a inicializace veličin,
- dále se objevuje volba dvou předem připravených setpointů (skoky a sinusový průběh),
- potom následuje smyčka diskretního času ve které se periodicky opakují následující úlohy
 - simulace, kdy se aplikuje řízení a měří se hodnoty nového výstupu,
 - odhad parametrů, kdy se přepočte statistika a počítají se bodové odhady; zde je rovněž uvedena volba zda se dále použijí odhadnuté parametry nebo se parametry simulované soustavy považují za známé,
 - optimalizace řízení na řídicím horizontu, kdy počítá předpis pro řízení na horizontu, se opírá o stavový model, který vznikne převodem z regresního,
 - aplikace řízení, kdy se použije poslední předpis pro výpočet řídicí veličiny.

V úvodu programu se objevují dva přepínače:

- Pomocí přepínače `estim` je možno zvolit buď řízení se známými parametry nebo adaptivní řízení, kdy se parametry modelu použité v optimalizaci řízení průběžně odhadují.
- Přepínač `setp` volí jeden ze zadaných setpointů.

Jako výsledek se zobrazují dva grafy. První graf zobrazuje řízený výstup soustavy - řídí se na nulu a druhý graf ukazuje vývoj odhadu parametrů modelu - horní obrázek ukazuje parametr a_1 u výstupní veličiny (rovná čára je jeho skutečná hodnota) a dolní obrázek totéž pro b_0 , koeficient u vstupu.

Doporučené pokusy

1. Pro předvolenou soustavu i pro další soustavy s jinými parametry porovnejte adaptivní řízení a řízení se známými parametry.
2. Porovnejte kvalitu řízení (jak adaptivního tak i s pevnými parametry) pro různě dlouhý horizont řízení.
Poznámka: Vývoj řídicího zákona na horizontu je možno sledovat tak, že zastavíte program (myši klik na čítku před číslem příslušného řádku programu) a zobrazíte proměnnou S . Číslo by v okamžiku použití (tj. na konci) měla být ustálená. Protože S je buňka, musíte pro její výpis použít příkaz `S{:}`.
3. Pokuste se upravit program tak, aby používal tzv. iterace rozložené v čase, tj. aby vývoj proměnné S (tj. Bellmanovy funkce) nepočítal vždy znovu z nulové hodnoty, ale z hodnoty, se kterou končil v minulém kroku. Délka horizontu se pak může zvolit mnohem kratší.

```
// Adaptive control (receding horizon with point estimates of parameters)
// 2nd order regression model with constant, control to given setpoint w
// - pre-estimation of parameters
exec("ScIntro.sce",-1)

nd=100; // length of the adaptive control
nh=15; // length of control horizon
estim=1; // estimation 1=yes, 2=no (true par. are used)
setp=1; // type of setpoint 1=step, 2=sine
pre=1; // pre-estimation (yes,no)

a1=.3; a2=.2; b0=1; b1=-.3; b2=.1; k=1; s=.1; // system parameters
la=.1; // penalization of control
y(1)=5; y(2)=5; u(2)=1; u(2:3)=rand(2,1,'unif'); // initial conditions for output and in
n2=fix(nd/2); ndh=nd+nh;
select setp // setpoint generation
case 1, w=[zeros(1,n2) ones(1,nd-n2) zeros(1,nh)];
case 2, w=sin(2*pi*(1:ndh)/ndh);
end
Om=zeros(5,5); Om(1,1)=1; Om(2,2)=la;
rf=[a1 a2 b0 b1 b2 k]; // references to sim. parameters
// Pre-estimation
V=eye(7)*1e-8; // initial statistics
if pre==1
```

```

for t=3:10;
    e=s*rand(1,1,'norm');
    u(t)=rand(1,1,'n');
    y(t)=a1*y(t-1)+a2*y(t-2)+b0*u(t)+b1*u(t-1)+b2*u(t-2)+k+e
    psi=[y(t:-1:t-2); u(t:-1:t-2); 1]; // reg. vector
    V=V+psi*psi'; // updt of stat.
end
end
R=0; // initial condition for Bellman function
for t=3:nd
    // SIMULATION
    e=s*rand(1,1,'norm');
    y(t)=a1*y(t-1)+a2*y(t-2)+b0*u(t)+b1*u(t-1)+b2*u(t-2)+k+e

    // ESTIMATION
    psi=[y(t:-1:t-2); u(t:-1:t-2); 1]; // reg. vector
    V=V+psi*psi'; // updt of stat.
    Vy=V(1,1); Vyp=V(2:$,1); Vp=V(2:$,2:$); // part. of stat.
    th(:,t)=inv(Vp+1e-8*eye(Vp))*Vyp; // reg. coeff.
    r(t)=(Vy-Vyp'*inv(Vp+1e-8*eye(Vp))*Vyp)/t; // noise variance
    if estim==1 // adaptive control ?
        a1=th(1,t); a2=th(2,t); b0=th(3,t);
        b1=th(4,t); b2=th(5,t); k=th(6,t);
    end

    // CONTROL OPTIMIZATION
    R=zeros(5); // init. for dyn. programing
    N=[b0 1 0 0 0]';
    M=zeros(5,5); M(1,:)=[a1 b1 a2 b2 k]; M(3,1)=1; M(4,2)=1; M(5,5)=1;
    for i=nh:-1:1
        st=w(t+i-1);
        Om(1,5)=-st; Om(5,1)=-st; Om(5,5)=st**2;
        T=R+Om; // dynamic
        A=N'*T*N; // programming
        B=N'*T*M;
        C=M'*T*M;
        S=inv(A)*B;
        R=C-S'*A*S;
    end
    // CONTROL APPLICATION
    x1=[y(t-1) u(t-1) y(t-2) u(t-2) 1]'; // state
    u(t+1)=-S*x1; // control variable
end

// RESULTS
set(figure(1),'position',[100 100 500 500])
set(gcf(),'background',8)
plot(1:nd,y,1:nd,u(1:nd),1:nd,w(1:nd),'c--')
title('Controlled output and control variable')

```

```

legend('output','input','setpoint')

set(figure(2),'position',[700 100 500 500])
set(gcf(),"background",8)
subplot(421),plot(1:nd,th(1,:),[1 nd],rf(1)*ones(1,2),'c--')
title('Coefficient a1')
set(gca(),'data_bounds',[1 nd -2 2])

subplot(422),plot(1:nd,th(2,:),[1 nd],rf(2)*ones(1,2),'c--')
title('Coefficient a2')
set(gca(),'data_bounds',[1 nd -2 2])

subplot(423),plot(1:nd,th(3,:),[1 nd],rf(3)*ones(1,2),'c--')
title('Coefficient b0')
set(gca(),'data_bounds',[1 nd -2 2])

subplot(424),plot(1:nd,th(4,:),[1 nd],rf(4)*ones(1,2),'c--')
title('Coefficient b1')
set(gca(),'data_bounds',[1 nd -2 2])

subplot(425),plot(1:nd,th(5,:),[1 nd],rf(5)*ones(1,2),'c--')
title('Coefficient b2')
set(gca(),'data_bounds',[1 nd -2 2])

subplot(426),plot(1:nd,th(6,:),[1 nd],rf(6)*ones(1,2),'c--')
title('Coefficient k')
set(gca(),'data_bounds',[1 nd -2 2])

subplot(427),plot(1:nd,r,[1 nd],rf(1)*ones(7,2),'c--')
title('Coefficient r')
set(gca(),'data_bounds',[1 nd -2 2])

// subplot(428),plot(1:nd,th(2,:),[1 nd],[a1 a1],'r')
// title('Coefficient b0')
// axis([1 nd -2 2])

```

11.7.3 T83adaptD

Adaptivní řízení s diskrétním modelem 1. řádu

- **metoda ustupujícího horizontu:** V každém kroku t je provedena syntéza řízení na celém horizontu $t + nh, t + nh - 1, \dots, t + 1$ s parametry, odhadnutými k časovému okamžiku t . Z navrženého řízení se realizuje jen jeden krok pro čas t . Nová data se přidávají do datového vzorku a opakuje se odhad parametrů.
- **bodové odhady parametrů:** Optimálním výsledkem odhadu je aposteriorní hp. Ta by měla být použita v syntéze řízení. V tomto příkladě se, pro jednoduchost, počítají bodové odhady parametrů (tj. jejich střední hodnota) a ty jsou dále použity pro syntézu řízení.

Teorie

Pro řízení je použito dynamické programování. Při něm se počítá optimální řízení s kriteriem daným penalizační tabulkou na celém intervalu budoucích vstupů a výstupů. Optimalizace začíná na konci intervalu a postupuje dopředu s tím, že v každém kroku se počítá střední hodnota příslušné části kriteriá a její minimalizací se získá předpis pro řízení (výpočet optimálního vstupu). Protože jsme v budoucích časových okamžicích, nelze řízení zatím realizovat, protože budoucí data jejichž je řízení funkcí zatím neznáme. Jak dojdeme do času $t + 1$ dostaneme předpis pro u_{t+1} , které je funkcí dat v času t a případně starších. Ty už známe a řízení můžeme realizovat. Tím získáme data v čase $t + 1$ a můžeme realizovat další řízení. A tak dále.

Metoda ustupujícího horizontu kombinuje řízení s odhadem. Z napočítaného řízení na časovém intervalu realizuje jen jeden, aktuální krok, a po změřených nových datech aktualizuje odhady parametrů.

Pro simulaci i jako model soustavy je zvolen regresní model 1. řádu

$$f(y_t | u_t, y_{t-1}),$$

kde y i u nabývají hodnot z množiny $\{1, 2\}$. Model je zadán tabulkou.

Program

... má následující strukturu:

- nejprve se zadají parametry úlohy a volí se typ vstupní veličiny - optimální s a bez znalosti skutečných parametrů a další pevné průběhy,
- dále se zadávají parametry pro simulaci a penalizační funkce pro řízení - obojí má formu tabulky,
- potom následují inicializace a smyčka diskrétního času ve které se periodicky opakují následující úlohy
 - optimalizace řízení na řídicím horizontu, kdy se počítá předpis pro řízení na celém horizontu; tady se také počítá střední hodnota výstupu - v diskrétním případě je postup následující: prvek po prvku se vynásobí tabulka penalizací + zbytek z minulého kroku tabulkou modelu a výsledné dva sloupce se sečtou,
 - aplikace řízení, kdy se použije poslední předpis pro výpočet řídicí veličiny; zde je také větvení podle kterého se použije nebo nepoužije vypočtená hodnota optimálního řízení,
 - simulace, kdy se aplikuje řízení a měří se hodnoty nového výstupu,
 - nakonec se provede odhad parametrů, kdy se přepočte statistika a počítají se bodové odhady.

Poznámka: Všimněte si, že časová smyčka probíhá trochu jinak než tomu bylo dříve. Není to tím, že se jedná o diskrétní model. Smyčku můžeme začít kdykoli, ale podle toho musíme dodat apriorní veličiny.

V úvodu programu je přepínač `control`, který volí typ řídicí veličiny.

Jako výsledek se zobrazují dva grafy. První graf zobrazuje řízený výstup soustavy - řídí se na nulu a druhý graf ukazuje vývoj odhadu parametrů modelu - horní obrázek ukazuje parametr a_1 u výstupní veličiny (rovná čára je jeho skutečná hodnota) a dolní obrázek totéž pro b_0 , koeficient u vstupu.

Doporučené pokusy

1. Pro předvolenou soustavu i pro další soustavy s jinými parametry porovnejte adaptivní řízení a řízení se známými parametry. Pokuste se soustavu i penalizační funkci zvolit podle nějakého pravidla - např. soustava nerada mění výstupní hodnoty a řídit se má tak, jak výstup tak i řízení měly malé hodnoty (tedy ne 2). Sledujte, do jaké míry řízení poslouchá to, co jste po něm chtěli.

Poznámka: Pokud neposlouchá, zkontrolujte zda je zadání takové, že vůbec lze vaše pokyny poslechnout.

2. Porovnejte kvalitu řízení (jak adaptivního tak i s pevnými parametry) pro různě dlouhý horizont řízení.

Poznámka: Vývoj řídicího zákona na horizontu je možno sledovat tak, že zastavíte program (myší klik na čírku před číslem příslušného řádku programu) a zobrazíte proměnnou us. Číslo by v okamžiku použití (tj. na konci) měla být ustálená.

```
// Adaptive control with discrete model
// -----
exec("ScIntro.sce",-1)

nd=200;           // length of control interval
nh=10;           // control horizon

control=5;       // type of control:
                 // 0=optimal with known parameters,
                 // 1=adaptive optimal,
                 // 2=no control, 3=step response,
                 // 4=sine input, 5=random

// model parameter
//yt(t) =1 =2   ut(t) yt(t-1)
thS=[ .71 .29   // 1   1
      .48 .52   // 1   2
      .55 .45   // 2   1
      .27 .73]; // 2   2
thS=[thS(:,1) 1-thS(:,1)];
// penalization
//yt(t) =1 =2   ut(t) yt(t-1)
om= [ .0  1   // 1   1
      .1  1   // 1   2
      .1 1.2  // 2   1
      .3 1.5]; // 2   2

J=0;             // initial value of criterion
y(1)=1;         // initial value of output (y=1 or y=2)
```

```

V=.1*ones(4,2); // initial value of statistics
p=V(:,1)./sum(V,2);
thE=[p 1-p]; // initial values of parameters

// time loop
thEt=[];
for t=2:nd
    fs=zeros(1,2); // initial conditio for Bellman function

    // CONTROL LAW COMPUTATION
    for i=nh:-1:1 // backwards loop for optimization
        if control==0, thC=thS; // parameters for control
        else thC=thE; end
        fp=om+ones(4,1)*fs; // penalty + reminder from last step
        // expectation
        f=sum((fp.*thC), 'c'); // expectation over y
        // minimization
        if f(1)<f(3), // for y(t-1)=1
            us(i,1)=1; fs(1)=f(1); // optimal control, minimum of criterion
        else
            us(i,1)=2; fs(1)=f(3); // optimal control, minimum of criterion
        end
        if f(2)<f(4), // for y(t-1)=2
            us(i,2)=1; fs(2)=f(2); // optimal control, minimum of criterion
        else
            us(i,2)=2; fs(2)=f(4); // optimal control, minimum of criterion
        end
    end
end

// CONTROL APPLICATION (simulation)
select control
    case 0, u(t)=us(1,y(t-1)); // optimal cont. with known pars
    case 1, u(t)=us(1,y(t-1)); // adaptive optimal control
    case 2, u(t)=1; // no control u=0
    case 3, u(t)=2; // response to step
    case 4, u(t)=floor(.99*sin(4*pi*t/nd))+2; // stepwise input
    case 5, u(t)=fix(double(rand(1,1,'unif')<.5))+1; // response to random input
end
// only one step of the designed control is applied
i=2*(u(t)-1)+y(t-1); // row index in parameter table
y(t)=2-fix(double(rand(1,1,'unif')<thS(i,1))); // generation of output (simulation)
J=J+om(i,y(t))*thC(i,y(t)); // criterion update

// ESTIMATION
V(i,y(t))=V(i,y(t))+1; // updte of statistics
p=V(:,1)./sum(V,2); // point est. of P(y=1)
thE=[p 1-p]; // estimate of model parameter
thEt=[thEt thE(:,1)];
end

```

```

// RESULTS
// final value of criterion (expectation)
disp(' ');
disp('The final value of criterion is ');
disp(J)
// plot of control and controlled output
set(figure(1), 'position',[100 100 500 500])
set(gcf(), "background",8)
s=(fix(4*nd/5)+1):nd;
plot(s,y(s), 'rx:',s,u(s), 'o:')
title('Adaptive control with a discrete model')
set(gca(), 'data_bounds',[min(s) max(s) .9 2.1])
legend('output','input')
// plot of estimated parameters
set(figure(2), 'position',[700 100 500 500])
set(gcf(), "background",8)
plot(thEt')
plot([1 nd],[thS(:,1) thS(:,1)])
title('Evolution of estimated parameters and their true values')

```

11.7.4 T84estS1S2

Odhadování s odlišnou strukturou modelu

- **odlišná struktura modelu pro simulaci a odhad:** Jestliže data simulujeme pomocí regresního modelu a jestliže pro odhad použijeme stejnou strukturu modelu (tj. stejný počet zpožděných veličin a u obou konstantu buď ano nebo ne), pak lze parametry simulovaného a odhadovaného modelu porovnávat a při dostatečném vybuzení vstupní veličinou (a simulovaným šumem) by odhady měly konvergovat k simulovaným hodnotám. Jestliže ale není shoda struktury simulovaného a odhadovaného modelu nebo dokonce jestliže pracujeme s reálnými daty, pak dochází k jisté aproximaci reálného (simulovaného) modelu pomocí modelu odhadovaného a odhadnuté parametry není s čím porovnat. Kontrolu úspěšného odhadu lze provést například jedno (nebo i více) krokovou předpovědí výstupu a porovnáním se skutečným výstupem.

Teorie

Odhad provádíme podle Bayesova vzorce. S postupně měřenými daty přepočítáváme statistiku a určujeme bodové odhady. Jednokrokovou předpověď počítanou dosazením do modelu s průběžně počítanými bodovými odhady porovnáujeme se simulovaným výstupem.

Program

... má následující strukturu:

- zadání parametrů úlohy včetně různých druhů vstupní veličiny,

- definice regresního vektoru pomocí anonymní funkce
 $\text{psi}=@(y,u,t) [y(t-1:-1:t-3) u(t-1:t-3) 1]'$;
 kterou je dále možno volat příkazem $\text{psi}(y,u,t)$, kde y a u jsou vektory výstupu a vstupu a t je okamžitý čas,
 stejně je pak dále definován regresní vektor Psi použitý v odhadu (simulace i odhad mají obecně odlišnou strukturu) - zde je možno volit buď shodu struktury nebo odlišné struktury,
- v časové smyčce se opakují úlohy: simulace - odhad - předpověď.

V úvodu programu je přepínače

- input, který volí typ vstupní veličiny a
- struc, kterým je možno zvolit shodu nebo neshodu struktury.

Jako výsledek se zobrazují dva grafy. První graf zobrazuje výstup soustavy a jeho předpověď a druhý graf ukazuje vývoj odhadu parametrů modelu a jejich skutečné hodnoty.

Doporučené pokusy

1. Pro předvolenou nebo i jinou soustavu porovnejte předpověď při shodě a neshodě struktury modelů.
2. V jednotlivých případech používejte různé vstupní signály a různou velikost šumu v simulaci. Udělejte závěry.

Poznámka: Dostatečné buzení soustavy je při identifikaci velmi důležité. Ty módy soustavy, které se neobjeví nemohou být odhadnuty. Např. konstantu modelu prakticky nelze odhadnout, jestliže vstupem je jen jeden skok.

Velký šum v simulaci pomáhá soustavu budit, na druhou stranu ale vnáší do soustavy velké chyby, což se projeví i v odhadu.

```
// Estimation with different model structure
// in simulation and estimation.
exec("ScIntro.sce",-1)
function z=psi(y,u,t), z=[y(t-1:-1:t-3) u(t-1:t-3) 1]'; endfunction
// regression vector for simulation
//psi=@(y,u,t) [y(t-1:-1:t-3) u(t-1:t-3) 1]';

nd=100; // number of data
input=3; // type of input: 1=randn, 2=sine, 3=step
struc=2; // model structure for estimation
// 1=full structure, 2=1st order, no const.

// simulated reg. coefficients
a1=.6; a2=-.2; a3=.06; // at output
b0=.5; b1=-.3; b2=.1; b3=-.2; // at input
k=1; s=.1; // constant, noise std.
y(1:3)=zeros(1,3); // initial conditions for output
select input // ciice of input
```

```

    case 1, u=rand(1,nd,'norm');
    case 2, u=sin(2*%pi*sin((1:nd)/nd));
    case 3, u=ones(1,nd);
end
ths=[a1 a2 a3 b0 b1 b2 b3 k]';           // simulated parameters
// regression vector for estimation
select struc
case 1,
    function z=Psi(y,u,t), z=psi(y,u,t); endfunction // the same as in simulation
case 2,
    function z=Psi(y,u,t), z=[y(t-1);u(t)]; endfunction // define your own reg. vec.
end // here: first order, no constant
npsi=length(Psi(y,u,4));
V=eye(npsi+1,npsi+1)*1e-8; // initial statistics

// time loop
for t=4:nd
    // SIMULATION
    ps=psi(y,u,t); // regression vector
    y(t)=ps'*ths+s*rand(1,1,'norm'); // simulator

    // ESTIMATION
    Ps=[y(t);Psi(y,u,t)]; // extended regression vector
    V=V+Ps*Ps'; // updt of stat.
    Vy=V(1,1); Vyp=V(2:$,1); // partitioning of stat.
    Vp=V(2:$,2:$);
    th(:,t)=inv(Vp)*Vyp; // estimates of reg. coef.
    r(t)=(Vy-Vyp'*inv(Vp)*Vyp)/t; // estimate of noise variance

    // PREDICTION
    yp(t)=Psi(y,u,t)*th(:,t); // predictor
end

// RESULTS
set(figure(1),'position',[100 100 500 500])
set(gcf(),"background",8)
plot(1:nd,y,1:nd,yp)
title('Output and its prediction')
legend('output','prediction')

set(figure(2),'position',[700 100 500 500])
set(gcf(),"background",8)
plot(th')
title('Evolution of parameter point estimates')

```

11.7.5 T85estStrR

Odhadování struktury modelu s reálnými daty

- **struktura modelu pro odhad:** Strukturou modelu nazýváme (i) výběr veličin, které v modelu vystupují, (ii) počet jejich zpožděných vzorků, které ovlivňují modelovanou veličinu v daném čase a (iii) přítomnost či nepřítomnost absolutního členu. Pro simulovaná data je optimální shoda struktury v simulaci a odhadu (npř. ve smyslu dobré předpovědi výstupu). Pro reálná data hledáme strukturu, která tato data nejlépe vystihne. Příliš malá struktura nedokáže vystihnout všechny módy chování systému, příliš velká struktura může vnášet závislosti mezi veličinami v regresním vektoru a vede na složitější výpočty, které zabírají více výpočetního času a vedou k větším numerickým chybám.

Teorie

Pro odhad struktury použijeme velmi jednoduchou metodu, založenou na posouzení vlivu veličin z regresního modelu na modelovanou veličinu. Odhad struktury začneme s nějakou maximální strukturou (třeba 10 řád s konstantou) a tuto strukturu redukuje.

Pro objektivní posouzení vlivu je třeba veličiny z regresního vektoru nejprve normovat tak, aby mely nulovou střední hodnotu a jednotkový rozptyl. Uvažujme regresní model

$$y_t = \theta_0 + \theta_1\psi_{1;t} + \theta_2\psi_{2;t} + \dots + \theta_n\psi_{n;t} + e_t,$$

kde $\psi_{i;t}$ jsou veličiny, jejichž vliv na y_t zkoumáme.

Tyto veličiny normujeme na základě apriori změřeného datového vzorku $\psi(t_I)$, kde t_I je velikost vzorku takto

$$v_{i;t} = \frac{\psi_{i;t} - m_i}{s_i},$$

kde $m_i = \frac{1}{t_I} \sum_{i=1}^{t_I} \psi_{i;t}$ je průměr veličiny ψ_i z apriorního vzorku dat a s_i je její směrodatná odchylka. Po tomto normování model přejde do tvaru

$$y_t = \vartheta_0 + \vartheta_1v_{1;t} + \vartheta_2v_{2;t} + \dots + \vartheta_{n;t}v_{n;t} + e_t,$$

kde ϑ_i jsou nové regresní koeficienty které získáme odhadem z normovaných dat.

Z modelu je vidět, že hodnota y_t je tvořena součtem příspěvků od jednotlivých veličin $v_{i;t}$. Protože veličiny jsou normované, jejich vliv na y_t je dán velikostí příslušného koeficientu ϑ_i . Malé koeficienty říkají, že veličina nemá na y_t žádný vliv a proto je vhodné ji z regresního vektoru vypustit.

Pro automatické vypouštění veličin lze použít následující postup: Normované veličiny (s předpokládaným normálním rozdělením) mají rozsah ± 3 . Zjistíme (z apriorních dat) jaká je směrodatná odchylka výstupu s_y . Konstanta modelu pokryje střední hodnotu y_t a zbytek výstupu má rozsah $\pm s_y$. Budeme požadovat, aby veličiny, které pokryjí méně než 1% rozsahu y_t byly vypuštěny. Potom pro hranici vypuštění platí

$$0.01s_y = \vartheta_i 3, \quad \rightarrow \quad \vartheta_i \doteq 0.003s_y.$$

Tedy, veličiny jejichž koeficienty jsou menší než 0.003 z regresního vektoru vypustíme.

Hladinu h (tady 1%) můžeme uvažovat obecně a zkusit ji měnit tak dlouho, až dostaneme „rozumný“ počet veličin. Potom platí: jestliže

$$\vartheta_i < \frac{hs_y}{3}$$

tak veličinu $v_{i,t}$ je třeba vypustit.

Pokud pro odhad normujeme i výstup y_t , je jeho rozsah také ± 3 a pravidlo pro odmítnutí veličiny je $\vartheta_i < h$. Tento způsob je také použit v programu.

Program

... pracuje s reálnými daty - intenzity dopravy v Legerově ulici v Praze. Tato data se na začátku programu loadují a část z nich slouží jako apriorní data, další část pro ověření pomocí předpovědi.

Dále jsou zadány parametry úlohy a inicializují se některé veličiny. Parametrem `ord` se volí počáteční řád modelu, který je dále redukován.

V další části programu se provádí normování dat a počáteční odhad z apriorních dat. Na základě tohoto odhadu se určuje struktura modelu.

Navržená struktura modelu se dále použije pro předpověď výstupu soustavy - tato předpověď je úlohou, pro kterou jsme odhad struktury dělali a která má zároveň ověřit, že i po redukcii jsme dostali model, který je dále dobře využitelný.

Jako výsledek se zobrazují dva grafy. První graf zobrazuje výstup soustavy a jeho předpověď a druhý sloupcový graf ukazuje hodnoty původních a redukováných parametrů. Modře jsou zobrazeny původní koeficienty, hnědě redukované. Tam, kde hnědý sloupek chybí, byl parametr a tedy také odpovídající veličina zrušena.

Programovací triky:

- `jth=find(abs(thi)>h)`; dá vektor indexů vektoru `thi` pro jejichž prvky platí že jejich absolutní hodnota je větší než h .

Doporučené pokusy

1. Zkuste měnit délku apriorních dat a počáteční řád modelu. Sledujte kvalitu předpovědi jako kritérium kvality předpovědi a tím i redukováného modelu.
2. Měňte hladinu h pro výběr platných regresních koeficientů. Sledujte velikost změn v redukcii parametrů při konstantní změně hladiny h .
3. Zkuste v programu změnit úlohu předpovědi na nějakou jinou, která by využívala redukováný model a tak ověřila jeho platnost. Případně zkuste existující jednokrokovou předpověď s bodovými parametry modelu rozšířit na vícekrokovou.
4. Ti, kdo se cítí pokročilí, mohou zkusit doplnit předpověď ne s bodovými odhady parametrů, ale s celou aposteriorní hp. Budu jim držet palce :)

```

// Model structure estimation (with real data sample)
exec("ScIntro.sce",-1)

//load data95 data
loadmatfile("data95.mat","data")

ni=200;           // number of initial data
nd=1000;         // number of data for estimation
h=.1;           // level for rejection variable from reg.vec.
ord=15;         // initial model order for str. estim.
                // 1=full structure, 2=1st order, no const.

// initial regression vector for str. est.
nPsi=ord+2;     // lengthof extended reg.vec.
V=eye(nPsi,nPsi)*1e-8; // initial statistics

yi=data(1,1:(ord+ni)); // initial data
yi=normOn(yi); // normalization of data

// INITIAL ESTIMATION
iPs=[1:ord 0];
for t=ord+1:ord+ni
    Ps=[yi(t:-1:t-ord) 1]';
    V=V+Ps*Ps';
end
Vy=V(1,1); Vyp=V(2:$,1); // partitioning of stat.
Vp=V(2:$,2:$);
thi=inv(Vp)*Vyp; // reg. coeff.
ri=(Vy-Vyp'*inv(Vp)*Vyp)/ni; // noise variance

jth=find(abs(thi)>h); // pointer to important reg. coefficients
yst=find(jth>0); // delays of y in reg.vec.
if isempty(find(jth==0)) // constant (yes or not)
    kst=[]; else kst=1;
end

// time loop for estimation and prediction
y=data(1,length(yi)+1:$); // data
V=eye(length(jth)+1)*1e-8; // initial statistics
for t=(ord+1):(ord+nd) // measured data follows the initial one
    // ESTIMATION
    pse=[y(t-yst) kst]'; // regression vector
    Ps=[y(t) pse]'; // extended regression vector
    V=V+Ps*Ps'; // updt of statstatistics
    Vy=V(1,1); Vyp=V(2:$,1); // partitioning of statistics
    Vp=V(2:$,2:$);
    th(:,t)=inv(Vp+1e-8*eye(Vp))*Vyp; // point estimates of reg. coef.
    r(t)=(Vy-Vyp'*inv(Vp+1e-8*eye(Vp))*Vyp)/t; // point estimate of noise variance
end

```

```

// PREDICTION
yp(t)=pse'*th(:,t);
end

// RESULTS
// output and its prediction
set(figure(1),'position',[100 100 500 500])
set(gcf(),"background",8)
s=1:nd;
plot(s,y(s),s,yp(s))
title('Estimation with optimal model structure')
legend('output','prediction')

// initial and reduced reg. coefficients
thr=thi(jth);
thp=zeros(1,nPsi-1);
thp(jth)=thr';
set(figure(2),'position',[700 100 500 500])
set(gcf(),"background",8)
bar([thi thp'])
title('Parameter reduction - selected items are double')

// prints
disp(' ')
disp('Result of parameter reduction')
disp('-----')
printf('output y, delayed by'), disp(yst')
if isempty(kst), disp('constant, no')
else disp('constant, yes'),
end
disp(' ')
disp('Final estimates of reg.coef. ')
Theta=th(:, $)
disp(' ')
disp('Prediction std')
s=1:nd;
err=(y(s)-yp(s)');
Ep=sqrt(err*err'/nd);
disp(Ep)

```

11.7.6 T86estStrS

Odhadování struktury modelu se simulovanými daty

Tato úloha zcela odpovídá úloze **T85estStrR** (Odhad struktury s reálnými daty) Rozdíl mezi těmito úlohami je v tom, že o generování dat se musíme sami postarat. Výhodou ale je, že víme, k jaké struktuře bychom se měli dopracovat nebo můžeme zjistit, jak moc je možno námi zadanou strukturu modelu pro simulaci „podrobit kompresi“ aniž bychom ztratili příliš informace.

Teorie

Je stejná jako pro úlohu **T85estStrR**. Zde se zmíníme pouze o odlišnostech v programu.

Program

... vychází z programu **T85estStrR.m** pouze de dodělána simulace dat.

Na začátku programu je možno zvolit: řád maximálního modelu `ord` a typ vstupního signálu (náhodný, sinusový a skokový). Dále jsou zde běžné parametry úlohy a inicializace.

Následuje generování apriorních dat a jejich normalizace.

Potom přichází cyklus napočítávání statistiky pro apriorní odhad, za cyklem je výpočet bodových odhadů.

Bodové odhady se dále využijí pro odhad struktury modelu, zcela stejně, jako v úloze s reálnými daty.

Další časová smyčka je věnována následné úloze, pro kterou jsme strukturu modelu odhadovali. Tak jako v úloze s reálnými daty se jedná o jednokrokovou předpověď výstupu.

Zobrazení výsledků je také stejné - porovnání výstupu s jeho předpovědí a zobrazení původních a redukovaných parametrů.

Programovací triky:

- v simulaci se používá anonymní funkce definovaná
`psi=@(y,u,t) [y(t-1:-1:t-2) u(t-1:t-2) 1]'`; která se volá příkazem
`ps=psi(y,u,t)`
- `jth=find(abs(thi)>h)`; dá vektor indexů vektoru `thi` pro jejichž prvky platí že jejich absolutní hodnota je větší než `h`.

Doporučené pokusy

1. Zkuste měnit délku apriorních dat a parametry simulované soustavy, včetně řádu modelu. Sledujte kvalitu předpovědi jako kritérium kvality předpovědi a tím i redukovaného modelu.
2. Pro zvolenou soustavu a velký řád maximálního modelu (např. 10) měňte hladinu h pro výběr platných regresních koeficientů. Sledujte podobnost modelu s odhadnutou strukturou a modelu použitého v simulaci.
3. Zkuste v programu změnit úlohu předpovědi na nějakou jinou, která by využívala redukovaný model a tak ověřila jeho platnost. Případně zkuste existující jednokrokovou předpověď s bodovými parametry modelu rozšířit na vícekrokovou.
4. Ti, kdo se cítí pokročilí, mohou zkusit doplnit předpověď ne s bodovými odhady parametrů, ale s celou aposteriorní hp . Budu jim taky držet palce :) .

```

// Model structure estimation (with simulated data)
exec("ScIntro.sce",-1)

ni=20; // number of initial data
nd=100; // number of data for estimation
h=.1; // level for rejection variable from reg.vec.
ord=5; // initial model order for str. estim.
input=1; // type of input: 1=randn, 2=sine, 3=step

// simulated reg. coefficients (2nd order + const.)
a1=.6; a2=-.2; // coeffs at output
b0=.5; b1=-.3; b2=.1; // coeffs at input
k=1; s=.1; // constant, noise std.
select input // selection of input
    case 1, u=rand(1,nd+ord,'norm');
    case 2, u=sin(2*%pi*sin((1:nd+ord)/(nd+ord)));
    case 3, u=ones(1,nd+ord);
end
y=zeros(1,nd+ord); // initial conditions for output
ths=[a1 a2 b0 b1 b2 k]'; // simulated parameters
// regression vector for simulation (anonymous function)
//psi=@(y,u,t) [y(t-1:-1:t-2) u(t-1:t-2) 1]';
function z=psi(y,u,t), z=[y(t-1:-1:t-2) u(t-1:t-2) 1]'; endfunction
// initial regression vector for str. est.
nPsi=2*(ord+1)+1; // length of extended reg.vect.
V=eye(nPsi,nPsi)*1e-8; // initial information matrix

// INITIAL DATA
ui=rand(1,ord+ni,'norm'); // input definition (system excitation)
yi(1:ord)=rand(1,ord,'norm'); // initial condition for initial data
// initial simulation
for t=ord+1:ord+ni
    ps=psi(yi,ui,t); // regression vector
    yi(t)=ps'*ths+s*rand(1,1,'norm'); // simulator
end

// normalization of data
yi=normOn(yi); // normalization of data to
ui=normOn(ui); // mean=0 and std=1

// INITIAL ESTIMATION
for t=ord+1:ord+ni
    Ps=[yi(t:-1:t-ord) ui(t:-1:t-ord) 1]'; // extended reg.vec.
    V=V+Ps*Ps'; // update of statistics
end
Vy=V(1,1); Vyp=V(2:$,1); // partitioning of stat.
Vp=V(2:$,2:$);
thi=inv(Vp)*Vyp; // reg. coeff.
ri=(Vy-Vyp'*inv(Vp)*Vyp)/ni; // noise variance

```

```

// STRUCTURE ESTIMATION
iPs=[1:ord (0:ord)+50 0]; // delays o data in reg.vec. (u shifted by 50)
jth=find(abs(thi)>h); // indexes of important reg.coef.
jPs=iPs(jth); // selected delays of variables (>50 are u)
jy=find(jPs<50 & jPs>0); yst=jPs(jy); // yst=delays of y
ju=find(jPs>=50); ust=jPs(ju)-50; // ust=delays of u
if isempty(find(jPs==0)), kst=[]; // constant yes/no
else kst=1; end

// time loop for dynamic estimation
V=eye(length(jth)+1)*1e-8; // initial statistics
for t=ord+1:ord+nd
// SIMULATION
pss=psi(y,u,t); // pss=gev.vec. for simulation
y(t)=pss'*ths+s*rand(1,1,'norm'); // simulator

// ESTIMATION
pse=[y(t-yst) u(t-ust) kst]'; // reg.vec. for estimation
Ps=[y(t) pse]'; // extended reg.vec.
V=V+Ps*Ps'; // updt of statistics
Vy=V(1,1); Vyp=V(2:$,1); // partitioning of stat.
Vp=V(2:$,2:$);
th(:,t)=inv(Vp+1e-8*eye(Vp))*Vyp; // reg. coeff.
r(t)=(Vy-Vyp'*inv(Vp+1e-8*eye(Vp))*Vyp)/t; // noise variance

// PREDICTION
yp(t)=pse'*th(:,t); // predictor
end

// RESULTS
// plot of output and its prediction
set(figure(1),'position',[100 100 500 500])
set(gcf(),"background",8)
s=1:nd;
plot(s,y(s),s,yp(s))
title('Esimation with optimal model structure')
legend('output','prediction')

// all and selected reg. ceoefficients
thr=thi(jth);
thp=zeros(1,nPsi-1);
thp(jth)=thr';
set(figure(2),'position',[700 100 500 500])
set(gcf(),"background",8)
bar([thi thp'])
title('Parameter reduction - selected items are double')

// prints
disp(' ')

```

```

disp('Result of parameter reduction')
disp('-----')
printf('output y, delayed by'), disp(yst)
printf('input u, delayed by '), disp(ust)
if isempty(kst), disp('constant, no')
else          disp('constant, yes'),
end

disp(' ')
disp('Final estimates of reg.coef. ')
Theta=th(:, $)
disp(' ')
disp('Prediction std')
s=1:nd;
err=(y(s)-yp(s)');
Ep=sqrt(err*err'/nd);
disp(Ep)

```

11.7.7 T87estDiLo

Předpověď s diskretním a logistickým modelem

Teorie

Oba modely, jak diskretní tak i logistický pracují s diskretní modelovanou veličinou. Přitom u diskretního modelu je třeba, aby i veličiny v regresním vektoru byly diskretní, u logistického modelu se v regresním vektoru připouštějí i spojité veličiny. Nicméně, i v logistickém modelu mohou být také všechny veličiny diskretní. Rozdíl mezi diskretním a logistickým modelem je potom ten, že diskretní model vyhodnocuje pravděpodobnosti všech možných stavů systému (tj. kombinací hodnot všech veličin), zatímco logistický model mezi stavy vytvoří určitou dělicí křivku a tím je rozdělí na dvě podmnožiny. Jedné přiřadí $y_t = 1$ a druhé $y_t = 2$. Jde tedy o určitý druh klastrování a klasifikace.

V tomto příkladě provedeme obě zmíněné úlohy a výsledky porovnáme.

Program

... má následující části:

Na začátku se zadává počet dat a generují se diskretní data.

V první části programu se provádí odhad diskretního modelu a po užití všech dat se počítají bodové odhady. Funkce `sum(V,2)` sčítá prvky matice V po řádcích. Výsledkem je tedy sloupcový vektor.

Bodové odhady se pak použijí pro výpočet jednokrokové předpovědi výstupu.

Zobrazení výsledků je také stejné - porovnání výstupu s jeho předpovědí a zobrazení původních a redukovanych parametrů.

Jako výsledek předpovědi s diskretním modelem se tiskne počet neshod výstupu s jeho předpovědí a graf hodnot výstupu a předpovědi.

V druhé části programu je stejná úloha řešena pomocí logistické regrese. Pro její odhad i předpověď jsou využity funkce Matlabu a to `b=mnrfit(x,y)`, která spočte hodnoty parametru `b` a `py=mnrval(b,x)`; která pro dané regresní vektory `x` a odhadnuté parametry `b` spočte pravděpodobnosti toho, že $y_t = 1$. Bodové odhady se pak určí podle toho, která z možných hodnot výstupu `{1, 2}` je pravděpodobnější.

Jako výsledky se opět tiskne počet omylů v předpovědi a graf s výstupem a jeho předpovědí.

Programovací triky:

- Pro odhad diskretního modelu používáme regresní vektor délky 3 (můžeme použít i obecný regresní vektor délky n). Co pro odhad (i simulaci) potřebujeme je pro konkrétní číselnou konfiguraci regresního vektoru určit řádek tabulky, kam tento vektor patří. Budeme uvažovat, že hodnoty všech veličin regresního vektoru x_1, x_2, \dots, x_n začínají od jedné a počty jejich hodnot jsou m_1, m_2, \dots, m_n . Potom řádek i tabulky modelu určíme podle vzorce

$$i = m_n m_{n-1} \cdots m_2 (x_1 - 1) \cdots + m_n m_{n-1} (x_{n-2} - 1) + m_n (x_{n-1} - 1) + x_n$$

viz funkce `psi2row`.

- Jestliže regresní vektory seřadíme po sloupcích do matice x , a platí předchozí předpoklady, pak počet různých hodnot regresních vektoru zjistíme příkazem

$$m = \max(x);$$

Doporučené pokusy

1. Pro různé simulované modely (s konkrétní funkcí - např. soustava, která má nerada dvakrát po sobě stejný výstup) sledujte rozdíly mezi kvalitou předpovědi u obou modelů. Kromě toho je také třeba sledovat velikost tabulky diskretního modelu.
2. Zkuste program dokončit tak, aby oba odhadnuté modely sloužily jako klasifikátory, tj. aby nově příchozím regresním vektorům přiřazovaly jejich příznak třídy - buď 1 nebo 2. Najděte praktickou interpretaci této klasifikační úlohy.

```
// Estimation and prediction
// with discrete and logistic model
// -----
exec("ScIntro.sce",-1)

// SIMULATION
nd=200; // number of data for estimation
x1=fix(6*rand(1,nd,'unif'))+1;
x2=(rand(1,nd,'unif')>.3)+1;
x3=fix(3*rand(1,nd,'unif')+2*rand(1,nd,'unif'))+1;
x=[x1' x2' x3']; // gression vextors (in rows)
y=((2*x1-5*x2+x3-3+2*rand(1,nd,'norm'))<0)+1; // output
```

```

// DISCRETE MODEL
// estimation
n=max(x,'r'); // numbers of different values in reg.vecs
V=.01*ones(prod(n),2); // initial statistics for discrete model
for t=1:nd
    i=n(2)*n(3)*(x(t,1)-1)+n(3)*(x(t,2)-1)+x(t,3); // row index of V
    V(i,y(t))=V(i,y(t))+1; // update of information matrix
end
p=V(:,1)./sum(V,2);
th=[p 1-p]; // estimated parameters of disc. model

// prediction
np=50;
for t=1:nd
    i=n(2)*n(3)*(x(t,1)-1)+n(3)*(x(t,2)-1)+x(t,3); // row index of V
    [xxx yp(t)]=max(th(i,:)); // prediction
end

// RESULTS FOR DISCRETE MODEL
ERR_discrete=sum(y~=yp)
disp(ERR_discrete,'ERR_discrete (wrong estimates)')
subplot(121)
plot(1:np,y(1:np),'.',1:np,yp(1:np),'o')
set(gcf(),'position',[60 60 1200 500])
title('Estimation with discrete model')
legend('output','prediction')
set(gca(),"data_bounds",[0 np+1 .9 2.1])

// _____
// LOGISTIC MODEL
[py, yp, yr, b]=lrEst([],y,x);

///// estimation
//[b,dev,stats] = mnrfi(x,y); // solution to logistic regression
///// prediction
//py=mnrval(b,x);
//[xxx yL]=max(py'); // solution to logistic prediction
//
// RESULTS FOR LOGISTIC MODEL
ERR_logist=sum(y~=yr)
disp(ERR_logist,'ERR_logist (wrong estimates)')
subplot(122)
plot(1:np,y(1:np),'.',1:np,yr(1:np),'o')
legend('output','prediction')
title('Estimation with logistic regression')
set(gca(),"data_bounds",[0 np+1 .9 2.1])

```

11.8 Pomocné procedury

— DISKRÉTNÍ MODEL —

11.8.1 dsim

Simulace diskrétního modelu

Realizuje se generování výstupu s hodnotami $\{1, 2\}$ diskrétního systému popsaného pravděpodobnostní funkcí $f(y_t|u_t, y_{t-1})$ zadané např. tabulkou

| u_t, y_{t-1} | $y_t = 1$ | $y_t = 2$ |
|----------------|-----------|-----------|
| 1, 1 | 0.2 | 0.8 |
| 1, 2 | 0.6 | 0.4 |
| 2, 1 | 0.1 | 0.9 |
| 2, 2 | 0.3 | 0.7 |

Program

... využívá funkci `psi2row` která k regresnímu vektoru $[u_t, y_{t-1}]$ určí číslo řádku tabulky která tomuto regresnímu vektoru odpovídá a podle jehož pravděpodobností se výstup generuje.

Příkaz `yy=rand<th(i,1)` generuje hodnoty 1 a 0 s pravděpodobnostmi $th(i,1)$ a $1 - th(i,1)$, příkaz `y=2-yy` tyto hodnoty převádí na 1 a 2.

```
function y=dsim(u,y1,th)
// y=dsim(u,y1,th)    Simulation of a discrete system
// y    new output
// u    input
// y1   old nput

    i=psi2row([u y1]); // index of conditional regressor
    yy=rand(1,1,'unif')<th(i,1); // probability of conditional regressor
    y=2-yy;           // output (with values 1, 2)
endfunction
```

11.8.2 psi2row

K diskrétnímu regresnímu vektoru počítá odpovídající číslo řádku jeho tabulky modelu

Diskrétní model je reprezentován pravděpodobnostní tabulkou. Regresní vektor určuje řádek tabulky s pravděpodobnostmi, podle kterých se simuluje výstup. Tato funkce k danému regresnímu vektoru počítá číslo řádku.

```

function i=psi2row(x,b)
// i=psi2row(x,b) i is row number of a model table corresponding
// to regression vector x with the base b;
// elements of x(i) are 1,2,...,nb(i)
// it is based on the relation
//  $i=b(n-1)b(n-2)\dots b(1)(x(n)-1)+\dots+b(1)(x(2)-1)+x(1)$ 

n=length(x);
if argn(2)<2, b=2*ones(1,n); end
b=[b(2:n) 1];
i=0;
for j=1:n
    i=(i+x(j)-1)*b(j);
end
i=i+1;
endfunction

```

11.8.3 row2psi

K číslu řádku tabulky modelu se počítá odpovídající regresní vektor

Diskrétní model je reprezentován pravděpodobnostní tabulkou. Regresní vektor určuje řádek tabulky s pravděpodobnostmi, podle kterých se simuluje výstup. Tato funkce k danému číslu řádku počítá regresní vektor.

```

function x=row2psi(i,b)
// x=row2psi(i,b) generates a discrete regression vector
// with a base b that corresponds to the
// i-th row of a table of the discrete model
// it is based on the relation
//  $i=b(n-1)b(n-2)\dots b(1)(x(n)-1)+\dots+b(1)(x(2)-1)+x(1)$ 

if isscalar(b), n=b; b=2*ones(1,n);
else n=length(b); end
if i>prod(b)
    disp('ERROR: The row number is too big')
    return
end
i=i-1;
for j=1:n
    i=i/b(n-j+1);
    x(n-j+1)=round((i-fix(i))*b(n-j+1)+1);
    i=fix(i);
end
endfunction

```

11.8.4 allRows

Generování všech regresních vektorů s bází b

Generuje všechny možné kombinace hodnot regresních vektorů $[v_1, v_2, \dots, v_n]$ kde $v_1 \in \{1, 2, \dots, b_1\}$, $v_2 \in \{1, 2, \dots, b_2\}$, \dots $v_n \in \{1, 2, \dots, b_n\}$, kde $b_i > 1$, celá. Vektor $b = [b_1, b_2, \dots, b_n]$ nazýváme bází regresních vektorů.

Program

... využívá funkci `row2psi` která k pořadí řádku tabulky popisující diskrétní model přiřazuje příslušný regresní vektor $[v_1, v_2, \dots, v_n]$ v rámci zadané báze regresních vektorů.

```
function x=allRows(b)
// x=allRows(b)      generates rows of the table x
//                  of all discrete regression vectors
//                  with the base b (elements are 1...nb)

    if isscalar(b), b=2*ones(1,b); end
    m=prod(b);
    n=length(b);
    x=zeros(m,n);
    for i=1:m
        x(i,:)=row2psi(i,b);
    end
endfunction
```

— SPOJITÝ MODEL —

11.8.5 Gauss2

Rozklad matice na horní trojúhelník diagonálu a dolní trojúhelník

Počítá hodnotu normální hp pro dané dvourozměrné $x = [x_1, x_2]^T$ a zadané charakteristiky - střední hodnotu m a kovarianční matici R .

Program

... počítá jednak hodnotu hp , jednak hodnotu logaritmu hp . Důvod je ten, že hustota normálního rozdělení rychle utíká k nule a hodnoty pro dva různé body x nelze dobře rozlišit. Pak je potřeba používat logaritmy.

```
function [p,Lp]=Gauss2(x,m,R)
// p=Gauss2(x,m,R)  Two dimensional Gaussian pdf
// p - value of Gauss pdf
// Lp - logarithm of p
// m - mean
```

```

// R - covariance

x=x(:); m=m(:); // column vectors
k=1/(2*%pi*sqrt(det(R))); // constant part
ex=(x-m)'*inv(R)*(x-m); // exponential part
p=k*exp(-.5*ex); // the whole pdf
Lp=log(k)-.5*ex;
endfunction

```

11.8.6 v2th

Výpočet bodových odhadů parametrů jednorozměrného regresního modelu ze statistiky

Teorie

Statistiku rozdělíme

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

kde pro statický model V_y je číslo, $V_{y\psi}$ je sloupcový vektor a V_ψ je matice 2×2 .

Bodové odhady se spočtou takto:

$$\hat{\theta} = V_\psi^{-1} V_{y\psi} \text{ a } \hat{r} = \frac{V_y - V'_{y\psi} V_\psi^{-1} V_{y\psi}}{n}$$

kde $\hat{\theta}$ je odhad regresních koeficientů a \hat{r} je odhad kovariance šumu, n je počet dat.

Funkce rozdělí informační matici V na odpovídající části a spočte bodové odhady regresních koeficientů, případně bodový odhad rozptylu šumu.

```

function [th, s2]=v2th(V, n)
// th=v2th(V)
// Computation of point parameter estimates
// from information matrix for scalar output
// th point estimates of regression coefficients
// s2 point estimate of noise variance
// V information matrix (statistics)
// n number of data used

Vy=V(1,1); // partitioning ov V
Vyf=V(2:end,1);
Vf=V(2:end,2:end);
th=inv(Vf)*Vyf; // reg. coefficients
if nargin==2
s2=(Vy-Vyf'*inv(Vf)*Vyf)/n; // noise variance
else s2=NaN; end // if n is not set, only
// reg.coefs are computed
endfunction

```

11.8.7 v2th2

Výpočet bodových odhadů parametrů dvourozměrného regresního modelu ze statistiky

Teorie

Statistiku rozdělíme

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

kde pro statický model V_y je matice 2×2 , $V_{y\psi}$ je řádkový vektor a V_ψ je číslo.

Bodové odhady se spočtou takto:

$$\hat{\theta} = \frac{1}{V_\psi} V_{y\psi} \text{ a } \hat{r} = \frac{V_y - V'_{y\psi} V_{y\psi} / V_\psi}{n}$$

kde $\hat{\theta}$ je odhad regresních koeficientů a \hat{r} je odhad kovariance šumu, n je počet dat.

Funkce rozdělí informační matici V na odpovídající části a spočte bodové odhady regresních koeficientů, případně bodový odhad kovariance šumu.

```
function [th,s2]=v2th2(V,n)
// [th,s2]=v2th2(V,n)
// Computation of point parameter estimates
// from information matrix for
// two-dimensional y and static model
// V information matrix
// n data counter

// paratitionong of V
Vy=V(1:2,1:2);
Vyf=V(3,1:2);
Vf=V(3,3);
// regression coefficients
th=Vyf/(Vf+1e-8);
// covariance
if argn(2)>1
s2=(Vy-Vyf'*th)/n;
else s2=%nan; end
endfunction
```

— STAVOVÝ MODEL —

11.8.8 Kalman

Kalmanův filtr

Provádí jeden krok odhadu stavu z měřených veličin.

Program

... sleduje vývoj charakteristik popisu stavové veličiny podle teorie.

```
function [xt,xf,Rx,yp]=Kalman(xt,yt,ut,M,N,A,B,Rw,Rv,Rx)
// [xt,xf,ey,Rx,yp]=Kalman(xt,yt,ut,M,N,A,B,Rw,Rv,Rx)
// Kalman filter for state estimation with the model
//
// xt = M*xt + N*ut + w
// yp = A*xt + B*ut + e
// xt state
// Rx state estimate covariance matrix
// yp output prediction
// yt output
// ut input
// M,N state model parameters
// A,B output model parameters
// Rw state model covariance
// Rv output model covariance

yp=A*xt+B*ut; // output prediction
Ry=Rv+A*Rx*A'; // noise covariance update
Rx=Rx-Rx*A'*inv(Ry)*A*Rx; // state est. covariance update
ey=yt-yp; // prediction error
KG=Rx*A'*inv(Rv); // Kalman gain
xt=xt+KG*ey; // data update of the state
xf=xt; // filtered state
xt=M*xt+N*ut; // time update of the state
Rx=Rw+M*Rx*M'; // time updt. of state covariance
endfunction
```

— LOGISTICKÝ MODEL —

11.8.9 logit

Funkce logit

Počítá hodnotu funkce *logit*

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right),$$

kde p je pravděpodobnost a $\text{logit}(p) \in R$.

```
function z=logit(p)
// z=logit(p) function logit log(p/(1-p))
// z value of logit (from R)
// p probability (from (0,1))
```

```
    z=log(p./(1-p));  
endfunction
```

11.8.10 logit_inv

Funkce inverzní k funkci logit

Počítá hodnotu funkce inverzní k *logit*

$$\text{logit_inv}(z) = \frac{\exp\{z\}}{1 + \exp\{z\}},$$

kde z je reálné číslo a p je pravděpodobnost.

```
function p=logit_inv(z)  
    // p=logit_inv(z)    function inverse logit: logit^(-1)  
    // p  value of the function (probability from (0,1))  
    // z  argument of the function (from R)  
  
    p=exp(z)./(1+exp(z));  
endfunction
```

11.9 Dcretization of differential equations

Tato část bude přeložena, jen co zase naberu síly :) Zatím alespoň anglicky.

11.9.1 Elementary differential equations

First order equations The first order homogeneous differential equation with constant coefficients has the form

$$y' + a_0y = 0, \quad y(0) = \tilde{y}_0 \quad (11.3)$$

Characteristic equation is linear with unique solution

$$\lambda + a_0 = 0 \rightarrow \lambda = -a_0 \quad (11.4)$$

The solution to the differential equation (11.3) is

$$y(t) = \tilde{y}_0 e^{\lambda t} \quad (11.5)$$

Second order equations The second order homogeneous differential equation with constant coefficients has the form

$$y'' + a_1y' + a_0y = 0, \quad y(0) = \tilde{y}_0, \quad y'(0) = \tilde{y}_1 \quad (11.6)$$

Characteristic equation is quadratic

$$\lambda^2 + a_1\lambda + a_0 = 0 \quad (11.7)$$

with the following types of solution

1. Two different real roots λ_1 and λ_2

The equation (11.6) is

$$y'' - (\lambda_1 + \lambda_2)y' + \lambda_1\lambda_2y = 0$$

The solution is

$$y(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}, \quad (11.8)$$

where the coefficients c can be obtained as a solution of the set of linear equations

$$\begin{aligned} c_1 + c_2 &= \tilde{y}_0 \\ \lambda_1 c_1 + \lambda_2 c_2 &= \tilde{y}_1 \end{aligned}$$

which gives the solution

$$c_1 = (\lambda_2 y_0 - y'_0)/(\lambda_2 - \lambda_1) \quad \text{and} \quad c_2 = (\lambda_1 y_0 - y'_0)/(\lambda_1 - \lambda_2)$$

2. One real double root λ

The equation (11.6) is

$$y'' - 2\lambda y' + \lambda^2 y = 0$$

The solution is

$$y(t) = c_1 e^{\lambda t} + c_2 t e^{\lambda t}, \quad (11.9)$$

where the coefficients c can be obtained as a solution of the set of linear equations

$$\begin{aligned} c_1 &= \tilde{y}_0 \\ \lambda c_1 + c_2 &= \tilde{y}_1 \end{aligned}$$

which gives the solution

$$c_1 = \tilde{y}_0 \quad \text{and} \quad c_2 = \tilde{y}_1 - \lambda \tilde{y}_0$$

3. Two complex roots $\lambda_1 = \rho + \omega i$ and $\lambda_2 = \rho - \omega i$

The equation (11.6) is

$$y'' - 2\rho y' + \rho^2 + \omega^2 = 0$$

The solution is

$$y(t) = c_1 e^{\rho t} \cos(\omega t) + c_2 e^{\rho t} \sin(\omega t), \quad (11.10)$$

where the coefficients c can be obtained as a solution of the set of linear equations

$$\begin{aligned} c_1 &= \tilde{y}_0 \\ \rho c_1 + \omega c_2 &= \tilde{y}_1 \end{aligned}$$

which gives the solution

$$c_1 = \tilde{y}_0 \quad \text{and} \quad c_2 = (\tilde{y}_1 - \rho \tilde{y}_0) / \omega$$

11.9.2 Elementary difference equations

Here, we will consider discrete time k for which it holds $t = kT$, where t is continuous time and T is a fix period of sampling.

First order equations The first order homogeneous difference equation with constant coefficients has the form

$$y_{k+1} + a_0 y_k = 0, \quad y_0 = \tilde{y}_0 \quad (11.11)$$

Characteristic equation is linear with unique solution

$$\lambda + a_0 = 0 \quad \rightarrow \quad \lambda = -a_0 \quad (11.12)$$

The solution to the differential equation (11.11) is

$$y_k = \tilde{y}_0 \cdot \lambda^k \quad (11.13)$$

Second order equations The second order homogeneous difference equation with constant coefficients has the form

$$y_{k+2} + a_1 y_{k+1} + a_0 y_k = 0, \quad y_0 = \tilde{y}_0, \quad y_1 = \tilde{y}_1 \quad (11.14)$$

Characteristic equation is quadratic

$$\lambda^2 + a_1 \lambda + a_0 = 0 \quad (11.15)$$

with the following types of solution

1. **Two different real roots** λ_1 and λ_2

The equation (11.14) is

$$y_{k+2} - (\lambda_1 + \lambda_2)y_{k+1} + \lambda_1\lambda_2y = 0$$

The solution is

$$y_k = c_1\lambda_1^k + c_2\lambda_2^k, \quad (11.16)$$

where the coefficients c can be obtained as a solution of the set of linear equations

$$\begin{aligned} c_1 + c_2 &= \tilde{y}_0 \\ \lambda_1 c_1 + \lambda_2 c_2 &= \tilde{y}_1 \end{aligned}$$

which gives the solution

$$c_1 = (\lambda_2\tilde{y}_0 - \tilde{y}_1)/(1 - \lambda_1) \quad \text{and} \quad c_2 = (\lambda_1\tilde{y}_0 - \tilde{y}_1)/(1 - \lambda_2)$$

2. **One real double root** λ

The equation (11.14) is

$$y_{k+2} - 2\lambda y_{k+1} + \lambda^2 y = 0$$

The solution is

$$y_k = c_1\lambda^k + c_2k\lambda^k, \quad (11.17)$$

where the coefficients c can be obtained as a solution of the set of linear equations

$$\begin{aligned} c_1 &= \tilde{y}_0 \\ \lambda c_1 + \lambda c_2 &= \tilde{y}_1 \end{aligned}$$

which gives the solution

$$c_1 = \tilde{y}_0 \quad \text{and} \quad c_2 = \tilde{y}_1/\lambda - \tilde{y}_0$$

3. **Two complex roots** $\lambda_1 = \rho + \omega i$ and $\lambda_2 = \rho - \omega i$

The equation (11.14) is

$$y_{k+2} - 2\rho y_{k+1} + \rho^2 + \omega^2 = 0$$

The solution is

$$y_k = |c|^k [c_1 \cos(\omega k) + c_2 \sin(\omega k)], \quad (11.18)$$

where the coefficients c can be obtained as a solution of the set of linear equations

$$\begin{aligned} c_1 &= \tilde{y}_0 \\ c_1 |Re\lambda| + c_2 |Im\lambda| &= \tilde{y}_1 \end{aligned}$$

which gives the solution

$$c_1 = y_0 \quad \text{and} \quad c_2 = (\tilde{y}_1 - \tilde{y}_0 |Re\lambda|)/|Im\lambda|$$

11.9.3 Discretization of a continuous model

Our aim is to construct a discrete regression model whose output is a sampled output of the corresponding continuous one - homogeneous differential equation of 1st or 2nd order with constant coefficients. We will call this task **discretization**.

Let us denote the continuous time by t and the discrete time by k . It holds

$$t = kT, \quad T \text{ is a period of sampling.}$$

First order

Consider a homogeneous differential equation with constant coefficient

$$y' + a_0y = 0, \quad y_0 = \tilde{y}_0. \quad (11.19)$$

Then the equivalent difference equation (whose response is the sampled response to the differential one) is

$$y_{k+1} = A_0y_k, \quad \text{where } A_0 = e^{-a_0T}. \quad (11.20)$$

Solution: The solution to the differential equation is

$$y_t = \tilde{y}_0 e^{-a_0t}.$$

To get the discrete version of the solution, we set $t = k$ for actual sample and $t + T = k + 1$ for the shifted one. So, for the actual sample, the solution the same but the substitution k for t and for the shifted sample it holds

$$y_{k+1} = y_{t+T} = \tilde{y}_0 e^{-a_0(t+T)} = \tilde{y}_0 e^{-a_0t} e^{-a_0T} = e^{-a_0T} y_k$$

which proves (11.20). \triangleleft

Second order

Two distinct real roots

Let us consider a homogeneous differential equation with constant coefficients

$$y'' + a_1y' + a_0y = 0, \quad y_0 = \tilde{y}_0, \quad y'_0 = \tilde{y}_1 \quad (11.21)$$

whose characteristic equation $\lambda^2 + a_1\lambda + a_0 = 0$ has two different real roots λ_1 and λ_2 .

Then the equivalent difference equation (whose response is the sampled response to the differential one) is

$$y_{k+2} = A_1y_{k+1} + A_0y_k, \quad (11.22)$$

where

$$A_1 = e^{\lambda_1T} + e^{\lambda_2T}, \quad A_0 = -e^{(\lambda_1+\lambda_2)T} \quad (11.23)$$

Solution: A response to the considered continuous model is

$$y_t = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}.$$

Sampling with $t = kT$ and the denotation $y_k = y_{kT}$ gives

$$y_k = c_1 e^{\lambda_1 kT} + c_2 e^{\lambda_2 kT}$$

This sampled response must obey the difference equation

$$y_{k+2} = A_1y_{k+1} + A_0y_k.$$

We express still the shifted responses

$$\begin{aligned} y_{k+1} &= c_1 e^{\lambda_1 kT} e^{\lambda_1 T} + c_2 e^{\lambda_2 kT} e^{\lambda_2 T} \\ y_{k+2} &= c_1 e^{\lambda_1 kT} e^{\lambda_1 2T} + c_2 e^{\lambda_2 kT} e^{\lambda_2 2T} \end{aligned}$$

and notice that they all are expressed in the basis with items $e^{\lambda_1 kT}$ and $e^{\lambda_2 kT}$. Thus we substitute into the difference equation and obtain

$$c_1 e^{\lambda_1 kT} e^{\lambda_1 2T} + c_2 e^{\lambda_2 kT} e^{\lambda_2 2T} = A_1 (c_1 e^{\lambda_1 kT} e^{\lambda_1 T} + c_2 e^{\lambda_2 kT} e^{\lambda_2 T}) + A_0 (c_1 e^{\lambda_1 kT} + c_2 e^{\lambda_2 kT}).$$

The coefficients B and A will be obtained by the comparison of items with the same basis element. We obtain the following system of equations

$$\begin{aligned} c_1 e^{\lambda_1 2T} &= A_1 c_1 e^{\lambda_1 T} + A_0 c_1 \\ c_2 e^{\lambda_2 2T} &= A_1 c_2 e^{\lambda_1 T} + A_0 c_2. \end{aligned}$$

The coefficients c get canceled (what is important) and the solution to this system is just what we want to prove. \triangleleft

One double root

Let the characteristic equation of (11.21) has one two-fold solution $\lambda = \lambda_1$.

Then the equivalent difference equation (11.21) has the coefficients

$$A_1 = 2e^{\lambda_1 T}, \quad A_0 = -e^{2\lambda_1 T} \quad (11.24)$$

Solution: The proof is formally the same as for the two distinct roots, only with basis elements $e^{\lambda_1 kT}$ and $kT e^{\lambda_1 kT}$.

The response to the continuous system is

$$y_k = c_1 e^{\lambda_1 t} + c_2 t e^{\lambda_1 t}$$

After expressing the sampled response and its shifted variants, substituting into (11.22) and comparing the terms at the individual basis items, we obtain the following set of equations

$$\begin{aligned} c_1 e^{2\lambda_1 T} + 2c_2 T e^{2\lambda_1 T} &= A_1 c_1 e^{\lambda_1 T} + A_1 T c_2 e^{\lambda_1 T} + A_0 c_1 \\ c_2 e^{2\lambda_1 T} &= A_1 c_2 e^{\lambda_1 T} + A_0 c_2 \end{aligned}$$

Again, the solution is just what we wanted to proof. \triangleleft

Two complex roots

Let the characteristic equation of (11.21) has two complex roots $\lambda_1 = \rho + \omega i$ and $\lambda_2 = \rho - \omega i$.

Then the equivalent difference equation (11.21) has the coefficients

$$A_1 = 2e^{\rho T} \cos(\omega T), \quad A_0 = -e^{2\rho T} \quad (11.25)$$

Solution: Again, the proof is formally the same as for the previous cases, only with the basis elements $e^{\rho kT} \sin(\omega kT)$ and $e^{\rho kT} \cos(\omega kT)$.

The response of the continuous system is

$$y_k = c_1 e^{\rho t} \cos(\omega t) + c_2 e^{\rho t} \sin(\omega t)$$

After expressing the sampled response and its shifted variants, substituting into (11.22) and comparing the terms at the individual basis items, we obtain the following set of equations

$$\begin{aligned} -c_1 e^{2\rho T} \sin(2\omega T) + c_2 e^{2\rho T} \cos(2\omega T) &= -A_1 c_1 e^{\rho T} \sin(\omega T) + A_1 c_2 e^{\rho T} \cos(\omega T) + A_0 c_2 \\ c_1 e^{2\rho T} \cos(2\omega T) + c_2 e^{2\rho T} \sin(2\omega T) &= A_1 c_1 e^{\rho T} \cos(\omega T) + A_1 c_2 e^{\rho T} \sin(\omega T) + A_0 c_1 \end{aligned}$$

Once more, the solution is just what we wanted to proof. \triangleleft