

Filtrace dat, Kalmanův filtr

Matematické metody pro ITS (11MAMY)

Jan Prikryl

10. přednáška 11MAMY

čtvrtek 16. dubna 2020

verze: 2020-04-19 08:41

Obsah

<i>Kalmanův filtr</i>	2
<i>Predikce</i>	5
<i>Filtrace</i>	6
<i>Kalmanův zisk</i>	7
<i>Implementace KF</i>	8
<i>Shrnutí</i>	9
<i>Trocha lineární algebry a statistiky</i>	9
<i>Kovariance transformované vícerozměrné náhodné veličiny</i>	9
<i>Rozdělení sdružené pravděpodobnosti dvou normálně rozdělených hodnot</i>	10

Tento text je do jisté míry experimentálním pískovištěm na odladění převodu textu prezentace vytvořené v L^AT_EXové třídě beamer do textu vysázeného pomocí tuft_e-handout. Obsah je oproti prezentaci mírně rozšířen o poznámky. Bude se ještě v průběhu semestru měnit, kontrolujte si prosím čas sestavení v záhlaví tohoto souboru.

V roce 1960 R. E. Kálmán publikoval svůj slavný článek popisující rekurzivní řešení problému lineárního filtrování diskretních dat¹. Od té doby byl Kalmanův filtr, z velké části díky pokrokům v oblasti digitálního zpracování dat, předmětem rozsáhlého výzkumu a četných aplikací, ze začátku zejména v oblasti autonomní nebo asistované navigace.

Kalmanův filtr je sada matematických rovnic, která poskytuje efektivní výpočetní (rekurzivní) nástroj pro odhad stavu procesu způsobem, minimalizujícím střední hodnotu kvadratické chyby. Filtr je velmi mocný v několika rovinách: podporuje odhady minulých, současných a dokonce budoucích stavů a může tak učinit i v situacích, kdy není známa přesná povaha modelovaného systému.

Předpokládejme nyní, že analyzujeme nějaký změřený digitální signál, například posloupnost opakovaných měření teploty, otáček, rychlosti, nebo zvuk, radarová nebo GPS měření nebo dokonce

Rudolf E. Kálmán (1930–2006) byl americký vědec maďarského původu. Jeho nejvýznamnějším dílem je filtrovací algoritmus pro optimální odstraňování šumu z měřených dat. Tento algoritmus byl poprvé masivně nasazen během programu Apollo, používá se v ponorkách, v bezpilotních leteckých prostředcích či střelách s plochou dráhou letu. Dnes má filtr obrovské množství průmyslových a domácích aplikací, včetně například GPS.

¹

digitalizovaný obrázek. A samozřejmě máte v okolním prostředí nějaké rušivé vlivy, šumy, které vaše měření tohoto signálu zkreslují a vnášejí do něj náhodné chyby.

Co můžete udělat, abyste tyto rušivé vlivy odstranili? Nejjednodušší věc, která mi přichází na mysl, a o níž jsme se již bavili v přednášce o měření dat, je: *použijme nějakou formu vyhlazování*, tedy (vážený) plovoucí průměr, exponenciální zapomínání nebo něco obdobného. Jak ale asi tušíte, tento jednoduchý přístup v reálném životě u většiny problémů samozřejmě nefunguje optimálně bez nějakého rozumného nastavení parametrů. Potřebujeme důmyslnější přístup: i pevné a v nějaký okamžik optimální přednastavení vyhlazovacích parametrů nezaručuje, že bude vždy dobře fungovat, a i když budeme předpokládat, že chování měřeného systému se v čase nemění (systém je tedy stacionární), mohou se v čase měnit poruchy, jež nám ovlivňují měřená data.

Uživatelé v oblasti digitálního zpracování signálu se zabývají tímto problémem po celá desetiletí a pro tento problém existuje mnoho technik. Kalmanův filtr je jednou z těchto technik. A velmi silnou: Kalmanův filtr totiž *hledá optimální průměrování pro každý diskrétní časový okamžik*. Díky tomu, že kromě historie měření používá i stavový model měřeného systému, má k dispozici informace o historii stavu tohoto systému. Ty pak používá k vyhodnocování šumu v měřených datech.

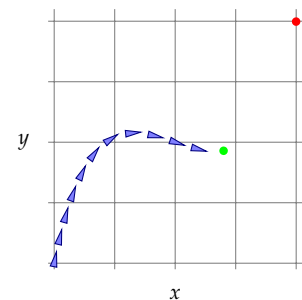
Kalmanův filtr

Začněme z tím, co Kalmanův filtr je: Jedná se o metodu předpovědi budoucího stavu systému založenou na předchozích stavech a měření výstupu systému. **■ je to observer, pozorovatel stavu, v případech, kdy stav nelze měřit přímo ■**

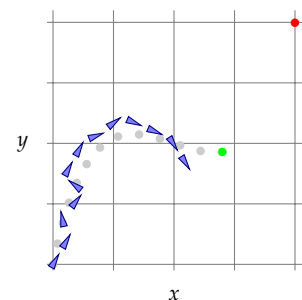
Chcete-li pochopit, co Kalmanův filtr dělá, podívejte se na obrázku vpravo. Z posloupnosti přesných poloh modrých šipek na obrázku 1 můžeme patrně i jednoduchým matematickým modelem (například extrapolací lineárního trendu z několika málo předchozích vzorků) předpovědět, že by měla následovat zelená tečka. Jak spolehlivě byste ovšem pomocí této metody dokázali předpovědět polohu červeného bodu vpravo nahoře? Pokud byste navíc neměli data přesně změřená, ale například ta, jež jsou zakreslena na obrázku 2, jak moc byste důvěřovali v předpověď budoucí polohy vozidla v zeleném bodě? A v červeném?

Z tohoto jednoduchého příkladu můžeme vysledovat tři důležité zásady:

- Nestací pouze předpovídat budoucí hodnoty – potřebujeme také vždy udávat úroveň důvěry v naši předpověď.
- Předpovídání daleko do budoucnosti je méně spolehlivé, než



Obrázek 1: Přesné měření pozice vozidla po jedné sekundě. Je celkem jednoduché předpovídat polohu vozidla o sekundu později (zelený bod).



Obrázek 2: Reálná měření pozice vozidla po jedné sekundě při slabším GPS signálu. Z měření není vůbec patrné, kde se bude vozidlo opravdu o sekundu později nacházet (zelený bod).

bližší předpovědi (toto je známý problém související s extrapolací známých trendů).

- Spolehlivost měření vstupních dat (šum měření) ovlivňuje spolehlivost vašich předpovědí.

Pokusme se nyní použít těchto zásad k tomu, abychom formulovali matematický model naší předpovědi předpokládaného korektního stavu systému. Pro jednoduchost zvolíme pouze *jednorozměrný lineární model*, například dynamický model vývoje rychlosti vozidla, jenž může sloužit jako jedna z komponent našeho oblíbeného a často diskutovaného tempomatu.

První komponenta, kterou dynamický model potřebuje, je **stav**. Stav je množina všech parametrů, jež potřebujeme k popisu současného chování systému a k provedení předpovědi budoucího chování. U našeho minimalistického příkladu použijeme dvě čísla: Aktuální rychlost² vozidla r a náš nejlepší odhad aktuální míry zrychlení (nazvěme jej a).³

Z předešlých přednášek si pamatujeme, že stav dynamického systému je obecně vektor, označovaný \mathbf{x} , jenž samozřejmě může zahrnovat i více parametrů než dva – pokud chceme modelovat složitější systémy, se dvěma parametry si těžko vystačíme. V našem případě bude ale stav systému v časovém kroku t popsán dvouprvkovým vektorem

$$\mathbf{x}_t = \begin{pmatrix} r_t \\ a_t \end{pmatrix}.$$

Další část, kterou potřebujeme, je matematický **model vývoje stavu**: Tento model popisuje, jak si myslíme, že se v čase stav systému mění, popisuje, jak si představujeme, že se systém chová. V běžném Kalmanově filtru je tento model vždy lineární a stacionární, modelované veličiny jsou lineární funkcí stavu. V našem jednoduchém případě uvažujeme primitivní model

$$\begin{aligned} r_{t+1} &= r_t + a_t \\ a_{t+1} &= a_t \end{aligned}$$

kde uvažujeme, rychlost je ovlivňována mírou zrychlení a že míra zrychlení se v čase nemění (zrychlení je konstantní, to platí pouze pro rovnoměrné změny pohybu – rovnoměrně zrychlený a rovnoměrně zpomalený pohyb, případně pohyb konstantní rychlostí, kde $a_t = 0$). V maticovém tvaru to lze zapsat ve formě nám již známé rovnice pro vývoj stavu diskrétního dynamického LTI systému

$$\mathbf{x}_{t+1} = \begin{pmatrix} r_{t+1} \\ a_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_t \\ a_t \end{pmatrix} \equiv \mathbf{A} \mathbf{x}_t. \quad (1)$$

Samozřejmě, že náš model není dokonalý⁴. Budeme předpokládat, že chyby, jichž se model dopouští, jsou normálně rozdělené se střední hodnotou nula a přidáme do stavu další člen, sloupcový

² Ano, měli bychom rychlost označovat v , ale z důvodů, které brzy vyplynou, použijeme jiné písmenko

³ Ono to není přesně zrychlení, protože nebudou sedět jednotky — jde o zrychlení naškálované časovým krokem.

⁴ Může se zdát, že kdyby náš model zcela odpovídal realitě, nepotřebovali bychom Kalmanův filtr, není to ale zcela pravda: Díky Kalmanově filtraci jsme schopni reagovat na nepřesná měření výstupů i jinak dokonalého modelu. Uvidíte to o kousek níže.

vektor **šumu procesu**⁵ \mathbf{v}_t , jenž nabývá náhodných hodnot z více-rozměrného rozdělení $\mathcal{N}(\mathbf{o}, \mathbf{Q})$ se střední hodnotou nula (danou nulovým sloupcovým vektorem \mathbf{o}) a kovarianční maticí⁶ \mathbf{Q} . Skutečné šumové hodnoty neznáme, předpokládáme ale, že dokážeme odhadnout, jak významný tento šum je. To nám dává aktualizovanou stavovou rovnici, která je

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{v}_t. \quad (2)$$

Touto rovnicí vlastně říkáme, že stav systému nemáme nikdy určen zcela přesně, je zatížen určitou chybou, jejíž očekávanou velikost udává kovarianční matice \mathbf{Q} .

Rovnicí (2), popisující model vývoje stavu, máme nyní popsané evoluční pravidlo našeho dynamického systému: ze známého stavu \mathbf{x}_t a chyby modelu \mathbf{v}_t odvodíme nový stav \mathbf{x}_{t+1} .

Poslední část, jenž nám chybí, je **model měřených veličin** jako funkce stavu (výstupní rovnice, v našem případě totiž měříme výstupy systému): Když naměříme nová data o výstupu modelovaného systému, budou se tato naměřená data jistě lišit od výstupních dat, která udává matematický model tohoto systému (jenž v tomto případě navíc předpokládá, že systém je stacionární a lineární). V takovém případě by se pak měly stavové parametry modelu mírně změnit tak, aby se zpřesnil náš současný model a budoucí předpovědi (kromě stavu je v modelu všechno stacionární a nepředpokládáme, že by se to nějak měnilo). Důležité je uvědomit si, že nemusíme (a často ani nemůžeme) měřit přesně stejné parametry, jako jsou ty, které jsou součástí stavu.⁷

V našem ilustračním příkladě měříme pouze rychlost (vertikální polohu nových bodů na grafu), nikoliv zrychlení (směrnici přímký mezi body). Výstupem modelu, který můžeme ověřit měřením, je tedy pouze

$$r_t = \begin{pmatrix} 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_t \\ a_t \end{pmatrix} = \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{x}_t$$

V obecnějším případě můžeme měřit více, než jednu veličinu, takže měření je vektor – označme jej \mathbf{z} (tento vektor je podmnožinou výstupů systému, vektoru \mathbf{y}). Samotná měření jsou také nepřesná, obecná rovnice měření má proto tvar

$$\mathbf{z}_t = \mathbf{C}\mathbf{x}_t + \mathbf{w}_t \quad (3)$$

kde vektor \mathbf{w}_t reprezentuje **šum měření** (opět normálně rozdělený s nulovou střední hodnotou a s kovarianční maticí, kterou tentokrát označíme \mathbf{R} , je tedy $\mathbf{w}_t \sim \mathcal{N}(\mathbf{o}, \mathbf{R})$) a \mathbf{C} je obecně matice, mapující stavové proměnné na měřené výstupy systému: její počet sloupců odpovídá počtu stavových proměnných a počet řádků odpovídá počtu měřených proměnných.

Nyní, když jsme sestavili model systému, můžeme nyní začít s fází predikce budoucího stavu — srdcem Kalmanova filtru.

⁵ Zde je ten důvod odchylky od standardní notace rychlosti: pletla by se nám s běžně užívaným označením pro šum procesu.

⁶ V námi modelovaném případě je \mathbf{Q} symetrická matice 2×2 , udávající jednak variance $\text{var}(r) = \sigma_r^2$ a $\text{var}(a) = \sigma_a^2$ a kovarianci $\text{cov}(r, a) = \sigma_{ra}^2$, tedy

$$\mathbf{Q} = \begin{pmatrix} \sigma_r^2 & \sigma_{ra}^2 \\ \sigma_{ra}^2 & \sigma_a^2 \end{pmatrix}$$

⁷ Například Kalmanův filtr, popisující pohyb auta, může chtít předpovídat zrychlení a , rychlost r a polohu (x, y) vozu, automobil je ale vybaven pouze senzory, měřícími úhel natočení volantu θ a úhlovou rychlost otáčení kol $(\omega_1, \omega_2, \omega_3, \omega_4)$.

Predikce

Začněme předpokladem, že náš model je zcela dokonalý, bez šumu procesu, čili $\mathbf{v}_t = \mathbf{o}$. Jak budeme předpovídat budoucí stav zařízení v kroku $t + 1$? Jednoduše: Je to jen jedno vyčíslední stavové rovnice,

$$\hat{\mathbf{x}}_{t+1} = \mathbf{A}\mathbf{x}_t,$$

kde $\hat{\mathbf{x}}$ označuje **statistický odhad** přesné, ale neznámé hodnoty \mathbf{x} .

Co očekáváme v kroku t za výstup za předpokladu, že měříme zcela přesně (tedy $\mathbf{w}_t = \mathbf{o}$)? Opět je to jednoduché: předpověď výstupu $\hat{\mathbf{z}}_t$ je dána předpovědí stavu $\hat{\mathbf{x}}_t$ použitou v rovnici měření s nulovým šumem, tedy

$$\hat{\mathbf{z}}_t = \mathbf{C}\hat{\mathbf{x}}_t. \quad (4)$$

Co ale opravdu měříme? Je velmi pravděpodobné, že měření neodpovídají zcela tomu, co udává model, mezi modelovanou hodnotou výstupu $\hat{\mathbf{z}}_t$ a její změřenou hodnotou \mathbf{z}_t bude tedy nejspíše nenulový rozdíl

$$\mathbf{e}_t = \mathbf{z}_t - \hat{\mathbf{z}}_t.$$

Hodnota \mathbf{e}_t (nazývaná také **inovace**) ukazuje, jak moc špatný je náš současný odhad měřených veličin (tedy výstup modelu) – kdyby vše bylo perfektní, byla by inovace rovna nule. Abychom tento fakt zohlednili, přidáme inovaci do původní rovnice vývoje stavu a vynásobíme ji maticovým faktorem, jenž nám říká, jak by se měl modelem odhadnutý stav $\hat{\mathbf{x}}_t$ na základě tohoto rozdílu mezi očekávaným a skutečným měřením optimálně změnit na korigovanou („filtrovanou“) hodnotu $\hat{\mathbf{x}}_t^*$:

$$\hat{\mathbf{x}}_t^* = \hat{\mathbf{x}}_t + \mathbf{K}_t\mathbf{e}_t. \quad (5)$$

Časově proměnná matice \mathbf{K}_t je známá jako **Kalmanův zisk**, a jak uvidíme, funguje zprostředkovaně jako škálovací faktor mezi předpovědí výstupu modelu a reálně změřenými hodnotami:⁸ Lze to jasně vidět v okamžiku, kdy použijeme rovnici (5) ke stanovení korigované hodnoty výstupu (tj. nejpravděpodobnější změřené hodnoty $\hat{\mathbf{z}}_t^*$):

$$\mathbf{C}\hat{\mathbf{x}}_t^* = \mathbf{C}\hat{\mathbf{x}}_t + \mathbf{C}\mathbf{K}_t\mathbf{e}_t$$

a pokud označíme $\mathbf{C}\mathbf{K}_t = \mathbf{W}_t$, využijeme vztahu (4) a s ním spojené analogie $\hat{\mathbf{z}}_t^* = \mathbf{C}\hat{\mathbf{x}}_t^*$, dostáváme velmi zajímavý výsledek

$$\begin{aligned} \hat{\mathbf{z}}_t^* &= \hat{\mathbf{z}}_t + \mathbf{W}_t(\mathbf{z}_t - \hat{\mathbf{z}}_t) \\ &= (1 - \mathbf{W}_t)\hat{\mathbf{z}}_t + \mathbf{W}_t\mathbf{z}_t. \end{aligned} \quad (6)$$

Dalo by se říci, že Kalmanův filtr odstraňuje nepřesnosti⁹ v modelovaných stavech tak, že způsobem podobným exponenciálnímu zapomínání průměruje data poskytnutá modelem systému a data, opravdu na výstupech zařízení změřená, přičemž váha změřených dat je dána maticí \mathbf{W}_t . Rozdíl mezi plovoucím průměrem respektive exponenciálním zapomínáním a Kalmanovým filtrem je v tom,

⁸ Vzpomeňte si na diskusi okolo exponenciálního zapomínání, o němž jsme se bavili v minulé přednášce.

⁹ Modelované jako šum s normálním rozdělením

že váha, použitá při průměrování obou složek $\hat{\mathbf{z}}_t$ a \mathbf{z}_t , není v čase konstantní — lze ukázat, že v případě Kalmanova filtru je váha \mathbf{K}_t respektive \mathbf{W}_t nastavována tak, aby vždy minimalizovala chybu aproximace výsledného stavu obdobně, jako metoda nejmenších čtverců v lineární regresi.

Předpověď stavu v Kalmanově filtru je tedy lineární kombinací předpovědi měření a reálně změřených hodnot. Dříve, než se dostaneme k vzorci pro výpočet \mathbf{K}_t , měli bychom popřemýšlet o tom, jak by se Kalmanův zisk měl chovat:

- Pokud je šum měření \mathbf{w}_t velký, možný nesoulad mezi \mathbf{z}_t a $\hat{\mathbf{z}}_t$ je pravděpodobně pouze artefakt tohoto šumu a ne opravdová inovace. Pokud je tedy šum měření velký, \mathbf{K}_t by mělo být malé, protože pak bude malé i \mathbf{W}_t .
- Pokud je šum procesu \mathbf{v}_t velký, tj. pokud očekáváme, že se stav může rychle měnit, měli bychom brát vliv inovace vážněji, protože je pravděpodobné, že se vnitřní stav zařízení skutečně změnil a náš model vývoje stavu tuto změnu nepostihl. Pokud je tedy šum procesu velký, \mathbf{K}_t by mělo být také velké, protože pak bude velké i \mathbf{W}_t .

Spojíme-li tyto dvě úvahy dohromady, dospějeme k závěru, že

$$\mathbf{K}_t \sim \frac{\text{šum procesu}}{\text{šum měření}}.$$

Filtrace

■ **chybí úvod** ■ Jeden způsob, jak vyhodnotit nejistotu (v našem podání šum) u hodnoty je podívat se na její odchylky od očekávané hodnoty, jež nám v jednorozměrném případě udává rozptyl, obecně pro vícerozměrné případy pak kovarianční matice pravděpodobnostního rozdělení chyb. První kovariance, ovlivňující chování našeho stavového modelu, je kovariance předpovědi budoucího stavu

$$\mathbf{P}_{t+1} = \text{cov}(\mathbf{x}_{t+1})$$

přičemž můžeme s pomocí rovnice vývoje stavu (1) odvodit i \mathbf{P}_{t+1} z její předchozí hodnoty:¹⁰

$$\mathbf{P}_{t+1} = \text{cov}(\hat{\mathbf{x}}_{t+1}) = \text{cov}(\mathbf{A}\mathbf{x}_t) = \mathbf{A} \text{cov}(\mathbf{x}_t) \mathbf{A}^T = \mathbf{A}\mathbf{P}_t\mathbf{A}^T$$

Předpokládáme zatím, že náš model pozorovaného procesu je dokonalý a není v něm, co bychom nemohli předvídat. Již jsem si ale řekli, že v reálné situaci ovšem existuje mnoho neznámých vlivů, které náš model nepostihuje a přesto ovlivňují vývoj stavu v řase (pro náš model vozidla jistě zanedbáváme vítr, tření, sklon komunikace, ...). Tyto neznámé vlivy zahrnujeme do modelu v rovnici (2) jako kovarianční matici \mathbf{Q} šumu procesu \mathbf{v}_t . Kovariance predikce

¹⁰ Pokud tápete, kde se to vzalo, koukněte na vztah (12) v příloze.

stavu se tak zvýší na

$$\mathbf{P}_{t+1} = \mathbf{A}\mathbf{P}_t\mathbf{A}^T + \mathbf{Q}.$$

Stav systému je tak v tento okamžik dán jako vícerozměrná náhodná veličina s normálním rozdělením o střední hodnotě $\hat{\mathbf{x}}_{t+1}$ a s kovariancí \mathbf{P}_{t+1} .

Posledním zdrojem šumu v našem systému je nepřesné měření. Podle stejné logiky získáme kovarianční matici pro $\hat{\mathbf{z}}_{t+1}$:

$$\mathbf{S}_{t+1} = \text{cov}(\hat{\mathbf{z}}_{t+1}) = \text{cov}(\mathbf{C}\hat{\mathbf{x}}_{t+1}) = \mathbf{C} \text{cov}(\hat{\mathbf{x}}_{t+1})\mathbf{C}^T = \mathbf{C}\mathbf{P}_{t+1}\mathbf{C}^T.$$

Stejně jako u předchozího kroku si pamatujeme, že naše měřené hodnoty \mathbf{z}_t jsou zatíženy normálně rozdělenými nepřesnostmi \mathbf{w}_t . Když jsme definovali rovnici pro výstup (3), řekli jsme si, že kovarianční matice, která popisuje šum těchto hodnot, značíme \mathbf{R} . Přidáme ji tedy k původní hodnotě kovarianční matice měření a obdržíme

$$\mathbf{S}_{t+1} = \mathbf{C}\mathbf{P}_{t+1}\mathbf{C}^T + \mathbf{R}. \quad (7)$$

Kalmanův zisk

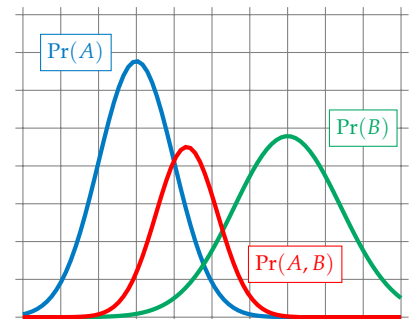
Tím jsme si připravili téměř vše, co je potřeba k výpočtu Kalmanova zisku \mathbf{K}_t . Způsobů odvození vztahu pro \mathbf{K}_t je mnoho, podle mého názoru je nejjednodušší ten, jenž vychází z úvahy, jak optimálně kombinovat změřenou a modelovanou hodnotu výstupních proměnných v případě, kdy o nich uvažujeme jako o náhodných proměnných.

Tou optimální kombinací je výpočet sdružené pravděpodobnosti dvou normálně rozdělených hodnot náhodných veličin, popisujících opravdu měřené výstupy \mathbf{z}_t a modelem předpokládané výstupy $\hat{\mathbf{z}}_t$. Zapsáno pomocí normálních rozdělení máme

$$\begin{aligned} \mathbf{z}_t &\approx \mathcal{N}(\mathbf{z}_t, \mathbf{R}) && \text{měřené výstupy} \\ \hat{\mathbf{z}}_t &\approx \mathcal{N}(\mathbf{C}\hat{\mathbf{x}}_t, \mathbf{C}\mathbf{P}_t\mathbf{C}^T) && \text{modelované výstupy.} \end{aligned}$$

Výstupní hodnoty považujeme za náhodné proměnné a můžeme tedy pro každou s nich stanovit pravděpodobnost, že nabývá určité hodnoty. Pro jakoukoliv možnou hodnotu výstupu ζ máme k dispozici dvě pravděpodobnostní hodnoty: (i) pravděpodobnost, že hodnotu ζ změříme na výstupu systému, $\Pr(\mathbf{z}_t = \zeta)$ a (ii) pravděpodobnost, že tuto hodnotu obdržíme jako výstup modelu, $\Pr(\hat{\mathbf{z}}_t = \zeta)$.

Nás ale zajímá nejpravděpodobnější hodnota výstupu ζ^* , která odpovídá pravděpodobnostnímu rozdělení modelovaných výstupů $\hat{\mathbf{z}}_t$ a měřených výstupů \mathbf{z}_t – tedy opravdová hodnota výstupu za situace, kdy změříme \mathbf{z}_t a na výstupu našeho modelu předpokládáme $\hat{\mathbf{z}}_t$. Tato hodnota je dána sdruženou pravděpodobností obou jevů. Pro jednorozměrný případ je to znázorněno na obrázku 3.



Obrázek 3: Sdružená hustota pravděpodobnosti dvou nezávislých jevů A a B , $\Pr(A, B) = \Pr(A)\Pr(B)$.

Pro sdruženou pravděpodobnost dvou nezávislých náhodných veličin platí

$$\Pr(A, B) = \Pr(A) \cdot \Pr(B)$$

a v našem případě tedy

$$\Pr(\mathbf{z}_t = \zeta, \hat{\mathbf{z}}_t = \zeta) = \Pr(\mathbf{z}_t = \zeta) \cdot \Pr(\hat{\mathbf{z}}_t = \zeta).$$

V rovnici (6) jsme si zavedli novou náhodnou veličinu $\hat{\mathbf{z}}_t^*$, určující nejpravděpodobnější hodnotu měřené veličiny jako optimální lineární kombinaci modelovaného výstupu \mathbf{z}_t a změřeného výstupu $\hat{\mathbf{z}}_t$. Pro hustotu pravděpodobnosti $\hat{\mathbf{z}}_t^*$ z výše uvedeného vyplývá

■ **patrně chyba ve škálovacím faktoru** ■

$$\hat{\mathbf{z}}_t^* = \mathbf{C}\hat{\mathbf{x}}_t^* \approx \mathcal{N}(\mathbf{C}\hat{\mathbf{x}}_t^*, \mathbf{C}\mathbf{P}_t^*\mathbf{C}^T) = \mathcal{N}(\mathbf{z}_t, \mathbf{R})\mathcal{N}(\mathbf{C}\hat{\mathbf{x}}_t, \mathbf{C}\mathbf{P}_t\mathbf{C}^T).$$

S využitím výsledků z přílohy, jmenovitě vztahu (14) dostaneme

$$\begin{aligned} \mathbf{C}\hat{\mathbf{x}}_t^* &= \mathbf{C}\hat{\mathbf{x}}_t + \tilde{\mathbf{K}}(\mathbf{z}_t - \mathbf{C}\hat{\mathbf{x}}_t) \\ \mathbf{C}\mathbf{P}_t^*\mathbf{C}^T &= \mathbf{C}\mathbf{P}_t\mathbf{C}^T - \tilde{\mathbf{K}}\mathbf{C}\mathbf{P}_t\mathbf{C}^T \end{aligned} \quad (8)$$

a podle rovnice (13) a dosazením z (7) bude škálovací faktor

$$\tilde{\mathbf{K}} = \mathbf{C}\mathbf{P}_t\mathbf{C}^T(\mathbf{C}\mathbf{P}_t\mathbf{C}^T + \mathbf{R})^{-1} = \mathbf{C}\mathbf{P}_t\mathbf{C}^T\mathbf{S}_t^{-1}. \quad (9)$$

Pohledem na rovnice (8) a (9) zjistíme, že se všechny členy násobí zleva maticí \mathbf{C} (jedno násobení je skryté ve vztahu pro $\tilde{\mathbf{K}}$) a v druhé rovnici i zprava maticí \mathbf{C}^T . Můžeme proto vše trochu zjednodušit:

$$\begin{aligned} \hat{\mathbf{x}}_t^* &= \hat{\mathbf{x}}_t + \mathbf{K}(\mathbf{z}_t - \mathbf{C}\hat{\mathbf{x}}_t) \\ \mathbf{P}_t^* &= \mathbf{P}_t - \mathbf{K}\mathbf{C}\mathbf{P}_t \end{aligned} \quad (10)$$

kde Kalmanův zisk je modifikovaný původní škálovací faktor $\tilde{\mathbf{K}}$

$$\mathbf{K}_t = \mathbf{P}_t\mathbf{C}^T(\mathbf{C}\mathbf{P}_t\mathbf{C}^T + \mathbf{R})^{-1} = \mathbf{P}_t\mathbf{C}^T\mathbf{S}_t^{-1}. \quad (11)$$

Implementace KF

Postupně vykonáváme následující kroky

1. Sestavení dynamického LTI stavového modelu – reprezentujeme jej maticemi \mathbf{A} , \mathbf{B} a \mathbf{C} , předpokládáme tvar

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \\ \mathbf{z}_t &= \mathbf{C}\mathbf{x}_t \quad (\text{pouze měřené výstupy}) \end{aligned}$$

2. Nastavení parametrů (počáteční podmínky $\hat{\mathbf{x}}_0^*$, počáteční šum procesu \mathbf{P}_0^* , kovariance \mathbf{Q} a \mathbf{R})
3. Iterační část většinou začíná predikčním krokem a opakuje dokola kroky

- (a) Predikce (angl. *time update*) – použijeme nejpravděpodobnější původní stav $\hat{\mathbf{x}}_t^*$, vstup \mathbf{u}_t a kovarianční matici šumu procesu \mathbf{P}_t k předpovědi $\hat{\mathbf{x}}_{t+1}$ a \mathbf{P}_{t+1} a měřených hodnot výstupu systému $\hat{\mathbf{z}}_{t+1}$.
- (b) Filtrace (angl. *measurement update*) – vypočteme aktuální Kalmanův zisk \mathbf{K}_{t+1} , použijeme měření \mathbf{z}_{t+1} k výpočtu inovace a korekci stavu systému $\hat{\mathbf{x}}_{t+1}^*$; spočteme aktualizovanou kovarianční matici šumu procesu \mathbf{P}_{t+1}^* , pokud to potřebujeme, určíme $\hat{\mathbf{z}}_{t+1}^*$.

nebo alternativně (mění se význam počátečních podmínek)

- (a) Filtrace
(b) Predikce

Shrnutí

Kalmanův filtr hledá optimální průměrování pro každý diskretní časový okamžik. Díky modelu systému má uloženy informace o historii stavu.

Není to signálový filtr, je to **estimátor**. Lze ukázat, že se jedná o *rekurzivní Bayesovský estimátor*, odhadující hustotu pravděpodobnosti náhodné veličiny (stavu) rekurzivně na základě historie informací o měření. V případě, kdy rozložení pravděpodobností není Gaussovské, lze Kalmanův filtr nahradit pokročilejšími metodami, jako je sekvenční Monte Carlo odhadování.

Neodhaduje chyby měření ani nepřesnosti modelu, oboje je dáno předem ve formě kovariančních matic \mathbf{Q} a \mathbf{R} .

Považován za jeden z nejvýznamnějších objevů 20. století.

Máme k dispozici implementace pro většinu běžných programovacích jazyků.

Původně odvozen pro lineární (nestacionární) systémy s normálně rozděleným šumem, existují i varianty pro nelineární systémy (EKF, UKF, DD1). Ty jsou však výrazně složitější a pomalejší.

Trocha lineární algebry a statistiky

Kovariance transformované vícerozměrné náhodné veličiny

Připomínám z definice kovariance, že pro sloupcový vektor \mathbf{x} je

$$\text{cov}(\mathbf{x}) = \mathbf{E}[\mathbf{x}\mathbf{x}^T] - \mathbf{E}[\mathbf{x}]\mathbf{E}[\mathbf{x}^T]$$

a z maticového počtu si připomeňme, že pro transpozici součinu matic platí

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T.$$

Kovarianci transformovaného sloupcového vektoru \mathbf{Ax} lze potom s využitím výše uvedených vztahů vyjádřit jako

$$\begin{aligned}
 \text{cov}(\mathbf{Ax}) &= \mathbb{E}[\mathbf{Ax}(\mathbf{Ax})^T] - \mathbb{E}[\mathbf{Ax}] \mathbb{E}[(\mathbf{Ax})^T] \\
 &= \mathbb{E}[\mathbf{Axx}^T \mathbf{A}^T] - \mathbb{E}[\mathbf{Ax}] \mathbb{E}[\mathbf{x}^T \mathbf{A}^T] \\
 &= \mathbf{A} \mathbb{E}[\mathbf{xx}^T] \mathbf{A}^T - \mathbf{A} \mathbb{E}[\mathbf{x}] \mathbb{E}[\mathbf{x}^T] \mathbf{A}^T \\
 &= \mathbf{A} \left(\mathbb{E}[\mathbf{xx}^T] - \mathbb{E}[\mathbf{x}] \mathbb{E}[\mathbf{x}^T] \right) \mathbf{A}^T \\
 &= \mathbf{A} \text{cov}(\mathbf{x}) \mathbf{A}^T.
 \end{aligned} \tag{12}$$

Rozdělení sdružené pravděpodobnosti dvou normálně rozdělených hodnot

Pokud máme dvě nezávislé a normálně rozdělené jednorozměrné náhodné veličiny

$$\begin{aligned}
 x &\approx \mathcal{N}(\mu_0, \sigma_0^2) \\
 y &\approx \mathcal{N}(\mu_1, \sigma_1^2)
 \end{aligned}$$

jaká bude střední hodnota a rozptyl rozdělení jejich sdružené pravděpodobnosti, tedy násobku $\mathcal{N}(\mu_0, \sigma_0^2) \mathcal{N}(\mu_1, \sigma_1^2)$?

Máme

$$\begin{aligned}
 \mathcal{N}(x, \mu_0, \sigma_0^2) &= \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}} \\
 \mathcal{N}(x, \mu_1, \sigma_1^2) &= \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}
 \end{aligned}$$

a proto

$$\begin{aligned}
 \mathcal{N}(x, \mu_0, \sigma_0^2) \mathcal{N}(x, \mu_1, \sigma_1^2) &= \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}} \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} = \\
 &= \frac{1}{\sqrt{4\pi^2\sigma_0^2\sigma_1^2}} e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2} - \frac{(x-\mu_1)^2}{2\sigma_1^2}} = \frac{1}{2\pi\sigma_0\sigma_1} e^{-\beta(x, \mu_0, \mu_1, \sigma_0, \sigma_1)}
 \end{aligned}$$

Pokud má být výsledek opět variantou normálního rozdělení, musí být argument exponenciály, β , opět ve tvaru $(x - \mu)^2 / (2\sigma^2)$, což lze provést, byť je to za cenu nepříliš vizuálně potěšitelného výsledku:

$$\begin{aligned}
 \beta(x, \mu_0, \mu_1, \sigma_0, \sigma_1) &= \frac{(x - \mu_0)^2}{2\sigma_0^2} + \frac{(x - \mu_1)^2}{2\sigma_1^2} = \\
 &= \frac{\sigma_1^2(x - \mu_0)^2 + \sigma_0^2(x - \mu_1)^2}{2\sigma_0^2\sigma_1^2} = \\
 &= \frac{\sigma_1^2(x^2 - 2\mu_0x + \mu_0^2) + \sigma_0^2(x^2 - 2\mu_1x + \mu_1^2)}{2\sigma_0^2\sigma_1^2} = \\
 &= \frac{(\sigma_0^2 + \sigma_1^2)x^2 - 2(\mu_0\sigma_1^2 + \mu_1\sigma_0^2)x + \mu_0^2\sigma_1^2 + \mu_1^2\sigma_0^2}{2\sigma_0^2\sigma_1^2} =
 \end{aligned}$$

(koeficient u x^2 musí být jednička, proto dělíme vše $(\sigma_0^2 + \sigma_1^2)$):

$$\begin{aligned}
 &= \frac{x^2 - 2\frac{\mu_0\sigma_1^2 + \mu_1\sigma_0^2}{\sigma_0^2 + \sigma_1^2}x + \frac{\mu_0^2\sigma_1^2 + \mu_1^2\sigma_0^2}{\sigma_0^2 + \sigma_1^2}}{2\frac{\sigma_0^2\sigma_1^2}{\sigma_0^2 + \sigma_1^2}} = \\
 &= \frac{x^2 - 2\mu x + \xi + (\mu^2 - \mu^2)}{2\sigma^2} = \frac{x^2 - 2\mu x + \mu^2 + \xi - \mu^2}{2\sigma^2} = \\
 &= \frac{(x^2 - \mu)^2}{2\sigma^2} + \frac{\xi - \mu^2}{2\sigma^2} = \frac{(x^2 - \mu)^2}{2\sigma^2} + \frac{(\mu_0 - \mu_1)^2}{2(\sigma_0^2 + \sigma_1^2)}
 \end{aligned}$$

kde druhý, červeně označený člen, je součást škálovací konstanty a střední hodnotu ani rozptyl sdružené pravděpodobnosti neovlivní.

Pro střední hodnotu a rozptyl dostaneme

$$\begin{aligned}
 \mu &= \frac{\mu_0\sigma_1^2 + \mu_1\sigma_0^2}{\sigma_0^2 + \sigma_1^2} = \frac{\mu_0\sigma_0^2 + \mu_0\sigma_1^2 + \mu_1\sigma_0^2 - \mu_0\sigma_0^2}{\sigma_0^2 + \sigma_1^2} \\
 &= \frac{\mu_0(\sigma_0^2 + \sigma_1^2)}{\sigma_0^2 + \sigma_1^2} + \frac{\sigma_0^2(\mu_1 - \mu_0)}{\sigma_0^2 + \sigma_1^2} \\
 &= \mu_0 + \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2}(\mu_1 - \mu_0) \\
 \sigma^2 &= \frac{\sigma_0^2\sigma_1^2}{\sigma_0^2 + \sigma_1^2} = \frac{\sigma_0^2\sigma_0^2 + \sigma_0^2\sigma_1^2 - \sigma_0^2\sigma_0^2}{\sigma_0^2 + \sigma_1^2} \\
 &= \frac{\sigma_0^2(\sigma_0^2 + \sigma_1^2)}{\sigma_0^2 + \sigma_1^2} - \frac{\sigma_0^2\sigma_0^2}{\sigma_0^2 + \sigma_1^2} \\
 &= \sigma_0^2 - \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2}\sigma_0^2
 \end{aligned}$$

a bližším pohledem zjistíme, že obě rovnice mají společný faktor $k = \sigma_0^2 / (\sigma_0^2 + \sigma_1^2)$, takže

$$\begin{aligned}
 \mu &= \mu_0 + k(\mu_1 - \mu_0) \\
 \sigma^2 &= \sigma_0^2 - k\sigma_0^2
 \end{aligned}$$

Stejně to funguje pro vícerozměrná rozdělení s kovariančními maticemi, kde

$$\tilde{\mathbf{K}} = \mathbf{\Sigma}_0(\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1)^{-1}, \quad (13)$$

takže

$$\begin{aligned}
 \boldsymbol{\mu} &= \boldsymbol{\mu}_0 + \tilde{\mathbf{K}}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \\
 \boldsymbol{\Sigma} &= \boldsymbol{\Sigma}_0 - \tilde{\mathbf{K}}\boldsymbol{\Sigma}_0
 \end{aligned} \quad (14)$$