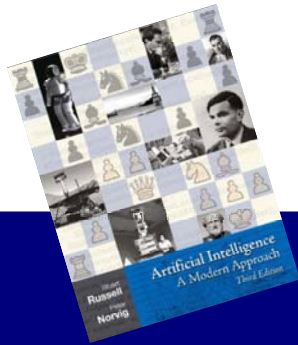


Umělá inteligence II

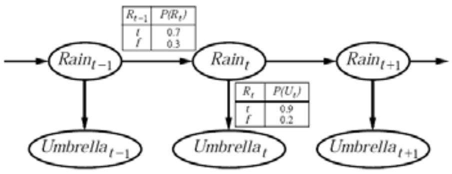
Roman Barták, KTIML

roman.bartak@mff.cuni.cz
http://ktiml.mff.cuni.cz/~bartak

5

Na úvod

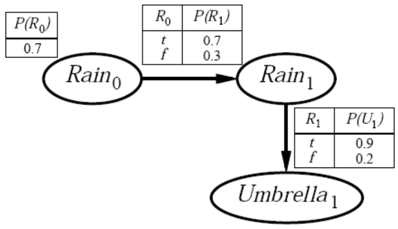
- Umíme **reprezentovat neurčitost v čase**
  - přechodový $\mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{t-1})$ a senzorický model $\mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_t)$
  - Markovské předpoklady
- Typické **řešené úlohy**
  - filtrace: $\mathbf{P}(\mathbf{X}_t \mid \mathbf{e}_{1:t})$
  - predikce: $\mathbf{P}(\mathbf{X}_{t+k} \mid \mathbf{e}_{1:t})$ pro k>0
  - vyhlazování: $\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t})$ pro k: $0 \leq k < t$
  - nejpravděpodobnější průchod: $\text{argmax}_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t} \mid \mathbf{e}_{1:t})$



- **Skryté Markovské modely** (HMM)
  - speciální případ s jedinou náhodnou a senzorickou proměnnou
  - efektivní řešení úloh pomocí maticových operací

Dynamické Bayesovské sítě

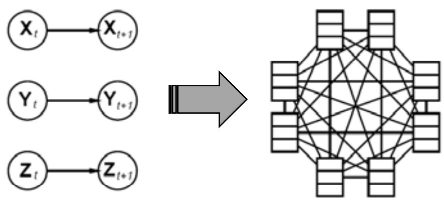
- **Dynamická Bayesovská síť** (DBN) reprezentuje temporální pravděpodobnostní model
  - skládá se z opakujících se vrstev proměnných
    - stačí tedy popsat první vrstvu
    - apriorní distribuci $\mathbf{P}(\mathbf{X}_0)$
    - přechodový model $\mathbf{P}(\mathbf{X}_1 \mid \mathbf{X}_0)$
    - senzorický model $\mathbf{P}(\mathbf{E}_1 \mid \mathbf{X}_1)$



  - dle Markovských předpokladů má každá proměnná rodiče buď ve stejné nebo předchozí vrstvě

DBN vs. HMM

- skrytý Markovský model je speciálním případem dynamické Bayesovské sítě
- podobně lze (diskrétní) Bayesovskou síť kódovat do skrytého Markovského modelu
  - jedna proměnná jejíž hodnoty jsou n-tice hodnot původních n proměnných

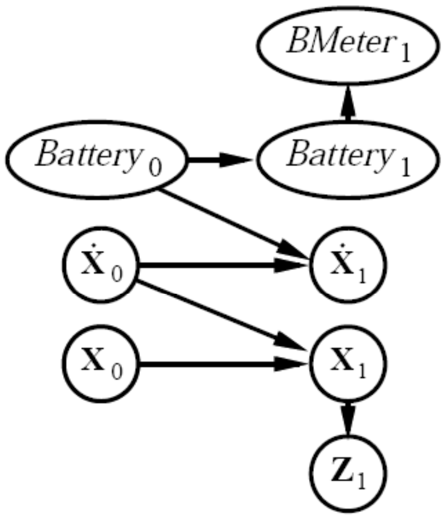


Jaký je tedy rozdíl?

- stejný jako rozdíl Bayesovské sítě a úplné sdružené distribuce
- DBN s 20 Booleovskými náhodnými proměnnými, každá má tři rodiče
  - pro přechodový model potřebujeme $20 \times 2^3 = 160$ pravděpodobností
- skrytý Markovský model s jedinou náhodnou proměnnou s 2^{20} stavy
  - pro přechodový model potřebujeme $2^{20} \times 2^{20} \approx 10^{12}$ pravděpodobností
  - tj. více paměti a pomalejší odvozování u HMM

Modelový příklad

- Uvažujme příklad složitější DBN popisující chování baterií poháněného robota

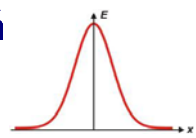


  - potřebujeme zachytit **polohu** robota $\mathbf{X}_t = (X_t, Y_t)$, jeho **rychlost** $\dot{\mathbf{X}}_t = (\dot{X}_t, \dot{Y}_t)$ a **stav baterie**
    - další poloha záleží na rychlosti a původní poloze
    - další rychlost záleží na předchozí rychlosti a stavu baterie
  - pozorovat můžeme polohu například prostřednictvím **GPS** Z_t
  - podobně můžeme pozorovat stav baterie pomocí **senzoru** $BMeter_t$

Vadné senzory

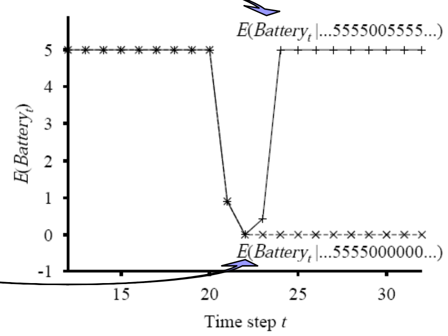
- **Přesný senzor** má v senzorickém modelu na diagonále 1 a jinde 0.

  Takové senzory ale nejsou časté. Přesnější model používá mírnou odchylku podobně jako u Gaussova rozložení – **Gaussův chybový model**



- **Reálné senzory** ale **chybují** a někdy vykazují úplně jinou hodnotu (např. ukazatel obsahu nádrže auta může při náklonu ukazovat prázdnou nádrž).

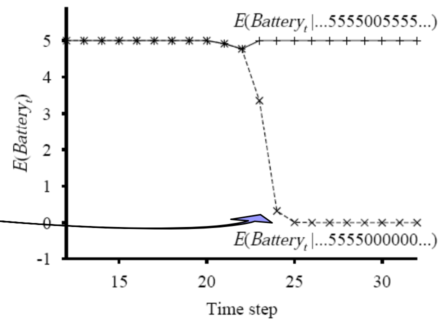
  Potom se Gaussův chybový model chová jinak než bychom chtěli.
  - po dvou chybných zprávách máme téměř jistotu, že baterie je prázdná
  - pokud potom přijde korektní měření, model najednou hlásí plnou baterii (**dočasná porucha**)
    - mezitím jsme ale na základě údaje o prázdné baterii mohli podniknout nevhodné kroky
  - pokud dále chodí chybné údaje, senzor si je „jistý", že baterie je prázdná, i když pravděpodobnost takové skokové změny je velmi malá (**trvalá porucha**)



Vadné senzory

oprava dočasné poruchy

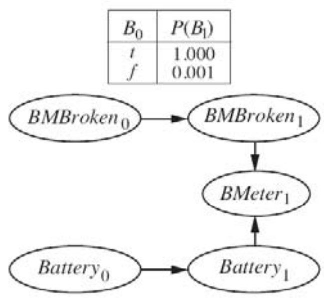
- Nejjednodušší úprava modelu pro zachycení poruchy spočívá v přidání nenulové pravděpodobnosti zcela mylného měření.
  $P(BMeter_t=0 \mid Battery_t=5) = 0.03$
- Hovoříme o **modelu dočasných poruch**.
- Tento model přidá do uvažování jisté zpoždění / **setrvačnost**, takže robot dojde k závěru o prázdné baterii později.
- Nakonec si ale stejně myslí, že baterie je prázdná, takže to neřeší problém s trvale vadným senzorem!

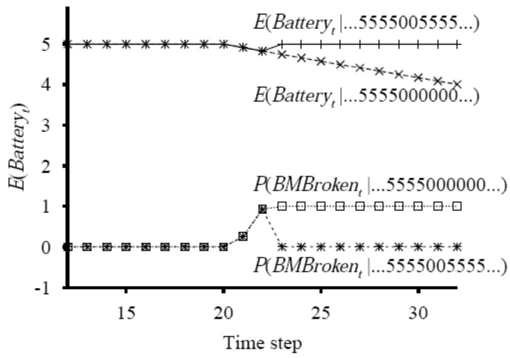


Vadné senzory

oprava trvalé poruchy

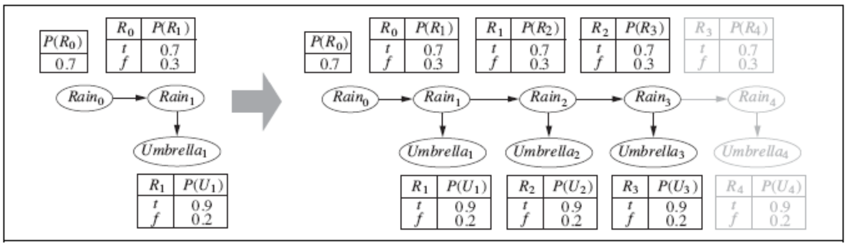
- Pro **model trvalých poruch** použijeme novou skrytou proměnnou BMBroken indikující stav senzoru.
- Pokud je senzor OK, chová se přechodový model stejně.
- Pokud dojde k trvalé poruše (kombinace možnosti poruchy a delší dobu přicházejících „podivných“ údajů ze senzoru), přepne se proměnná BMBroken trvale na 1 (chybný senzor) a zprávy o stavu baterie se řídí „normálním“ průběhem vybíjení.





Odvozování v DBN
exaktní metody

- DBN je v podstatě **Bayesovská síť**, takže pro odvozování můžeme použít stejné techniky.
- Přesněji, nejprve je potřeba DBN **rozvinout** podle získaných pozorování

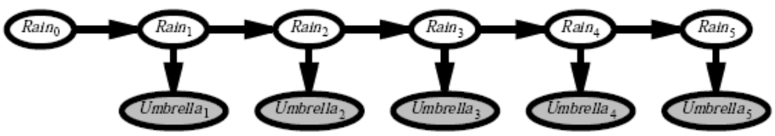


a potom můžeme použít známé metody odvozování.

- Tento **naivní přístup** má ale zásadní nevýhodu v tom, že čas odvozování se z každým dalším pozorováním prodlužuje.
- Můžeme použít **inkrementální metody**, kdy stačí v paměti udržovat dva časové řezy.
  - podobné jako **eliminace proměnných**
  - špatná zpráva je, že paměť pro pamatování si posledního výsledku je exponenciální v počtu proměnných ve vrstvě $O(d^{n+k})$, což se projeví i v času inference (n-počet proměnných, d-počet hodnot, k-počet rodičů)

Odvozování v DBN
aproximační metody

- Můžeme zkusit aproximační metody, z nichž **vážení věrohodností** se lépe adaptuje na DBN.
- Náhodné proměnné budeme vzorkovat od času 0 po čas posledního pozorování.
  - každý vzorek musí projít celou sítí
  - můžeme vzít N vzorků a projít s nimi síť najednou (podobné jako propagace zprávy u exaktních metod)



- Je tady ale **problém**!
  - **vzorky jsou generovány nezávisle na pozorování!**
  - to obvykle vede k malé váze vzorků a potřebě generovat mnohem větší počet vzorků
    - potřebný **počet vzorků roste exponenciálně s časem**
    - pokud používáme konstantní počet vzorků a vzorky jen inkrementálně rozšiřujeme, klesá s časem přesnost metody

Odvozování v DBN

částicové filtrování

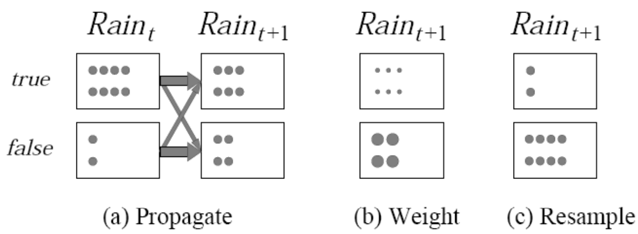
- Při generování vzorků se potřebujeme držet v oblastech s velkou pravděpodobností stavů.
- **Částicové filtrování (particle filtering)** množinu vzorků (částic) po každém kroku převzorkuje.
  - začneme s N vzorky **vzorkovanými** podle $\mathbf{P}(\mathbf{X}_0)$
  - každý vzorek **propagujeme** vpřed výběrem hodnoty $\mathbf{x}_{t+1}$ podle $\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t)$
  - vzorek je **zvážen** podle pozorování $\mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{x}_{t+1})$
  - populace vzorků je **převzorkována** podle vah
    - vybereme N vzorků, kde pravděpodobnost výběru vzorku je dána jeho váhou (nové vzorky nemají váhu)

Odvozování v DBN

algoritmus částicového filtrování

function PARTICLE-FILTERING(e, N, dbn) returns a set of samples for the next time step
 inputs: e, the new incoming evidence
  N, the number of samples to be maintained
  dbn, a DBN with prior $\mathbf{P}(\mathbf{X}_0)$, transition model $\mathbf{P}(\mathbf{X}_1|\mathbf{X}_0)$, and sensor model $\mathbf{P}(\mathbf{E}_1|\mathbf{X}_1)$
 static: S, a vector of samples of size N, initially generated from $\mathbf{P}(\mathbf{X}_0)$
 local variables: W, a vector of weights of size N

 for i = 1 to N **do**
  $S[i] \leftarrow$ sample from $\mathbf{P}(\mathbf{X}_1|\mathbf{X}_0 = S[i])$
  $W[i] \leftarrow \mathbf{P}(\mathbf{e}|\mathbf{X}_1 = S[i])$
 $S \leftarrow$ WEIGHTED-SAMPLE-WITH-REPLACEMENT(N, S, W)
 return S



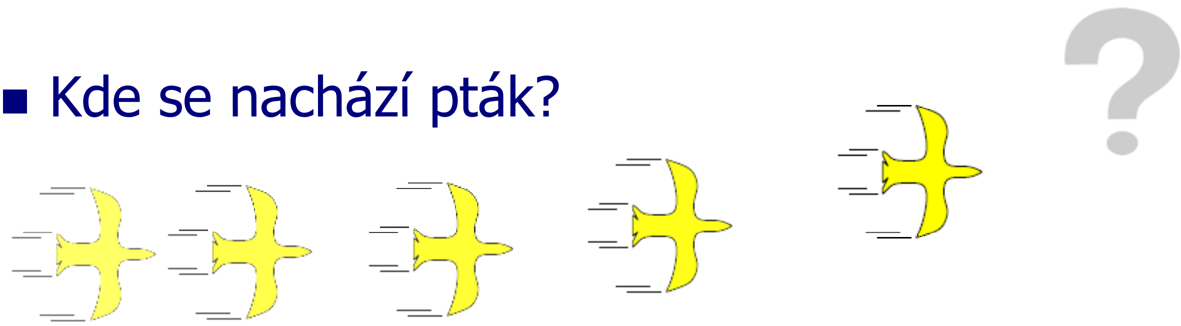
Odvozování v DBN

korektnost částicového filtrování

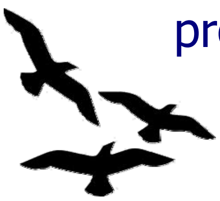
- Předpokládejme, že vzorky v kroku t jsou konzistentní s pravděpodobnostní distribucí
  $N(\mathbf{x}_t \mid \mathbf{e}_{1:t}) / N = P(\mathbf{x}_t \mid \mathbf{e}_{1:t})$
- Po **propagaci** do času t+1 dostaneme
  $N(\mathbf{x}_{t+1} \mid \mathbf{e}_{1:t}) = \Sigma_{\mathbf{x}_t} P(\mathbf{x}_{t+1} \mid \mathbf{x}_t)\, N(\mathbf{x}_t \mid \mathbf{e}_{1:t})$
- Celková **váha** všech vzorků pro stav $\mathbf{x}_{t+1}$ je
  $W(\mathbf{x}_{t+1} \mid \mathbf{e}_{1:t+1}) = P(\mathbf{e}_{t+1} \mid \mathbf{x}_{t+1})\, N(\mathbf{x}_{t+1} \mid \mathbf{e}_{1:t})$
- Po **převzorkování** dostaneme
  $N(\mathbf{x}_{t+1} \mid \mathbf{e}_{1:t+1}) / N = \alpha\, W(\mathbf{x}_{t+1} \mid \mathbf{e}_{1:t+1})$
  $= \alpha\, P(\mathbf{e}_{t+1} \mid \mathbf{x}_{t+1})\, N(\mathbf{x}_{t+1} \mid \mathbf{e}_{1:t})$
  $= \alpha\, P(\mathbf{e}_{t+1} \mid \mathbf{x}_{t+1})\, \Sigma_{\mathbf{x}_t} P(\mathbf{x}_{t+1} \mid \mathbf{x}_t)\, N(\mathbf{x}_t \mid \mathbf{e}_{1:t})$
  $= \alpha`\, P(\mathbf{e}_{t+1} \mid \mathbf{x}_{t+1})\, \Sigma_{\mathbf{x}_t} P(\mathbf{x}_{t+1} \mid \mathbf{x}_t)\, P(\mathbf{x}_t \mid \mathbf{e}_{1:t})$
  $= P(\mathbf{x}_{t+1} \mid \mathbf{e}_{1:t+1})$

Motivační příklad

- Kde se nachází pták?

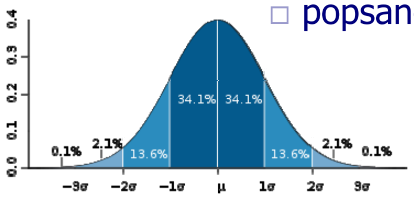


- Jakou techniku použijeme pro řešení?
  - filtraci (zjišťujeme aktuální polohu na základě dosavadních pozorování)
  - potřebujeme ale pracovat se spojitými náhodnými proměnnými (poloha a rychlost letu)



Spojité proměnné

- Dosud jsme pracovali pouze s diskrétními proměnnými, kde se pravděpodobnostní distribuce popisovala tabulkou.
- **Jak pracovat se spojitými proměnnými?**
  - **diskretizace**
    - množina hodnot se rozdělí do disjunktních intervalů
    - často vede ke ztrátě přesnosti a velkým tabulkám
  - přímé metody použitím **standardních pravděpodobnostních rozdělení**
    - Gaussovo (normální) rozdělení
      - často používané rozdělení pravděpodobnosti spojité proměnné
      - popsané střední hodnotou μ a rozptylem σ^2



$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)},$$

Spojité proměnné

podmíněné pravděpodobnosti

- Jak popisovat podmíněné pravděpodobnosti v Bayesovské síti?
  - pro závislost **spojité proměnné na spojité proměnné** se často používá lineární Gaussovo rozdělení
    - střední hodnota závisí lineárně na parametrech rodiče
    - rozptyl se nemění

$$\begin{aligned} & P(Cost = c \mid Harvest = h, Subsidy? = true) \\ & = N(a_t h + b_t, \sigma_t)(c) \\ & = \frac{1}{\sigma_t\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{c-(a_t h + b_t)}{\sigma_t}\right)^2\right) \end{aligned}$$

  - pro závislost **spojité proměnné na diskrétní proměnné**
    - enumerace (pro jednotlivé hodnoty diskrétní proměnné se definují samostatná rozdělení)
  - pro závislost **diskrétní proměnné na spojité proměnné**
    - měkký práh

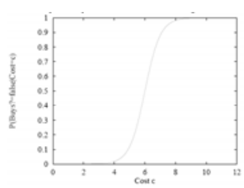
probit (probability unit) rozdělení

$$\Phi(x) = \int_{-\infty}^{x} N(0,1)(x)dx$$
$$P(Buys? = true \mid Cost = c) = \Phi((-c+\mu)/\sigma)$$

logit (logistic function) rozdělení

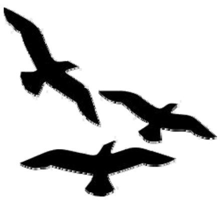
$$P(Buys? = true \mid Cost = c) = \frac{1}{1 + exp(-2\frac{-c+\mu}{\sigma})}$$

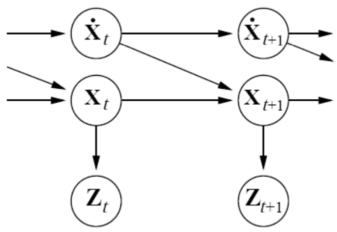
- probit rozdělení lépe vystihuje reálné situace (je to jako pevný práh, jehož poloha je dána s Gaussovskou chybou)
- logit rozdělení je „protáhlejší“ (long tail), snáze se počítá



Kalmanův filtr

- Vraťme se k příkladu s letem ptáka, který popíšeme pomocí dynamické Bayesovské sítě.
  - náhodné proměnné jsou poloha a rychlost ptáka
    - další poloha záleží na předchozí poloze a rychlosti formou lineárního Gaussova rozdělení
      $P(X_{t+\Delta}=x_{t+\Delta} \mid X_t=x_t \,,\, \dot{X}_t=\dot{x}_t) = N(x_t+\dot{x}_t\Delta,\sigma^2)\ (x_{t+\Delta})$
  - pozorujeme polohu ptáka Z_t
    - opět můžeme předpokládat použití Gaussova rozdělení





Kalmanův filtr
vlastnosti

- Použití Gaussova rozdělení dává zajímavé vlastnosti pro filtraci, predikci a vyhlazování.
- Je-li současné rozdělení $\mathbf{P}(\mathbf{X}_t \mid \mathbf{e}_{1:t})$ Gaussovo a přechodový model $\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t)$ je lineárně Gaussovský, potom $\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t})$ je také Gaussovo rozdělení.

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) = \int_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t) P(\mathbf{x}_t|\mathbf{e}_{1:t})\, d\mathbf{x}_t$$

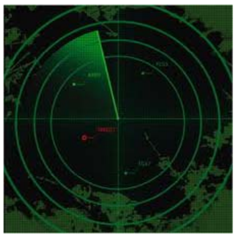
- Je-li $\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t})$ Gaussovo rozdělení a senzorický model $\mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1})$ je lineárně Gaussovský, potom $\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$ je také Gaussovo rozdělení.

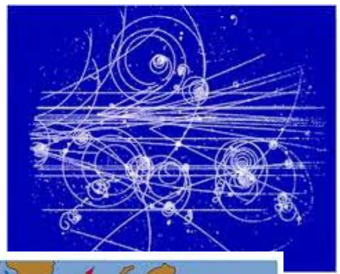
$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t})$$

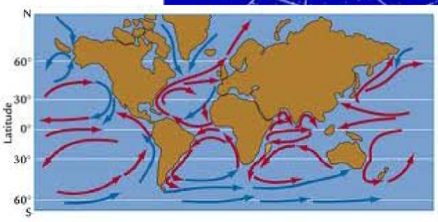
- Můžeme proto používat techniku posílání zpráv

Kalmanův filtr
aplikace

- Klasickou aplikací Kalmanova filtru je **sledování** letadel a raket na radaru.
- Používá se také na **rekonstrukci drah** částic ve fyzice elementárních částic případně na sledování mořských proudů.
- Aplikace ale sahají i mimo oblast sledování pohybu, obecně se hodí pro libovolný systém popsaný spojitými náhodnými proměnnými a měření se šumem (chemické provozy, ekosystémy, ekonomiky ...).



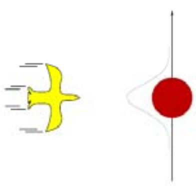




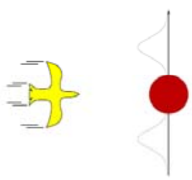
Kalmanův filtr
problémy

- Kalmanův filtr funguje pro modely s lineárními přechody a Gaussovskou chybou.
- **Co když se ale objekt nechová zcela lineárně?**
  příklad: ve směru letu ptáka je strom

Kalmanův filtr

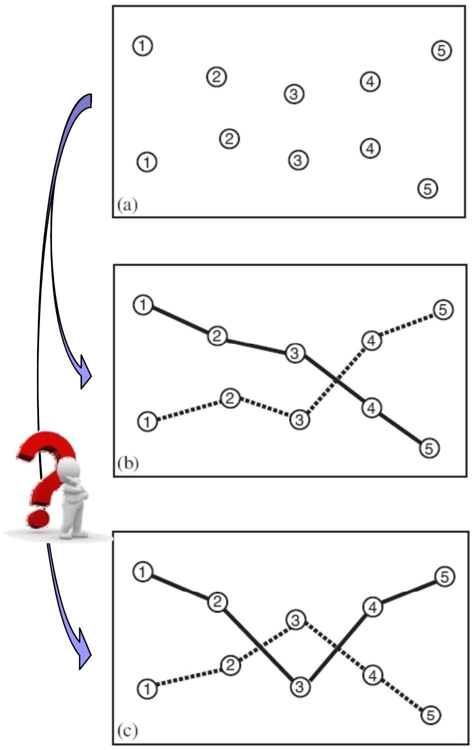


realističtější model



- □ Možné řešení: **přepínací (switching) Kalmanův filtr**
  - uvažuje se paralelně několik Kalmanových filtrů pro různé situace (přímý let, manévr vlevo, manévr vpravo)
  - vezme se vážený součet předpovědí, kde váha popisuje, jak dobře situace odpovídá datům
  - je to vlastně **rozšíření DBN o novou náhodnou proměnnou**

Sledování více objektů



- Zatím jsme mlčky předpokládali, že sledujeme jeden objekt, ale v praxi často sledujeme **více objektů najednou** (např. radary).

- **Kde je problém?**
  - Nevíme, které pozorování patří jakému objektu!

Sledování více objektů

používané metody

Problém asociace dat:
nevíme, které pozorování patří jakému objektu

- Při exaktním uvažování tedy musíme **vyščítat** pravděpodobnosti přes všechna **možná přiřazení**
  - pro jeden časový okamžik a n objektů je to n! možností
  - pro T časových okamžiků máme $(n!)^T$ možností
- Používají se spíše **aproximační metody**
  - v každém kroku se např. vybere nejlepší přiřazení objektů k pozorováním
    - **metoda nejbližšího souseda** (spáruje se pozorování s objektem, který má nejbližší předpovězenou polohu)
    - přesnější metoda je vzít přiřazení, které **maximalizuje sdruženou pravděpodobnost** pozorování při daných předpovězených polohách
  - výběr špatného přiřazení v jednom okamžiku znehodnocuje budoucí přiřazení
    - **částicové filtrování** (udržuje větší množinu možných přiřazení)
    - **MCMC** (prochází prostor možných přiřazení)

Sledování více objektů
reálná praxe

Situace v praxi je ještě komplikovanější.

- **falešné alarmy** (pozorování neodpovídá žádnému objektu)
- **chyba detekce** (existující objekt není pozorován)
- **nové a mizející objekty** (objevují se nové objekty a jiné objekty mizí)