بسم الله الرحمن الرحیم

هوش مصنوعی پیشرفته

فصل ۱۵

استدلال احتمالاتی در امتداد زمان

Probabilistic Reasoning over Time

کاظم فولادی
دانشکده مهندسی برق و کامپیوتر
دانشگاه تهران

http://courses.fouladi.ir/aai

هوش مصنوعی

استدلال احتمالاتی در امتداد زمان

۱

زمان
و
عدم
اطمینان

زمان و عدم اطمینان

Time and Uncertainty

دنیا **تغییر** می‌کند

$\Downarrow$

نیاز داریم آن را دنبال و پیش‌بینی کنیم.

ایده‌ی اصلی:	متغیرهای حالت و شاهد را برای هر گام زمانی کپی می‌کنیم

مجموعه‌ی متغیرهای حالت مشاهده‌ناپذیر در زمان t

$\mathbf{X}_t$ = set of unobservable state variables at time t

e.g., $BloodSugar_t$, $StomachContents_t$, etc.

مجموعه‌ی متغیرهای شاهد مشاهده‌پذیر در زمان t

$\mathbf{E}_t$ = set of observable evidence variables at time t

e.g., $MeasuredBloodSugar_t$, $PulseRate_t$, $FoodEaten_t$

با فرض **زمان گسسته**؛ «اندازه‌ی گام» وابسته به مسئله است.

Notation: $\mathbf{X}_{a:b} = \mathbf{X}_a, \mathbf{X}_{a+1}, \ldots, \mathbf{X}_{b-1}, \mathbf{X}_b$

فرآیندهای مارکوف (زنجیره‌های مارکوف)

ایجاد یک شبکه‌ی بیزی از متغیرهای دارای زمان

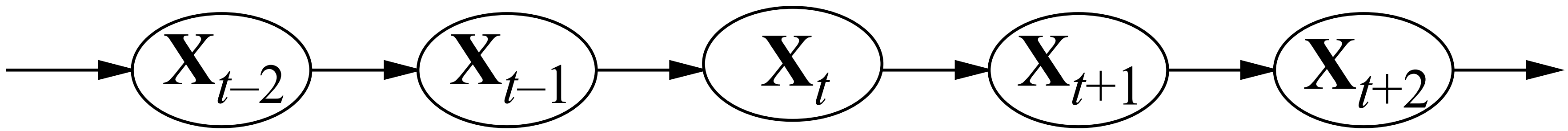
پرسش: والد هر گره چگونه باید انتخاب شود؟

پاسخ: استفاده از فرض مارکوف

فرض مارکوف *Markov Assumption*	$\mathbf{X}_t$ به زیرمجموعه‌ای کران‌دار از $\mathbf{X}_{0:t-1}$ وابسته است.

فرآیند مارکوف مرتبه اول

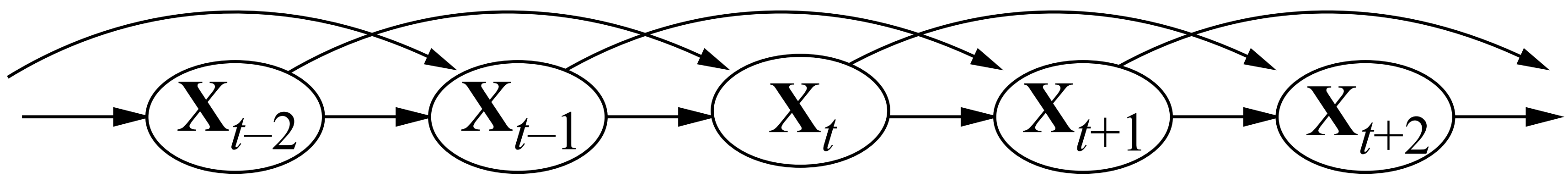
First-order Markov process: $\mathbf{P}(\mathbf{X}_t | \mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t | \mathbf{X}_{t-1})$



وابستگی حالت فعلی فقط به یک حالت قبلی

فرآیند مارکوف مرتبه دوم

Second-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-2}, \mathbf{X}_{t-1})$



وابستگی حالت فعلی فقط به دو حالت قبلی

فرض مارکوف حسگری

Sensor Markov assumption: $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{0:t}, \mathbf{E}_{0:t-1}) = \mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$

وابستگی مشاهده‌ی فعلی فقط به حالت فعلی

فرآیند ایستان

Stationary Process

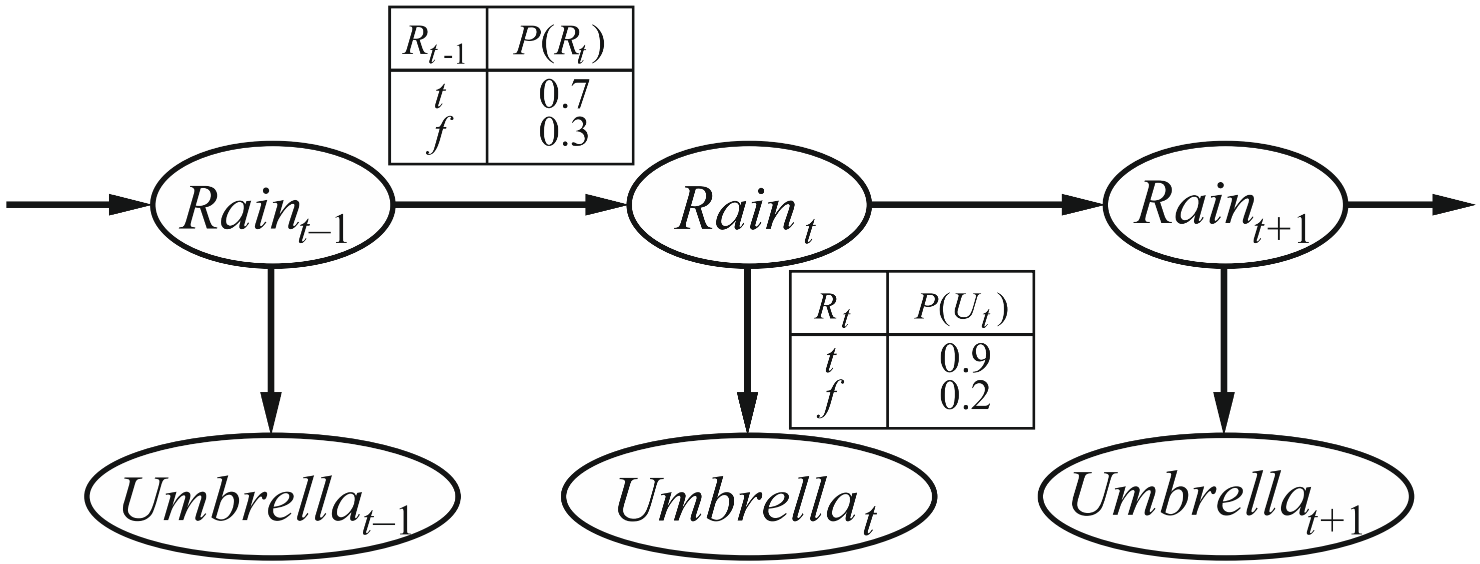
Stationary process: transition model $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$ and sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$ fixed for all t

فرآیند ایستان: مدل گذر و مدل حسگر برای همه‌ی زمان‌ها ثابت است.

فرآیند مارکوف مرتبه اول

مثال: دنیای چتر

Umbrella World



فرآیند مارکوف مرتبه اول

مشکلات

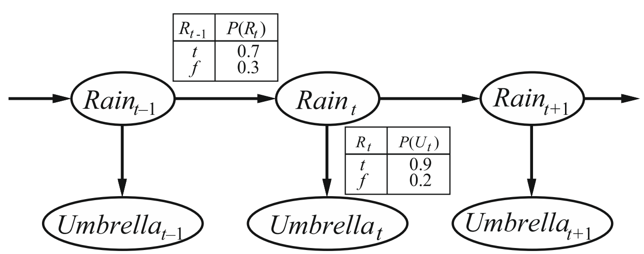
فرض مارکوف مرتبه اول در دنیای واقعی دقیقاً درست نیست!

راه‌حل‌های ممکن:

۱) **افزایش مرتبه‌ی** فرآیند مارکوف

۲) **گسترش حالت (Augment)**: افزودن به پارامترهای حالت

مثلاً: اضافه کردن $Temp_t$ ، $Pressure_t$ به دنیای چتر



مثلاً: اضافه کردن $Battery_t$ به مجموعه‌ی موقعیت و سرعت در مسئله‌ی حرکت ربات

هوش مصنوعی

استدلال احتمالاتی در امتداد زمان

۲

استنتاج در مدل‌های زمانی

وظیفه‌های استنتاج در امتداد زمان

INFERENCE TASKS OVER TIME

فیلتر کردن
Filtering

حالت باور ـــ ورودی به فرآیند تصمیم یک عامل رسیونال

Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$

پیش‌بینی
Prediction

ارزیابی دنباله‌های کنش ممکن (مشابه فیلتر کردن بدون شاهد)

Prediction: $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$ for $k > 0$

هموارسازی
Smoothing

بهتر کردن تخمین حالت‌های قبلی (ضروری برای یادگیری)

Smoothing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \leq k < t$

محتمل‌ترین تبیین
Most Likely Explanation

کاربرد در بازشناسی گفتار، کدگشایی با یک کانال نویزی

Most likely explanation: $\arg\max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$

فیلتر کردن

FILTERING

فیلتر کردن
Filtering

حالت باور (دنبال کردن حالت باور) ــــ ورودی به فرآیند تصمیم یک عامل رسیونال

$$\text{Filtering: } \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$$

هدف: طراحی یک الگوریتم **بازگشتی** برای تخمین حالت

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

$$\begin{aligned}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) &= \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}, \mathbf{e}_{t+1}) \\ &= \alpha \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}, \mathbf{e}_{1:t})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) \\ &= \alpha \underbrace{\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})}_{\text{تخمین}}\underbrace{\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t})}_{\text{پیش‌بینی}}\end{aligned}$$

فیلترینگ، نظارت (monitoring) هم نامیده می‌شود (هدف: دنبال کردن حالت باور).

فیلتر کردن

بخش پیش‌بینی

FILTERING

فیلتر کردن
Filtering

حالت باور (دنبال کردن حالت باور) ـــ ورودی به فرآیند تصمیم یک عامل رسیونال

Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$

پیش‌بینی با مجموع‌گیری روی $\mathbf{X}_t$

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

$\mathbf{f}_{1:t+1} = \textsc{Forward}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1})$ where $\mathbf{f}_{1:t} = \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$

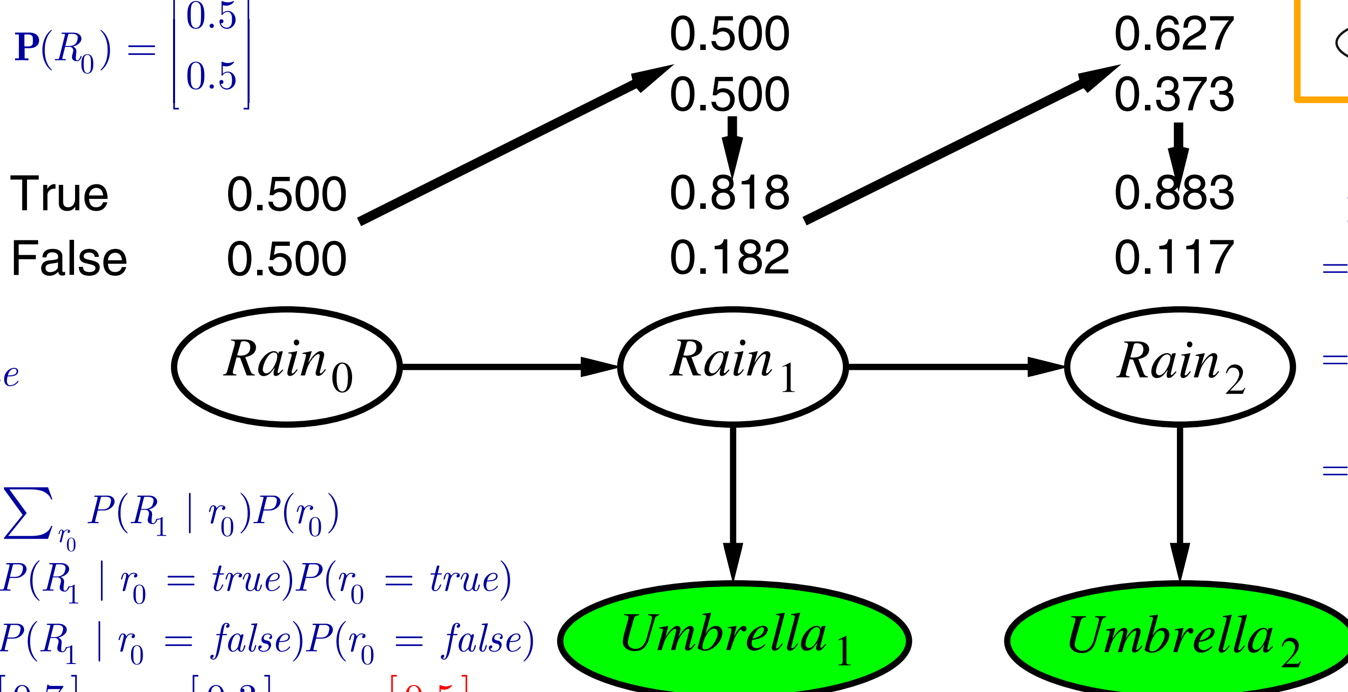
زمان و فضای لازم **ثابت** است (مستقل از t)

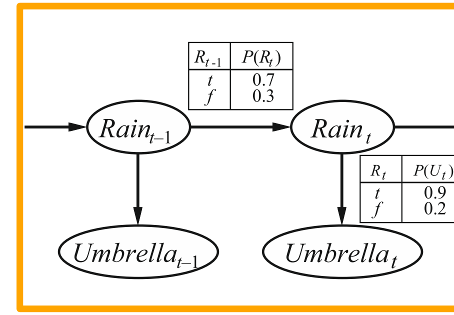
فیلتر کردن

مثال

FILTERING

$$\mathbf{P}(R_0) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$





$U_1 = true$

$$\begin{aligned}\mathbf{P}(R_1) &= \sum_{r_0} P(R_1 \mid r_0)P(r_0) \\ &= P(R_1 \mid r_0 = true)P(r_0 = true) \\ &+ P(R_1 \mid r_0 = false)P(r_0 = false) \\ &= \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} 0.5 + \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} 0.5 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}\end{aligned}$$

$$\begin{aligned}\mathbf{P}(R_1 \mid u_1) &= \alpha \mathbf{P}(u_1 \mid R_1)\mathbf{P}(R_1) \\ &= \alpha \begin{bmatrix} 0.9 \\ 0.2 \end{bmatrix} \cdot \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \\ &= \alpha \begin{bmatrix} 0.45 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.818 \\ 0.182 \end{bmatrix}\end{aligned}$$

$$\begin{aligned}\mathbf{P}(R_2 \mid u_1) &= \sum_{r_1} \mathbf{P}(R_2 \mid r_1)P(r_1 \mid u_1) \\ &= \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} 0.818 + \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} 0.182 \\ &= \begin{bmatrix} 0.627 \\ 0.373 \end{bmatrix}\end{aligned}$$

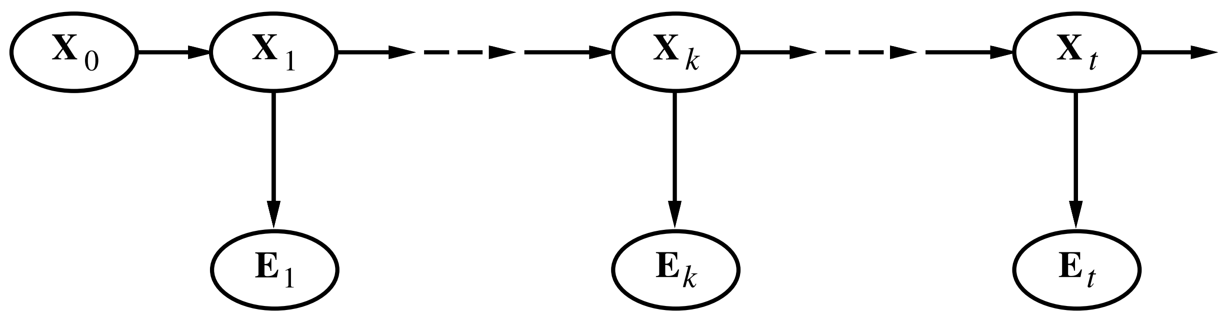
$$\begin{aligned}\mathbf{P}(R_2 \mid u_1, u_2) &= \alpha \mathbf{P}(u_2 \mid R_2)\mathbf{P}(R_2 \mid u_1) \\ &= \alpha \begin{bmatrix} 0.9 \\ 0.2 \end{bmatrix} \cdot \begin{bmatrix} 0.627 \\ 0.373 \end{bmatrix} \\ &= \alpha \begin{bmatrix} 0.565 \\ 0.075 \end{bmatrix} = \begin{bmatrix} 0.883 \\ 0.117 \end{bmatrix}\end{aligned}$$

SMOOTHING

بهتر کردن تخمین حالت‌های قبلی (ضروری برای یادگیری)

هموارسازی
Smoothing

Smoothing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \leq k < t$



Divide evidence $\mathbf{e}_{1:t}$ into $\mathbf{e}_{1:k}$, $\mathbf{e}_{k+1:t}$:

$$\begin{aligned}
\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t}) &= \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k}, \mathbf{e}_{k+1:t}) \\
&= \alpha \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{e}_{1:k}) \\
&= \alpha \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) \\
&= \alpha \mathbf{f}_{1:k}\mathbf{b}_{k+1:t}
\end{aligned}$$

هموارسازی

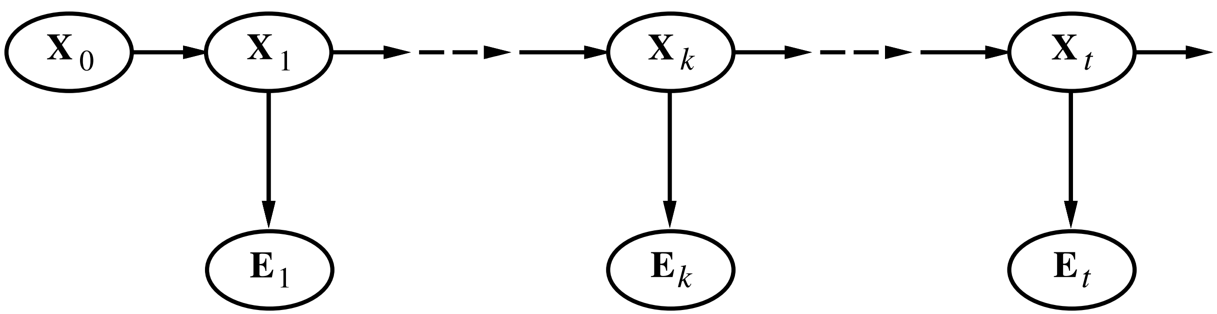
بازگشت پس‌رو برای محاسبه‌ی پیام‌های پس‌رو

SMOOTHING

هموارسازی
Smoothing

بهتر کردن تخمین حالت‌های قبلی (ضروری برای یادگیری)

Smoothing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \le k < t$



پیام‌های پس‌رو به‌وسیله‌ی یک بازگشت پس‌رو محاسبه می‌شوند:

$$\begin{aligned}
\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) &= \textstyle\sum_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\
&= \textstyle\sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\
&= \textstyle\sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)
\end{aligned}$$

$\boxed{\mathbf{b}_{k+1:t}}$

هموارسازی

الگوریتم پیش‌رو- پس‌رو

FORWARD–BACKWARD ALGORITHM

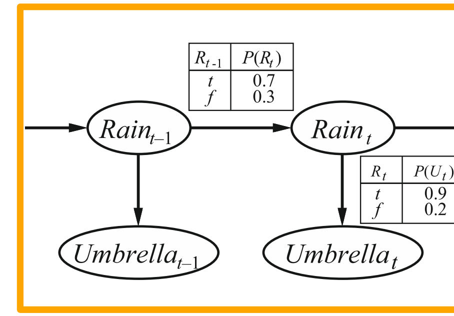
الگوریتم پیش‌رو- پس‌رو پیام‌های پیش‌رو را در طول مسیر می‌گیرد.

زمان: خطی بر حسب t (استدلال چنددرختی)
فضا: $O(t|\mathbf{f}|)$

هموارسازی

مثال

SMOOTHING



$$\mathbf{P}(R_1 \mid u_1, u_2) = \alpha \mathbf{P}(R_1 \mid u_1)\mathbf{P}(u_2 \mid R_1)$$

$$\mathbf{P}(R_1 \mid u_1) = \begin{bmatrix} 0.818 \\ 0.182 \end{bmatrix}$$

(forward filtering)

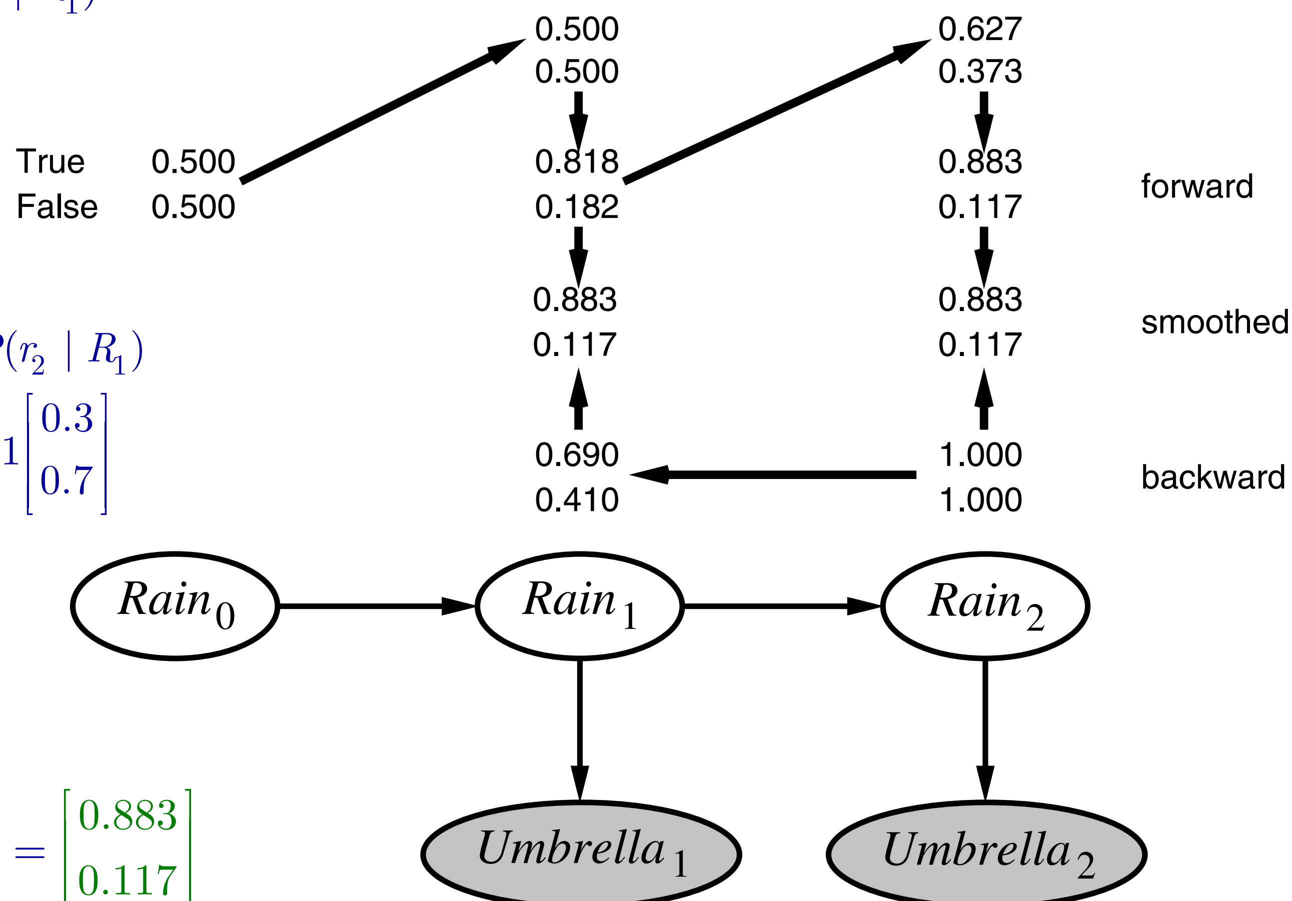
$$\mathbf{P}(u_2 \mid R_1) = \sum_{r_2} \mathbf{P}(u_2 \mid r_2) P(r_2) \mathbf{P}(r_2 \mid R_1)$$

$$= 0.9 \times 1 \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} + 0.2 \times 1 \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix}$$

$$= \begin{bmatrix} 0.69 \\ 0.41 \end{bmatrix}$$

(backward messages)

$$\mathbf{P}(R_1 \mid u_1, u_2) = \alpha \begin{bmatrix} 0.818 \\ 0.182 \end{bmatrix} \cdot \begin{bmatrix} 0.69 \\ 0.41 \end{bmatrix} = \begin{bmatrix} 0.883 \\ 0.117 \end{bmatrix}$$



محتمل‌ترین تبیین

Most Likely Explanation

محتمل‌ترین تبیین
Most Likely Explanation

کاربرد در بازشناسی گفتار، کدگشایی با یک کانال نویزی

Most likely explanation: $\arg\max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$

محتمل‌ترین دنباله ≠ دنباله‌ی محتمل‌ترین حالت‌ها !!!

محتمل‌ترین مسیر تا هر $\mathbf{x}_{t+1}$ = محتمل‌ترین مسیر به **یک** $\mathbf{x}_t$ + یک گام دیگر

$$\max_{\mathbf{x}_1\ldots\mathbf{x}_t} \mathbf{P}(\mathbf{x}_1,\ldots,\mathbf{x}_t,\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1})$$
$$= \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \max_{\mathbf{x}_t}\left(\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t) \max_{\mathbf{x}_1\ldots\mathbf{x}_{t-1}} P(\mathbf{x}_1,\ldots,\mathbf{x}_{t-1},\mathbf{x}_t|\mathbf{e}_{1:t})\right)$$

مشابه فیلتر کردن، به‌جز اینکه $\mathbf{f}_{1:t}$ جایگزین می‌شود با:

$$\mathbf{m}_{1:t} = \max_{\mathbf{x}_1\ldots\mathbf{x}_{t-1}} \mathbf{P}(\mathbf{x}_1,\ldots,\mathbf{x}_{t-1},\mathbf{X}_t|\mathbf{e}_{1:t})$$

محتمل‌ترین تبیین

الگوریتم وتربای

VITERBI ALGORITHM

محتمل‌ترین تبیین
Most Likely Explanation

کاربرد در بازشناسی گفتار، کدگشایی با یک کانال نویزی

Most likely explanation: $\arg\max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$

$$\mathbf{m}_{1:t} = \max_{\mathbf{x}_1 \ldots \mathbf{x}_{t-1}} \mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{X}_t|\mathbf{e}_{1:t})$$

$\mathbf{m}_{1:t}(i)$ احتمال محتمل‌ترین مسیر منتهی به حالت i را می‌دهد.

برای به‌هنگام‌سازی، مجموع‌گیری با ماکزیمم‌گیری جایگزین می‌شود، که الگوریتم وتربای را به‌دست می‌دهد:

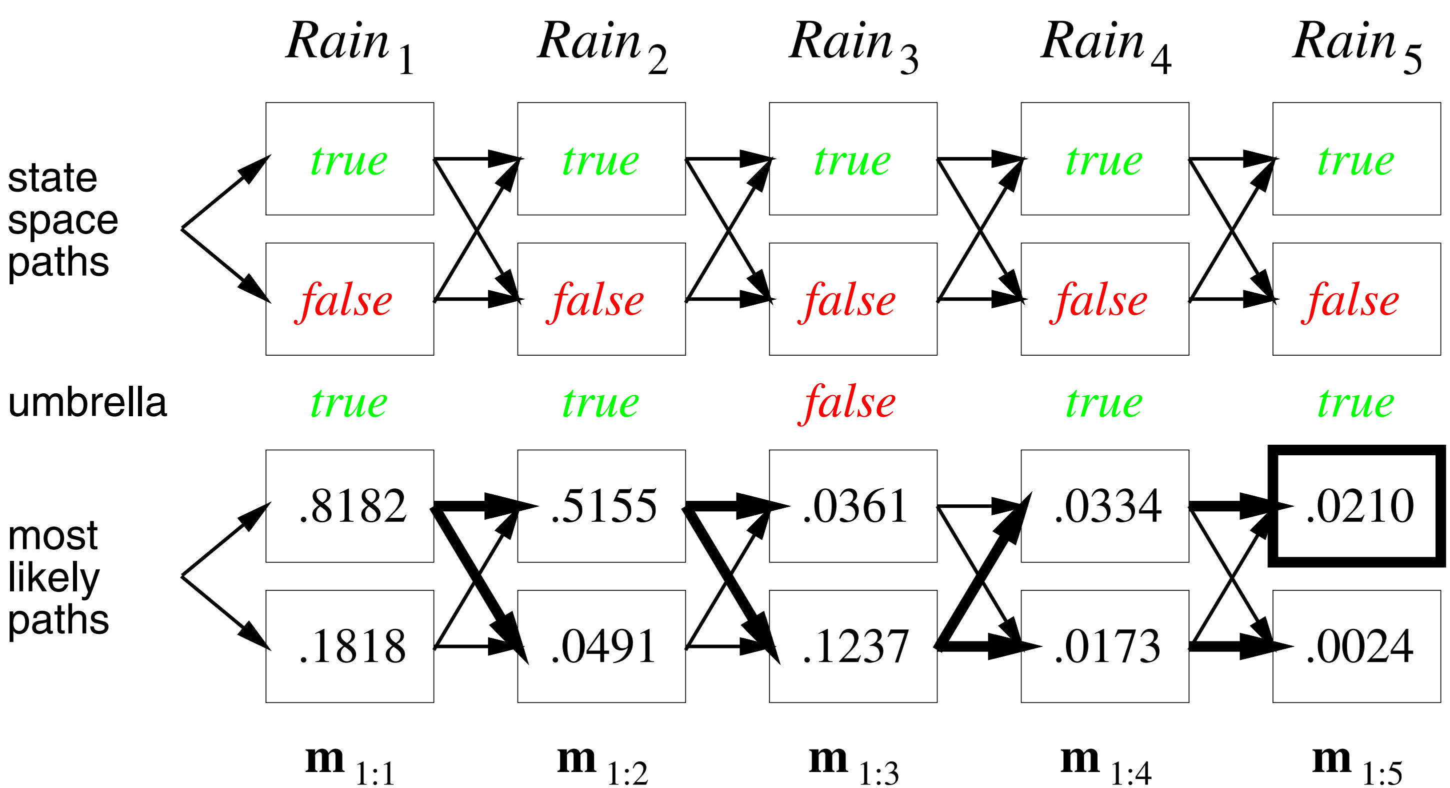
$$\mathbf{m}_{1:t+1} = \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \max_{\mathbf{x}_t} (\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)\mathbf{m}_{1:t})$$

محتمل‌ترین تبیین

الگوریتم وتربای: مثال

VITERBI ALGORITHM

می‌توان دنباله‌های حالت ممکن برای $Rain_t$ را در قالب مسیرهایی در نمودار حالت‌های ممکن در نظر گرفت:



عملکرد الگوریتم وتربای برای دنباله‌ی مشاهدات چتر داده شده:

مقادیر پیام $\mathbf{m}_{1:t}$ برای هر t: احتمال بهترین دنباله‌ی حاصل از هر حالت در زمان t

استدلال احتمالاتی در امتداد زمان

۳

مدل‌های مارکوف مخفی

مدل مارکوف

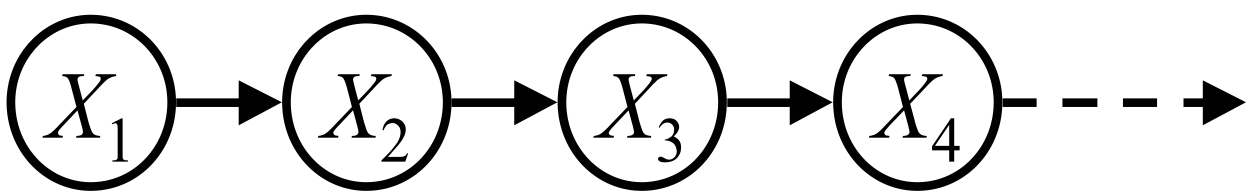
Markov Model

یک مدل مارکوف، یک شبکه‌ی بیزی با ساختار زنجیره‌ای است:

همه‌ی گره‌ها دارای یک توزیع یکسان (ایستان) هستند.

مقدار X در یک زمان داده شده، حالت (state) نام دارد.

به عنوان یک شبکه‌ی بیزی:



$P(X_1)$ $\qquad$ $P(X_t|X_{t-1})$

پارامترها: احتمالات گذر: تعیین چگونگی تغییر حالت در طول زمان

مدل مارکوف

استقلال شرطی

MARKOV MODEL

$$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \dashrightarrow$$

$P(X_1)$ $\qquad$ $P(X_t|X_{t-1})$

استقلال شرطی پایه:

- گذشته و آینده مستقل از حال است.
- هر گام زمانی، فقط وابسته به گام قبلی است **(فرض مارکوف مرتبه اول).**

زنجیره تنها یک شبکه‌ی بیزی (نمو کننده) است.

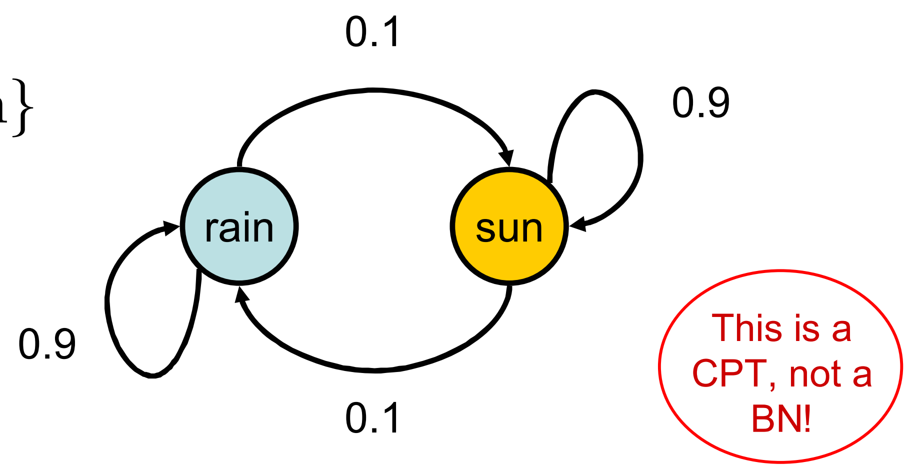
مدل مارکوف

مثال: زنجیره‌ی مارکوف

MARKOV MODEL

Weather:

- States: $X = \{\text{rain}, \text{sun}\}$
- Transitions:



- Initial distribution: 1.0 sun

توزیع احتمال بعد از یک گام چیست؟

$$P(X_2 = \text{sun}) = P(X_2 = \text{sun}|X_1 = \text{sun})P(X_1 = \text{sun}) + P(X_2 = \text{sun}|X_1 = \text{rain})P(X_1 = \text{rain})$$

$$0.9 \cdot 1.0 + 0.1 \cdot 0.0 = 0.9$$

مدل مارکوف

محاسبه‌ی احتمال بودن در حالت s در زمان t : (راه‌حل کند)

احتمال بودن در حالت s در زمان t چیست؟

برشماری همه‌ی دنباله‌های به طول t که به s ختم می‌شوند
و جمع کردن احتمالات آنها

$$P(X_t = s) = \sum_{x_1,\ldots,x_{t-1}} P(x_1, \ldots, x_{t-1}, s)$$

مثال

$$P(X_1 = sun)P(X_2 = sun|X_1 = sun)P(X_3 = sun|X_2 = sun)P(X_4 = sun|X_3 = sun)$$

$$P(X_1 = sun)P(X_2 = rain|X_1 = sun)P(X_3 = sun|X_2 = rain)P(X_4 = sun|X_3 = sun)$$

$$\vdots$$

مدل مارکوف

محاسبه‌ی احتمال بودن در حالت s در زمان t : (راه‌حل بهتر: الگوریتم پیش‌رو کوچک)

MINI-FORWARD ALGORITHM

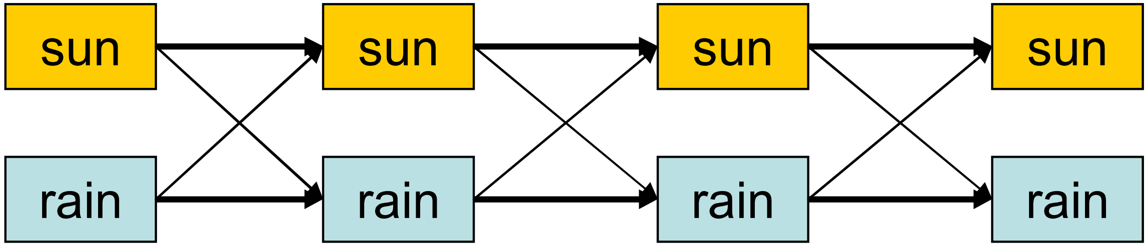
احتمال بودن در حالت s در زمان t چیست؟

با محاسبه‌ی افزایشی (به صورت بازگشتی)
باورها به صورت افزایشی بهنگام می‌شوند:

$$P(x_t) = \sum_{x_{t-1}} P(x_t|x_{t-1})P(x_{t-1})$$

$$P(x_1) = \text{known}$$

Forward simulation



مدل مارکوف

محاسبه‌ی احتمال بودن در حالت s در زمان t : مثال

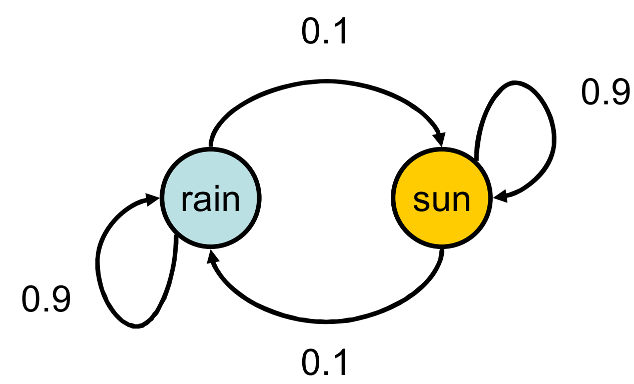
Mini-Forward Algorithm

از مشاهده‌ی اولیه‌ی sun :

$$\begin{array}{ccccc} \left\langle \begin{matrix}1.0\\0.0\end{matrix}\right\rangle & \left\langle \begin{matrix}0.9\\0.1\end{matrix}\right\rangle & \left\langle \begin{matrix}0.82\\0.18\end{matrix}\right\rangle & \Rightarrow & \left\langle \begin{matrix}0.5\\0.5\end{matrix}\right\rangle \\ P(X_1) & P(X_2) & P(X_3) & & P(X_\infty) \end{array}$$

از مشاهده‌ی اولیه‌ی rain :

$$\begin{array}{ccccc} \left\langle \begin{matrix}0.0\\1.0\end{matrix}\right\rangle & \left\langle \begin{matrix}0.1\\0.9\end{matrix}\right\rangle & \left\langle \begin{matrix}0.18\\0.82\end{matrix}\right\rangle & \Rightarrow & \left\langle \begin{matrix}0.5\\0.5\end{matrix}\right\rangle \\ P(X_1) & P(X_2) & P(X_3) & & P(X_\infty) \end{array}$$



توزیع ایستان

برای یک زنجیره

STATIONARY DISTRIBUTION

توزیع ایستان یک زنجیره:

توزیع به‌دست آمده در نهایت مستقل از توزیع اولیه است.

(برای بیشتر زنجیره‌ها این‌گونه است)

مدل مارکوف

محتمل‌ترین تبیین

Most Likely Explanation

محتمل‌ترین دنباله‌ی منتهی به حالت s در زمان t چیست؟

برای مثال: اگر حالت روز چهارم sun باشد، محتمل‌ترین دنباله چیست؟

به طور شهودی: احتمالاً هر چهار روز sun باشد.

مدل مارکوف

محتمل‌ترین تبیین: محاسبه (روش کند)

MOST LIKELY EXPLANATION

محتمل‌ترین دنباله‌ی منتهی به حالت s در زمان t چیست؟

برشماری همه‌ی دنباله‌ها و رتبه‌بندی آنها

$$P(X_t = s) = \max_{x_1,\ldots,x_{t-1}} P(x_1, \ldots, x_{t-1}, s)$$

مثال

$$P(X_1 = sun)P(X_2 = sun|X_1 = sun)P(X_3 = sun|X_2 = sun)P(X_4 = sun|X_3 = sun)$$

$$P(X_1 = sun)P(X_2 = rain|X_1 = sun)P(X_3 = sun|X_2 = rain)P(X_4 = sun|X_3 = sun)$$

$$\vdots$$

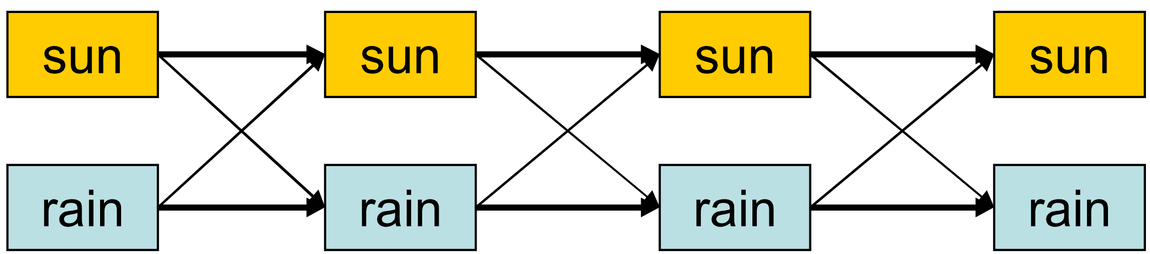
مدل مارکوف

محتمل‌ترین تبیین: محاسبه (الگوریتم وتربای کوچک)

MINI-VITERBI ALGORITHM

محتمل‌ترین دنباله‌ی منتهی به حالت s در زمان t چیست؟

با محاسبه‌ی افزایشی (به صورت بازگشتی)
باورها به صورت افزایشی بهنگام می‌شوند:



الگوریتم وتربای کوچک

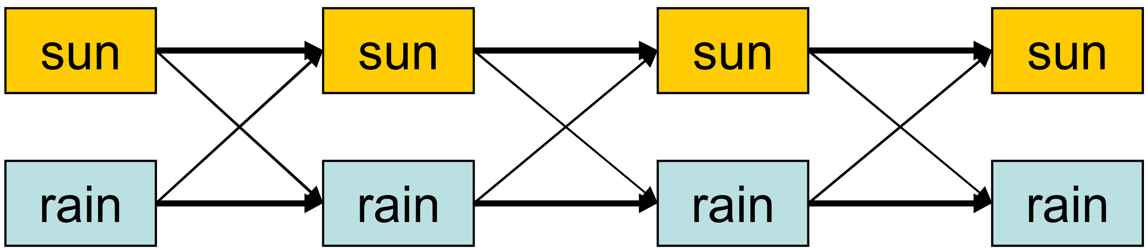
Mini-Viterbi Algorithm

تعریف می‌کنیم:

$$m_t[x] = \max_{x_{1:t-1}} P(x_{1:t-1}, x)$$

$$a_t[x] = \arg\max_{x_{1:t-1}} P(x_{1:t-1}, x)$$

بهترین دنباله را از بردارهای m و a می‌خوانیم.



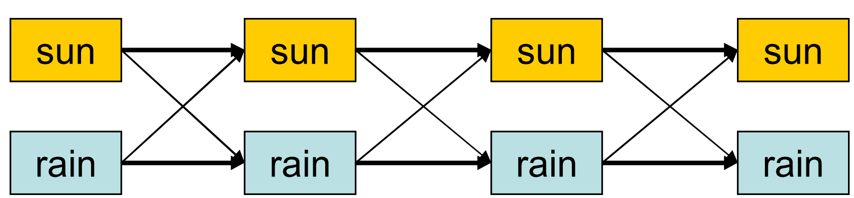
مدل مارکوف

الگوریتم وتربای کوچک: رابطه‌ی بازگشتی

MINI-VITERBI ALGORITHM

$$
\begin{aligned}
m_t[x] &= \max_{x_{1:t-1}} P(x_{1:t-1}, x) \\
&= \max_{x_{1:t-1}} P(x_{1:t-1}) P(x \mid x_{t-1}) \\
&= \max_{x_{t-1}} P(x \mid x_{t-1}) \max_{x_{1:t-2}} P(x_{1:t-1}) \\
&= \max_{x_{t-1}} P(x \mid x_{t-1}) m_{t-1}[x]
\end{aligned}
$$

$$m_1[x] = P(x_1)$$



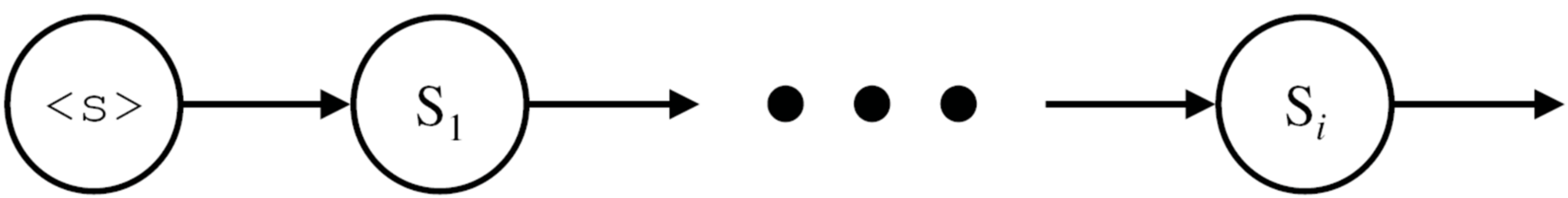
مدل مارکوف

الگوریتم وتربای کوچک: مثال (۱ از ۴)

MINI-VITERBI ALGORITHM

s	$\pi(s)$
<s>	1

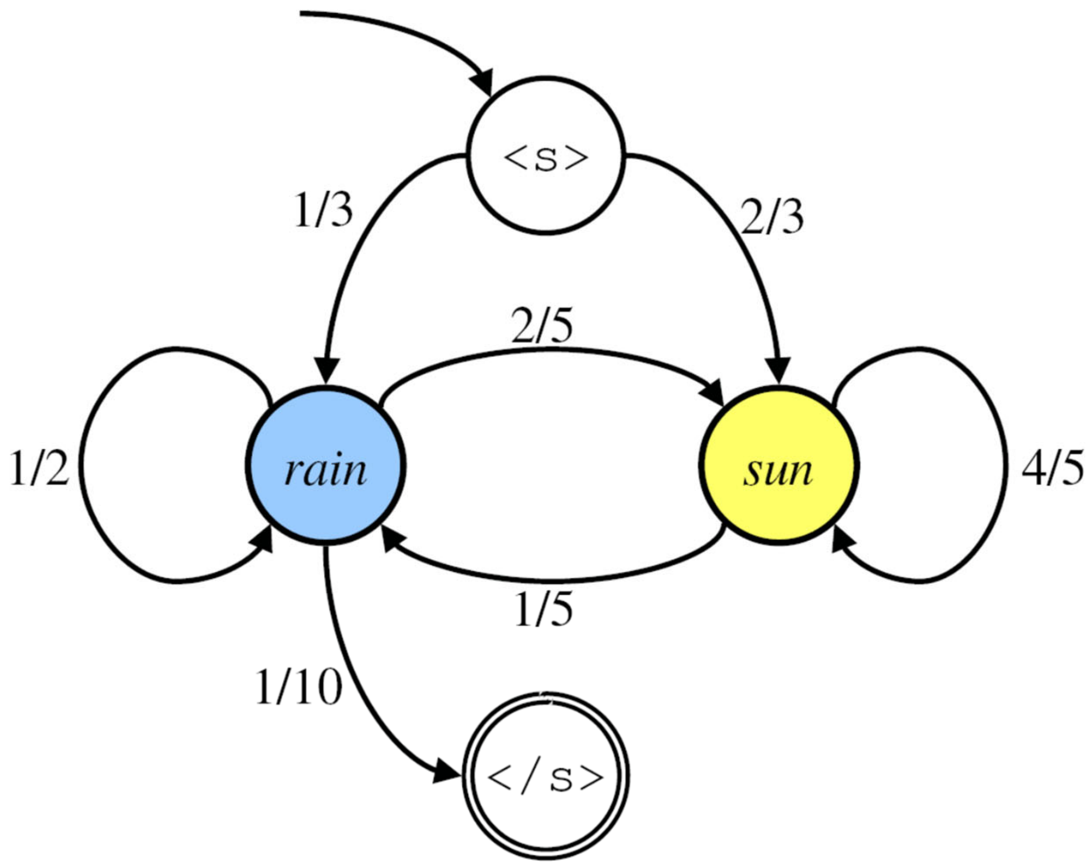
s'	s	$t(s \mid s')$
rain	*rain*	3/5
rain	*sun*	2/5
sun	*rain*	1/5
sun	*sun*	4/5
<s>	*rain*	1/3
<s>	*sun*	2/3



مدل مارکوف

الگوریتم وتربای کوچک: مثال (۲ از ۴)

MINI-VITERBI ALGORITHM

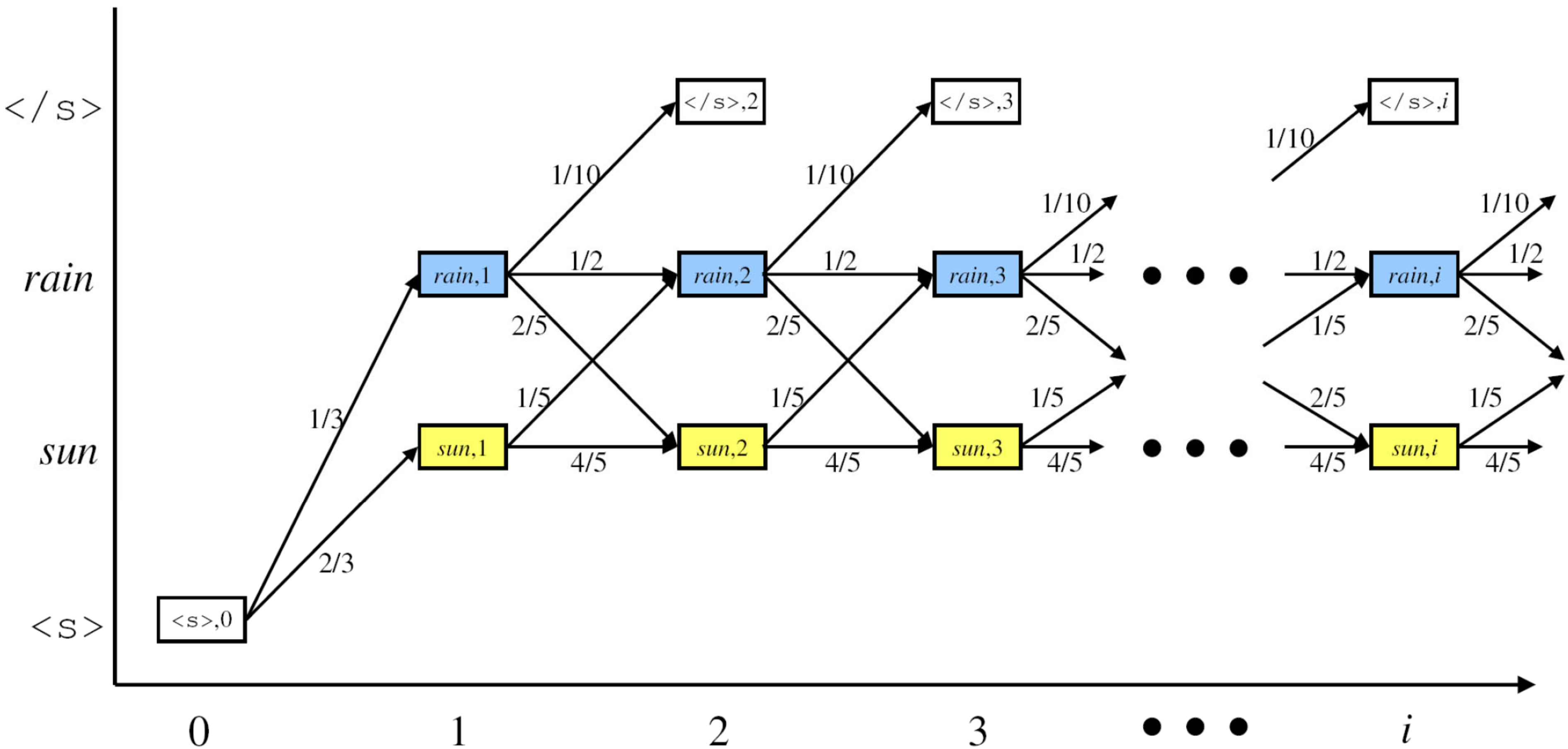


s’	s	t(s | s’)
rain	*rain*	1/2
rain	*sun*	2/5
sun	*rain*	1/5
sun	*sun*	4/5
<s>	*rain*	1/3
<s>	*sun*	2/3

مدل مارکوف

الگوریتم وتربای کوچک: مثال (۳ از ۴)

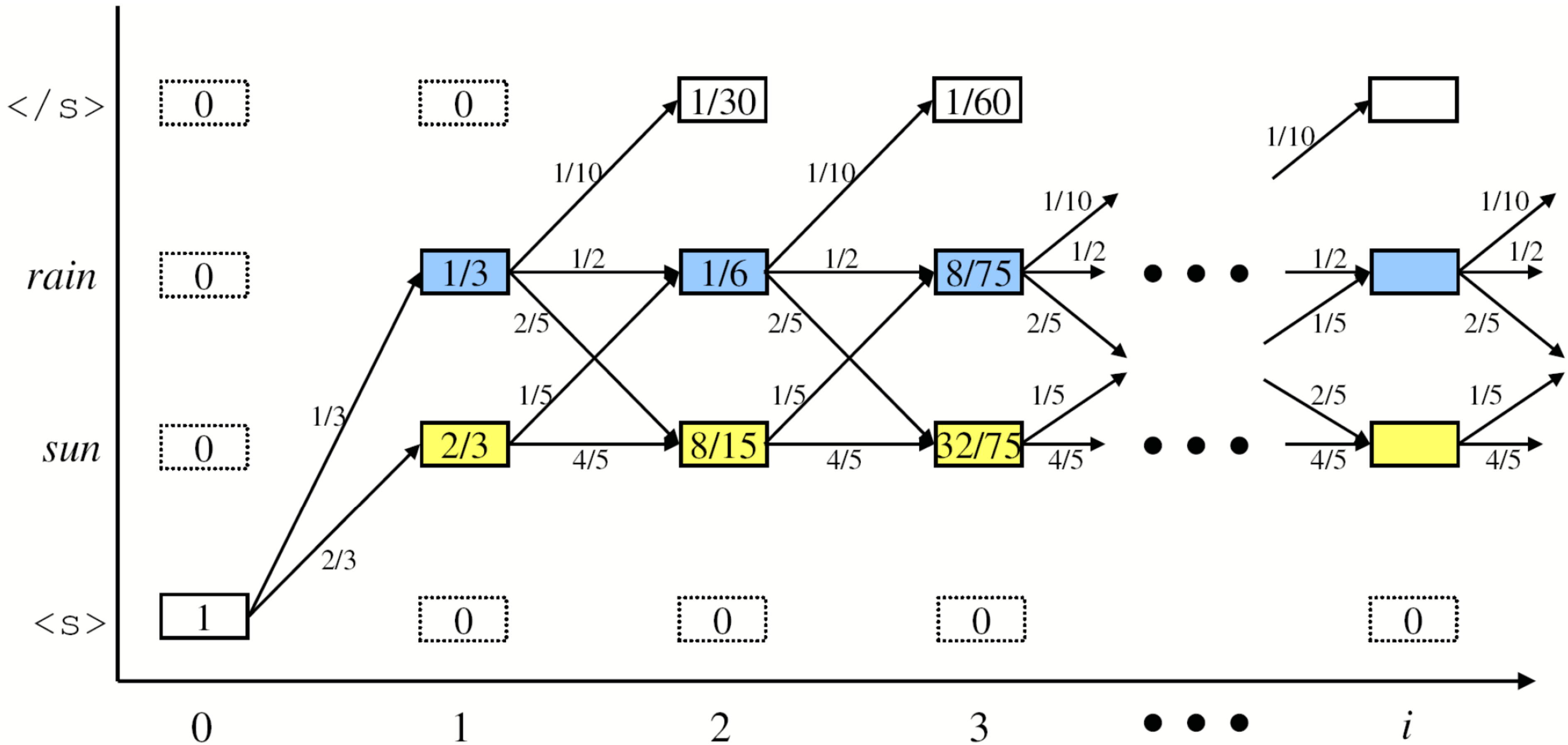
MINI-VITERBI ALGORITHM



مدل مارکوف

الگوریتم وتربای کوچک: مثال (۴ از ۴)

MINI-VITERBI ALGORITHM



مروری بر احتمالات

قانون‌های احتمال

- Marginalization

$$P(a) = \sum_b P(a, b)$$

- Definition of conditional probability

$$P(a|b) = P(a, b)/P(b)$$

- Chain rule

$$P(a, b, c) = P(a)P(b|a)P(c|a, b)$$

- Combinations, e.g. conditional chain rule

$$P(b, c|a) = P(b|a)P(c|a, b)$$

مروری بر احتمالات

قانون‌های احتمال

- Chain rule (always true)

$$P(a, b, c) = P(a)P(b|a)P(c|a, b)$$

- With A and C independent given B

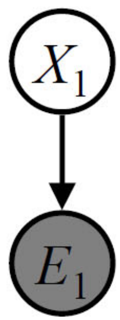
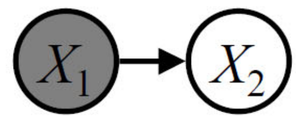
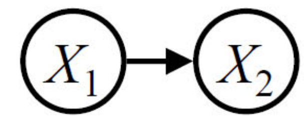
$$P(a, b, c) = P(a)P(b|a)P(c|a)$$

- If we want a conditional distribution over A, can just normalize the corresponding joint wrt A

$$P(a|b) = P(a, b)/P(b)$$
$$\propto_A P(a, b)$$

مروری بر احتمالات

مدل‌های ساده‌ی احتمالات

مدل			
پرس‌وجو	$P(X_1 \| e_1)$	$P(X_2 \| x_1)$	$P(X_2)$

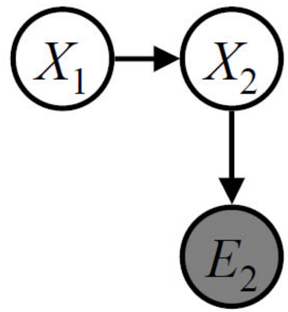
$$\begin{aligned} P(x_1|e_1) &= P(x_1, e_1)/P(e_1) \\ &\propto_{X_1} P(x_1, e_1) \\ &= P(x_1)P(e_1|x_1) \end{aligned}$$

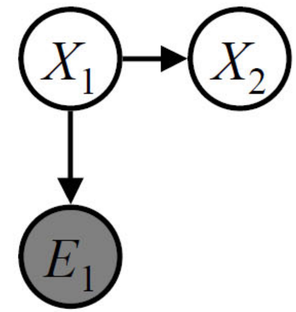
$$\begin{aligned} P(x_2) &= \sum_{x_1} P(x_1, x_2) \\ &= \sum_{x_1} P(x_1)P(x_2|x_1) \end{aligned}$$

مروری بر احتمالات

مدل‌های ساده‌ی احتمالات

مدل





پرس‌وجو

$$P(X_2|e_2)$$

$$P(X_2|e_1)$$

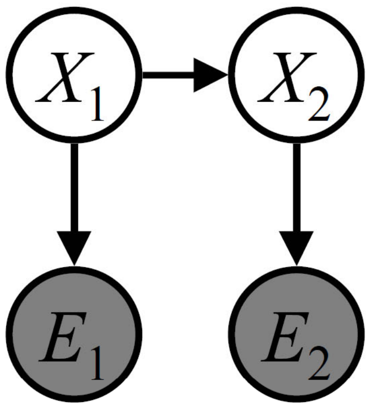
$$\begin{aligned}
P(x_2|e_2) &= P(x_2, e_2)/P(e_2) \\
&\propto_{X_2} P(x_2, e_2) \\
&= P(x_2)P(e_2|x_2) \\
&= \sum_{x_1} P(x_1, x_2)P(e_2|x_2) \\
&= P(e_2|x_2) \sum_{x_1} P(x_1)P(x_2|x_1)
\end{aligned}$$

$$\begin{aligned}
P(x_2|e_1) &= P(x_2, e_1)/P(e_1) \\
&\propto_{X_2} P(x_2, e_1) \\
&= \sum_{x_1} P(x_1, x_2, e_1) \\
&= \sum_{x_1} P(x_2|x_1)P(e_1|x_1)P(x_1)
\end{aligned}$$

مروری بر احتمالات

مدل‌های ساده‌ی احتمالات

مدل



پرس‌وجو

$$P(X_2|e_2, e_1)$$

$$\propto_{X_2} P(x_2, e_1, e_2)$$

$$= \sum_{x_1} P(x_1, x_2, e_1, e_2)$$

$$= \sum_{x_1} P(x_1)P(x_2|x_1)P(e_1|x_1)P(e_2|x_2)$$

$$= P(e_2|x_2) \sum_{x_1} P(x_2|x_1)P(e_1|x_1)P(x_1)$$

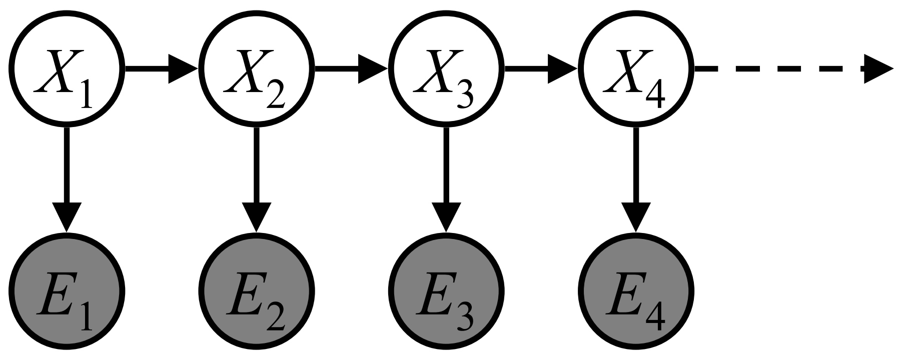
مدل مخفی مارکوف

HIDDEN MARKOV MODEL

خروجی‌ها (اثرات) را در هر گام زمانی مشاهده می‌کنیم

(حالت‌ها پنهان / مشاهده‌ناپذیر هستند.)

به عنوان یک شبکه‌ی بیزی:



$P(X_t)$ — **توزیع آغازین** *Initial Distribution*

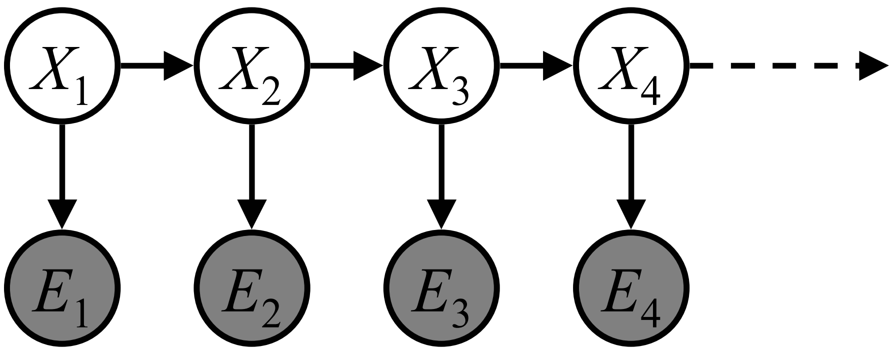
$P(X_t|X_{t-1})$ — **گذرها** *Transitions*

$P(E_t|X_t)$ — **صدورها** *Emissions*

مدل مخفی مارکوف

استقلال شرطی

MARKOV MODEL



دو خاصیت مهم استقلال برای HMM :

- در فرآیند مخفی مارکوف، **آینده** از طریق **حال** به **گذشته** وابسته است.
- مشاهده‌ی فعلی مستقل از همه‌ی حالت‌های فعلی داده شده است.

مدل مخفی مارکوف

الگوریتم پیش‌رو

FORWARD ALGORITHM

احتمال بودن در حالت s در زمان t چیست؟

با داشتن حالت باور فعلی، چگونه آن را با یک شاهد بهنگام کنیم؟ (monitoring, filtering)

$$P(X_t = x_t | e_{1:t})$$

$$\begin{aligned}
P(x_t|e_{1:t}) &\propto P(x_t, e_{1:t}) \\
&= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t}) \\
&= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) P(x_t|x_{t-1}) P(e_t|x_t) \\
&= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}, e_{1:t-1})
\end{aligned}$$

مدل مخفی مارکوف

الگوریتم پیش‌رو: رابطه‌ی بازگشتی

FORWARD ALGORITHM

تعریف می‌کنیم:

$$f_t[x_t] = P(x_t, e_{1:t})$$

با جایگذاری به رابطه‌ی بازگشتی زیر می‌رسیم:

$$f_t[x_t] = P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) f_{t-1}[x_{t-1}]$$

مدل مخفی مارکوف

محتمل‌ترین تبیین: محاسبه (الگوریتم وترباي)

Viterbi Algorithm

محتمل‌ترین دنباله از حالت‌ها با داشتن مشاهده‌ها چیست؟

با محاسبه‌ی افزایشی (به صورت بازگشتی)
باورها به صورت افزایشی بهنگام می‌شوند:

$$x^*_{1:T} = \arg\max_{x_{1:T}} P(x_{1:T}|e_{1:T})$$

$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$

$$= \max_{x_{1:t-1}} P(x_{1:t-1}, e_{1:t-1}) P(x_t|x_{t-1}) P(e_t|x_t)$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) \max_{x_{1:t-2}} P(x_{1:t-1}, e_{1:t-1})$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]$$

Hidden Markov Models (HMM)

یک متغیر گسسته $\mathbf{X}_t$ is a single, discrete variable (usually $\mathbf{E}_t$ is too)
Domain of X_t is $\{1, \ldots, S\}$

ماتریس گذر Transition matrix $\mathbf{T}_{ij} = P(X_t = j | X_{t-1} = i)$, e.g., $\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$

ماتریس حس‌گر Sensor matrix $\mathbf{O}_t$ for each time step, diagonal elements $P(e_t | X_t = i)$
e.g., with $U_1 = true$, $\mathbf{O}_1 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$

Forward and backward messages as column vectors:

$$\begin{aligned} \mathbf{f}_{1:t+1} &= \alpha \mathbf{O}_{t+1} \mathbf{T}^\top \mathbf{f}_{1:t} \\ \mathbf{b}_{k+1:t} &= \mathbf{T}\mathbf{O}_{k+1} \mathbf{b}_{k+2:t} \end{aligned}$$

Forward-backward algorithm needs time $O(S^2 t)$ and space $O(St)$

مدل‌های مارکوف مخفی

الگوریتم «کانتری دانس»

Country Dance Algorithm

با اجرای پس‌روی الگوریتم پیش‌رو،
می‌توان از ذخیره‌سازی همه‌ی پیام‌های پیش‌رو در مرحله‌ی هموارسازی اجتناب کرد:

$$\begin{aligned}
\mathbf{f}_{1:t+1} &= \alpha \mathbf{O}_{t+1} \mathbf{T}^\top \mathbf{f}_{1:t} \\
\mathbf{O}_{t+1}^{-1} \mathbf{f}_{1:t+1} &= \alpha \mathbf{T}^\top \mathbf{f}_{1:t} \\
\alpha' (\mathbf{T}^\top)^{-1} \mathbf{O}_{t+1}^{-1} \mathbf{f}_{1:t+1} &= \mathbf{f}_{1:t}
\end{aligned}$$

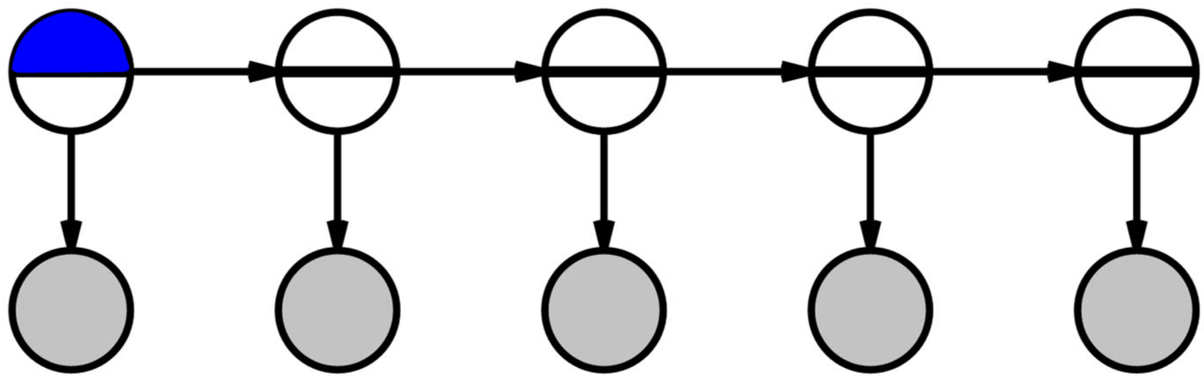
الگوریتم: گذر پیش‌رو $\mathbf{f}_t$ را محاسبه می‌کند؛ گذر پس‌رو $\mathbf{f}_i$ و $\mathbf{b}_i$ را محاسبه می‌کند.

مدل‌های مارکوف مخفی

الگوریتم «کانتری دانس»: مثال (۱ از ۱۰)

COUNTRY DANCE ALGORITHM

الگوریتم: گذر پیش‌رو $\mathbf{f}_t$ را محاسبه می‌کند؛ گذر پس‌رو $\mathbf{f}_i$ و $\mathbf{b}_i$ را محاسبه می‌کند.

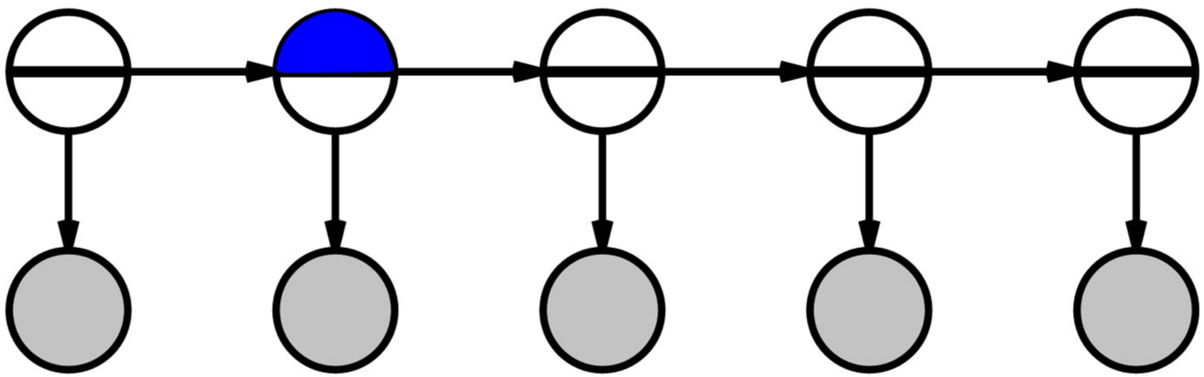


مدل‌های مارکوف مخفی

الگوریتم «کانتری دانس»: مثال (۲ از ۱۰)

COUNTRY DANCE ALGORITHM

الگوریتم: گذر پیش‌رو $\mathbf{f}_t$ را محاسبه می‌کند؛ گذر پس‌رو $\mathbf{f}_i$ و $\mathbf{b}_i$ را محاسبه می‌کند.

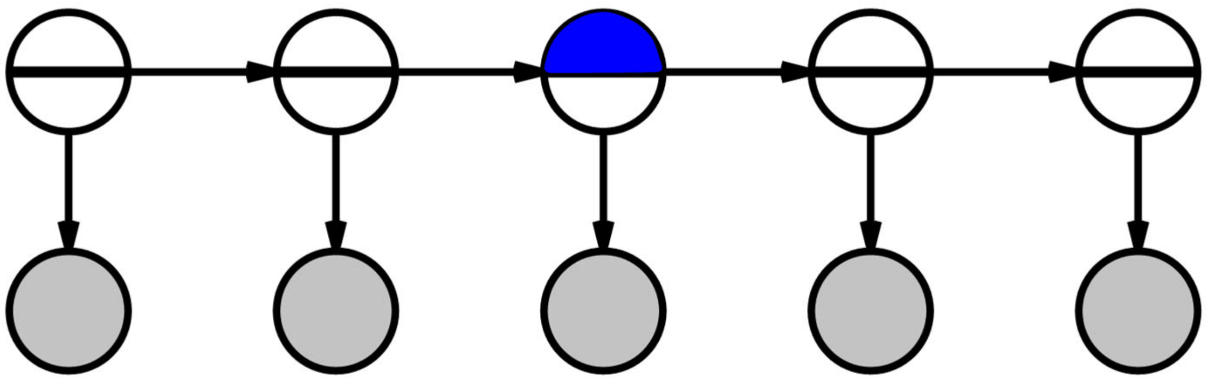


مدل‌های مارکوف مخفی

الگوریتم «کانتری دانس»: مثال (۳ از ۱۰)

COUNTRY DANCE ALGORITHM

الگوریتم: گذر پیش‌رو $\mathbf{f}_t$ را محاسبه می‌کند؛ گذر پس‌رو $\mathbf{f}_i$ و $\mathbf{b}_i$ را محاسبه می‌کند.

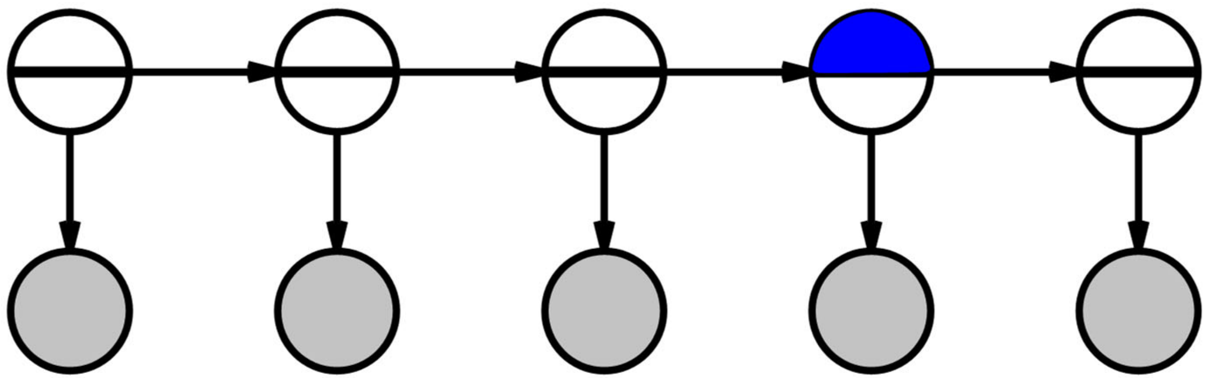


مدل‌های مارکوف مخفی

الگوریتم «کانتری دانس»: مثال (۴ از ۱۰)

COUNTRY DANCE ALGORITHM

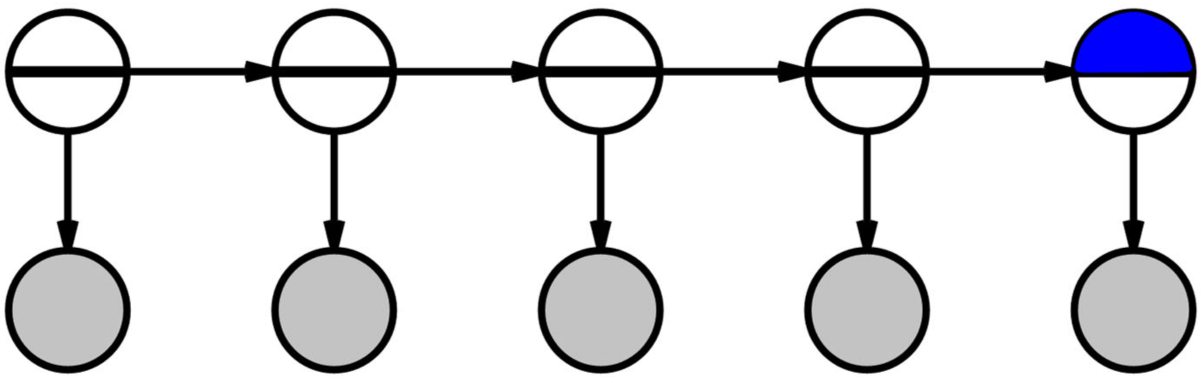
الگوریتم: گذر پیش‌رو $\mathbf{f}_t$ را محاسبه می‌کند؛ گذر پس‌رو $\mathbf{f}_i$ و $\mathbf{b}_i$ را محاسبه می‌کند.



الگوریتم «کانتری دانس»: مثال (۵ از ۱۰)

COUNTRY DANCE ALGORITHM

الگوریتم: گذر پیش‌رو $\mathbf{f}_t$ را محاسبه می‌کند؛ گذر پس‌رو $\mathbf{f}_i$ و $\mathbf{b}_i$ را محاسبه می‌کند.

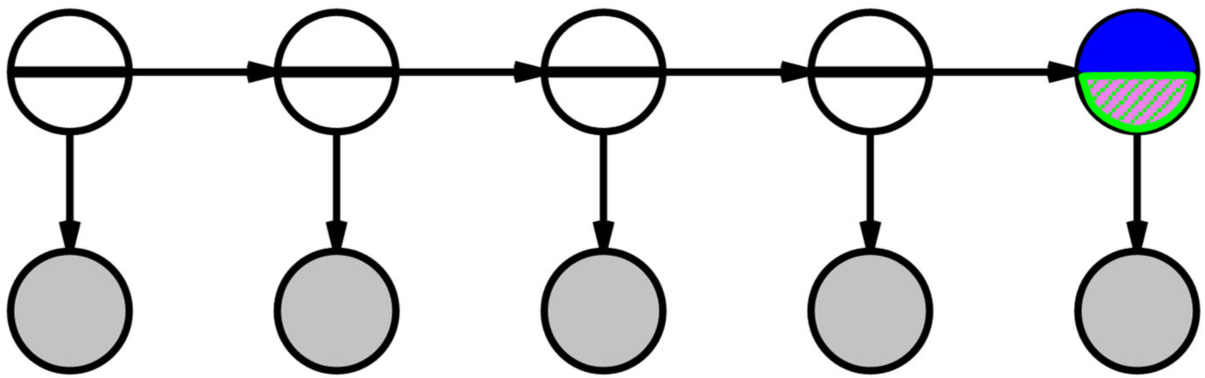


مدل‌های مارکوف مخفی

الگوریتم «کانتری دانس»: مثال (۶ از ۱۰)

Country Dance Algorithm

الگوریتم: گذر پیش‌رو $\mathbf{f}_t$ را محاسبه می‌کند؛ گذر پس‌رو $\mathbf{f}_i$ و $\mathbf{b}_i$ را محاسبه می‌کند.

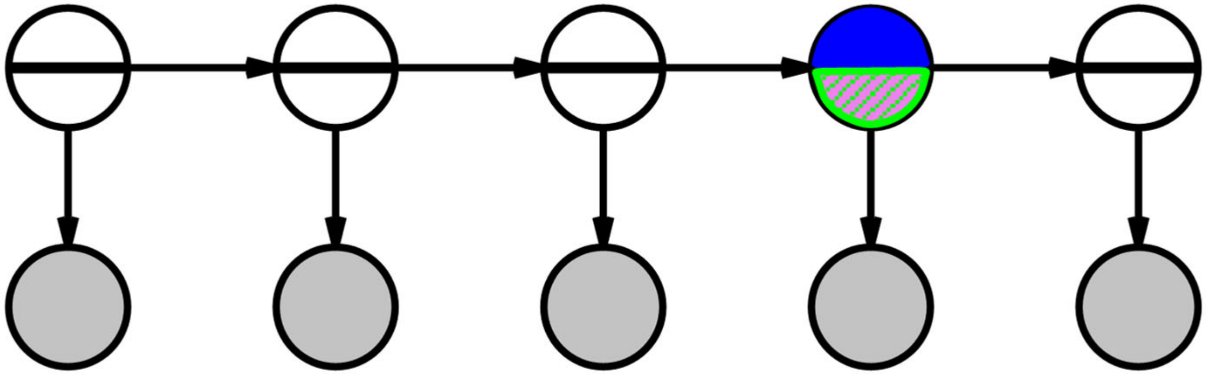


مدل‌های مارکوف مخفی

الگوریتم «کانتری دانس»: مثال (۷ از ۱۰)

COUNTRY DANCE ALGORITHM

الگوریتم: گذر پیش‌رو $\mathbf{f}_t$ را محاسبه می‌کند؛ گذر پس‌رو $\mathbf{f}_i$ و $\mathbf{b}_i$ را محاسبه می‌کند.

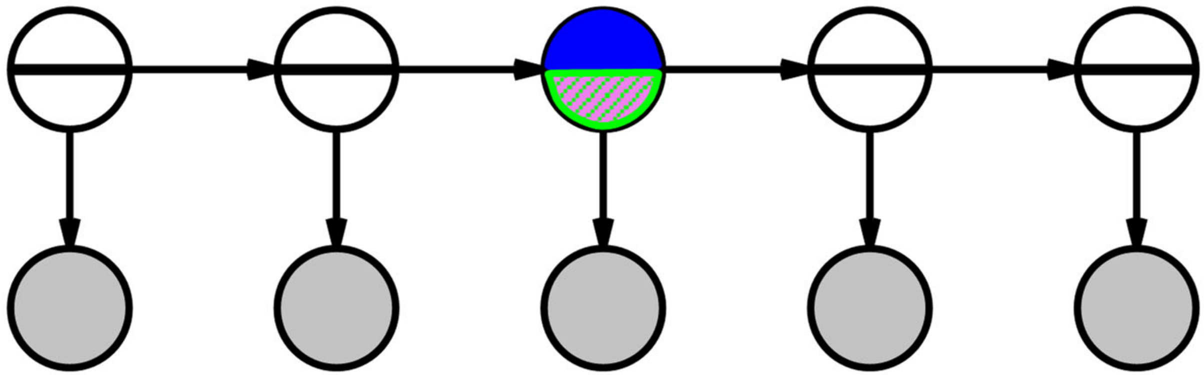


مدل‌های مارکوف مخفی

الگوریتم «کانتری دانس»: مثال (۸ از ۱۰)

COUNTRY DANCE ALGORITHM

الگوریتم: گذر پیش‌رو $\mathbf{f}_t$ را محاسبه می‌کند؛ گذر پس‌رو $\mathbf{f}_i$ و $\mathbf{b}_i$ را محاسبه می‌کند.

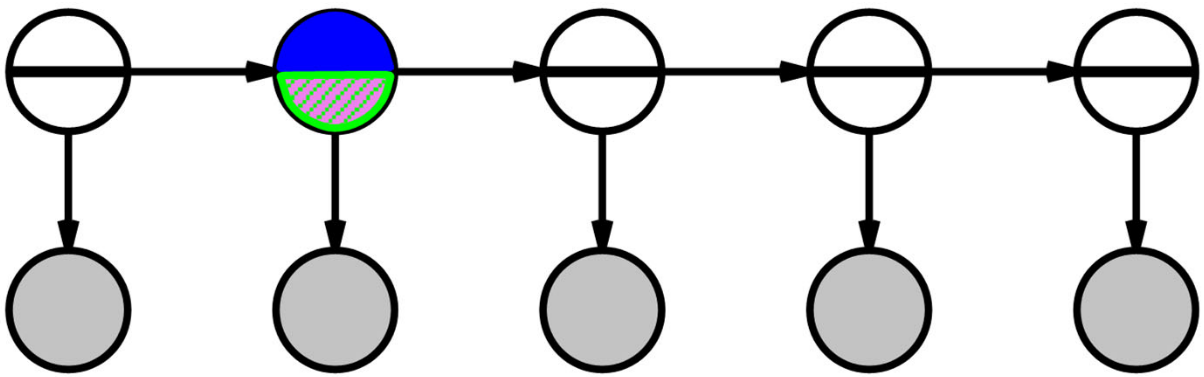


مدل‌های مارکوف مخفی

الگوریتم «کانتری دانس»: مثال (۹ از ۱۰)

COUNTRY DANCE ALGORITHM

الگوریتم: گذر پیش‌رو $\mathbf{f}_t$ را محاسبه می‌کند؛ گذر پس‌رو $\mathbf{f}_i$ و $\mathbf{b}_i$ را محاسبه می‌کند.

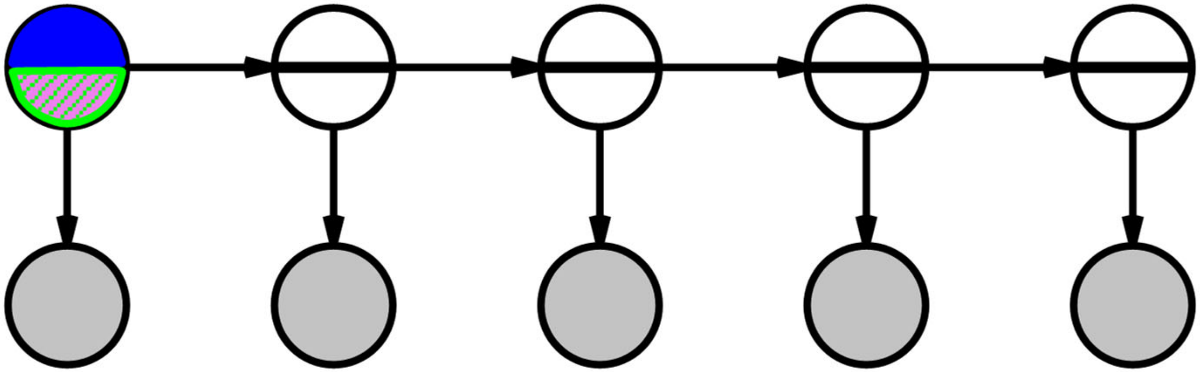


مدل‌های مارکوف مخفی

الگوریتم «کانتری دانس»: مثال (۱۰ از ۱۰)

COUNTRY DANCE ALGORITHM

الگوریتم: گذر پیش‌رو $\mathbf{f}_t$ را محاسبه می‌کند؛ گذر پس‌رو $\mathbf{f}_i$ و $\mathbf{b}_i$ را محاسبه می‌کند.



هوش مصنوعی

استدلال احتمالاتی در امتداد زمان

۴

فیلترهای کالمن

فیلترهای کالمن

KALMAN FILTERS

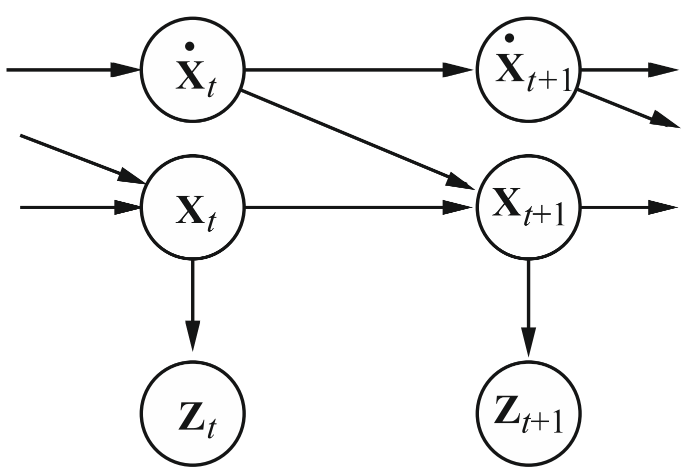
هدف : **پیش‌بینی حالت با وجود مشاهده‌های نویزی**

مدل‌سازی سیستم‌های توصیف‌شده با مجموعه‌ای از متغیرهای پیوسته

برای مثال : دنبال کردن پرواز یک پرنده :

$$\mathbf{X}_t = X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}$$

یا هواپیما، ربات، اکوسیستم، سیستم اقتصادی، پلنت شیمیایی، سیاره‌ها، ...



پیش‌فرض‌های فیلتر کالمن: مدل گذر و مدل حس‌گر گاوسی خطی

فیلترهای کالمن

به‌هنگام‌سازی توزیع‌های گاوسی

KALMAN FILTERS

گام پیش‌بینی

اگر $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$ گاوسی باشد، آن‌گاه پیش‌بینی

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

نیز گاوسی است.

اگر $\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t})$ گاوسی باشد، آن‌گاه توزیع به‌هنگام‌شده‌ی

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

نیز گاوسی است.

بنابراین، توزیع $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$ یک گاوسی $N(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ است برای هر t

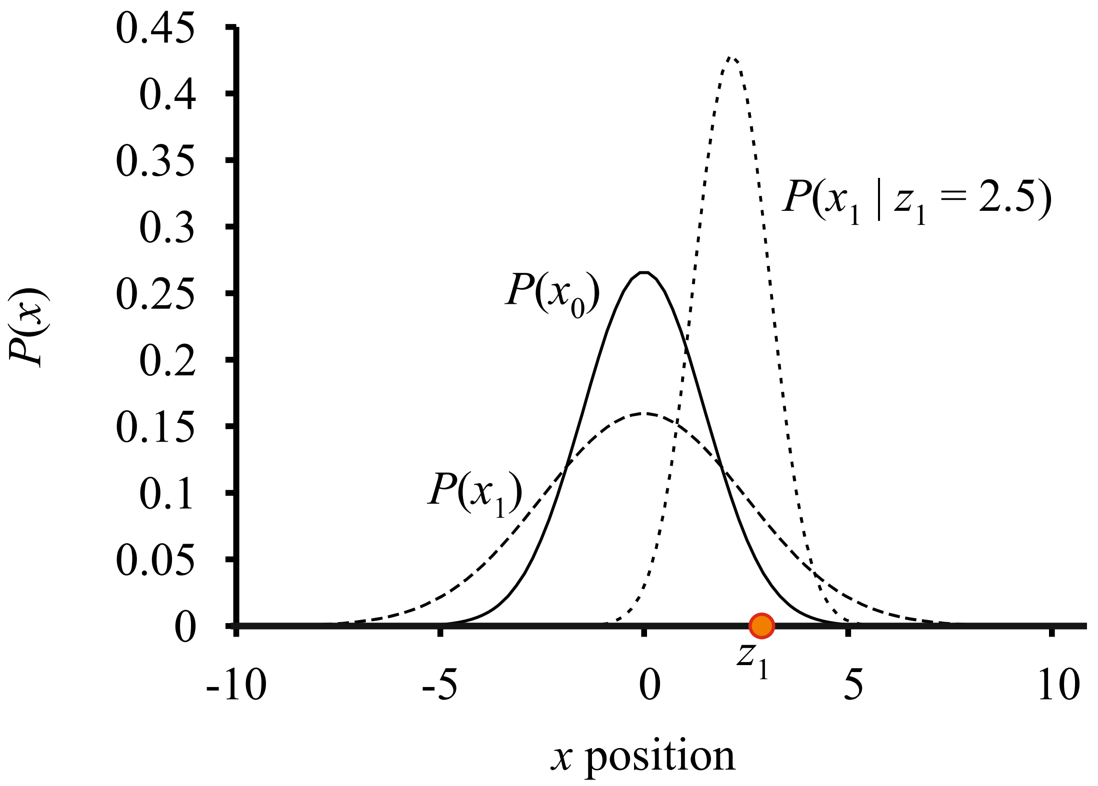
برای فرآیندهای عمومی (غیرخطی/غیرگاوسی): توصیف توزیع پسین به صورت **بی‌کران** رشد می‌کند با $t \to \infty$

مثال ساده‌ی یک‌بعدی

KALMAN FILTERS

Gaussian random walk on X–axis, s.d. σ_x, sensor s.d. σ_z

$$\mu_{t+1} = \frac{(\sigma_t^2 + \sigma_x^2) z_{t+1} + \sigma_z^2 \mu_t}{\sigma_t^2 + \sigma_x^2 + \sigma_z^2} \qquad \sigma_{t+1}^2 = \frac{(\sigma_t^2 + \sigma_x^2)\sigma_z^2}{\sigma_t^2 + \sigma_x^2 + \sigma_z^2}$$



General Kalman Update

Transition and sensor models:

$$\begin{aligned} P(\mathbf{x}_{t+1}|\mathbf{x}_t) &= N(\mathbf{F}\mathbf{x}_t, \boldsymbol{\Sigma}_x)(\mathbf{x}_{t+1}) \\ P(\mathbf{z}_t|\mathbf{x}_t) &= N(\mathbf{H}\mathbf{x}_t, \boldsymbol{\Sigma}_z)(\mathbf{z}_t) \end{aligned}$$

$\mathbf{F}$ is the matrix for the transition; $\boldsymbol{\Sigma}_x$ the transition noise covariance
$\mathbf{H}$ is the matrix for the sensors; $\boldsymbol{\Sigma}_z$ the sensor noise covariance

Filter computes the following update:

$$\begin{aligned} \boldsymbol{\mu}_{t+1} &= \mathbf{F}\boldsymbol{\mu}_t + \mathbf{K}_{t+1}(\mathbf{z}_{t+1} - \mathbf{H}\mathbf{F}\boldsymbol{\mu}_t) \\ \boldsymbol{\Sigma}_{t+1} &= (\mathbf{I} - \mathbf{K}_{t+1})(\mathbf{F}\boldsymbol{\Sigma}_t\mathbf{F}^\top + \boldsymbol{\Sigma}_x) \end{aligned}$$

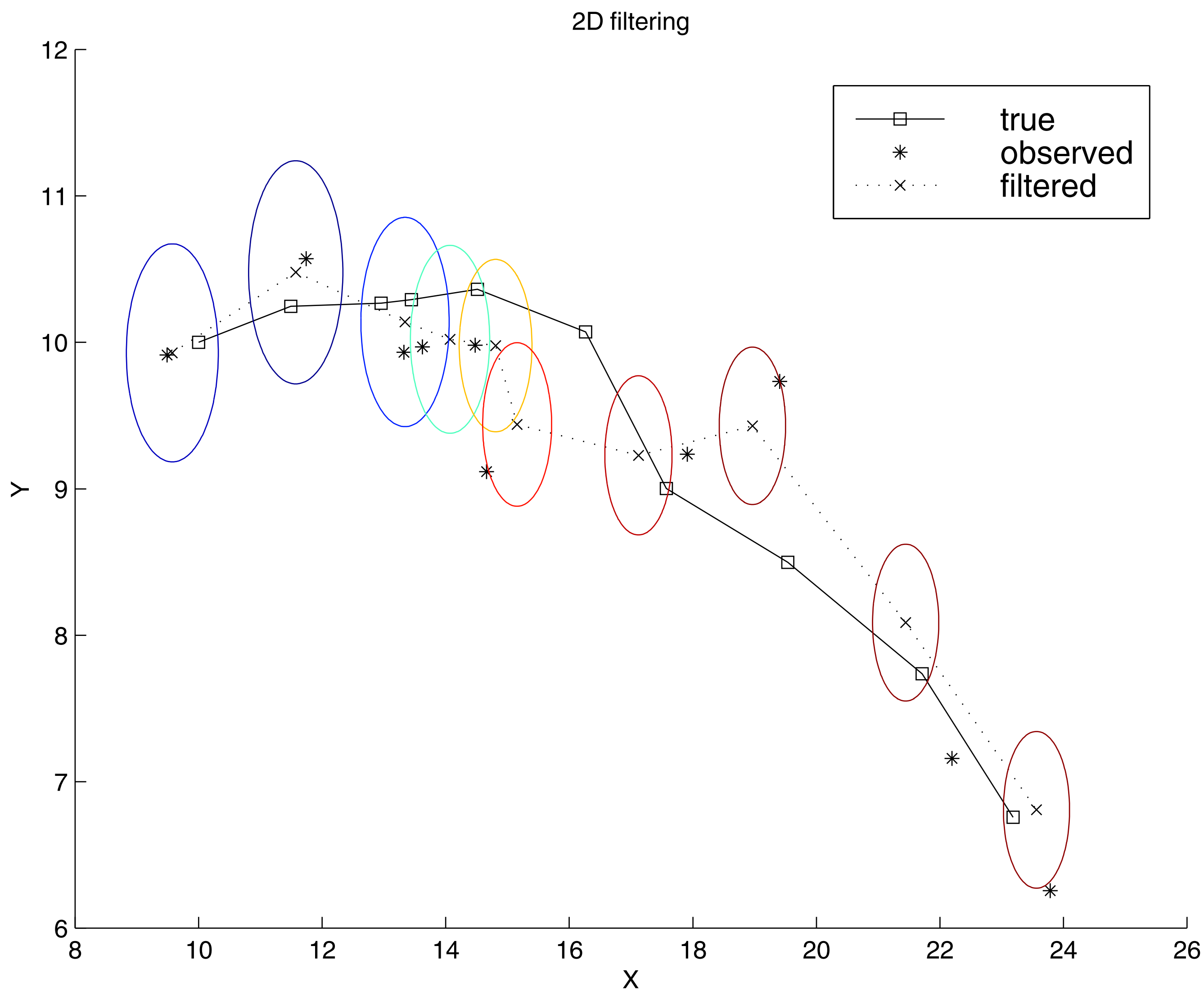
where $\mathbf{K}_{t+1} = (\mathbf{F}\boldsymbol{\Sigma}_t\mathbf{F}^\top + \boldsymbol{\Sigma}_x)\mathbf{H}^\top(\mathbf{H}(\mathbf{F}\boldsymbol{\Sigma}_t\mathbf{F}^\top + \boldsymbol{\Sigma}_x)\mathbf{H}^\top + \boldsymbol{\Sigma}_z)^{-1}$
is the Kalman gain matrix

$\boldsymbol{\Sigma}_t$ and $\mathbf{K}_t$ are independent of observation sequence, so compute offline

فیلترهای کالمن

مثال: فیلتر کردن برای دنبال کردن دوبعدی

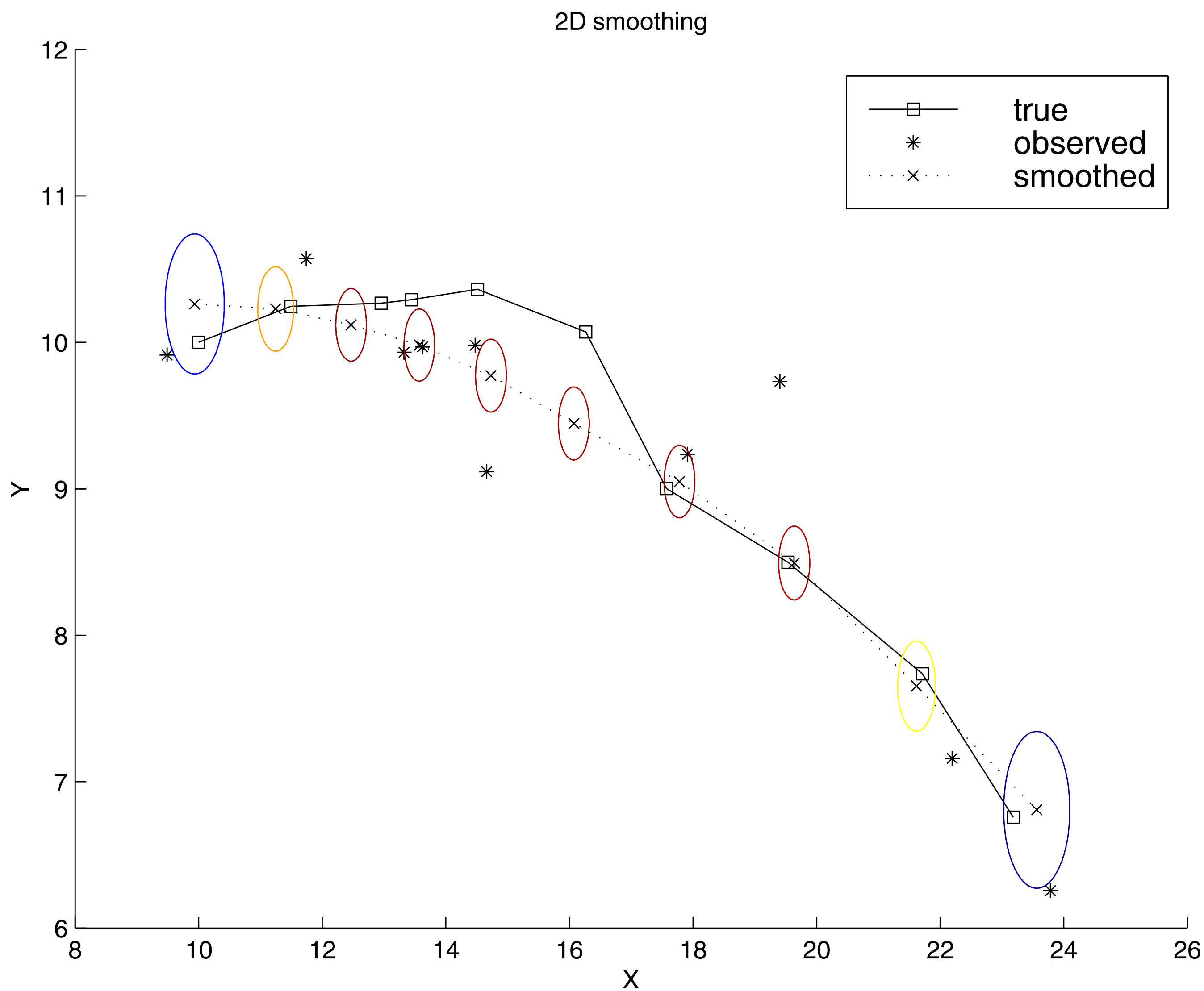
2-D Tracking Example: Filtering



فیلترهای کالمن

مثال: هموارسازی برای دنبال کردن دوبعدی

2-D Tracking Example: Filtering



فیلترهای کالمن

جایی که فیلتر کالمن شکست می‌خورد

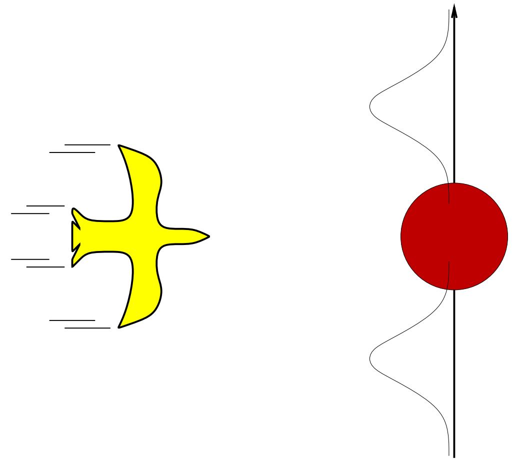
WHERE IT BREAKS

فیلتر کالمن نمی‌تواند استفاده شود اگر مدل گذار غیرخطی باشد.

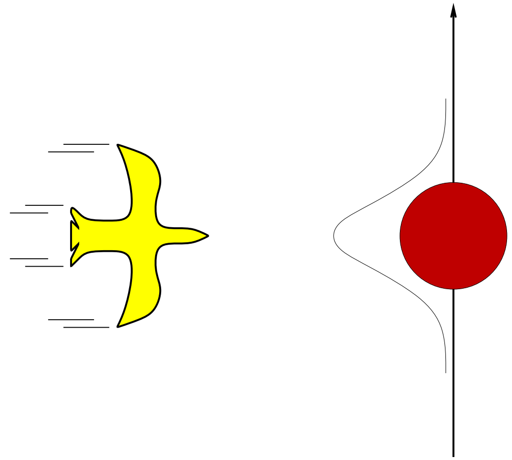
فیلتر کالمن گسترش‌یافته
Extended Kalman Filter

گذرها را به‌صورت خطی محلی حول بردار میانگین $\mathbf{x}_t = \boldsymbol{\mu}_t$ مدل می‌کند.
❋ اگر سیستم به‌صورت محلی ناهموار باشد، شکست می‌خورد.

مثال: پرواز یک پرنده به سمت درخت (از نمای بالا)



۲) در یک مدل واقع‌بینانه، پرنده به سوی یکی از دو طرف مانع اقدام به فرار می‌کند. (غیرخطی)



۱) فیلتر کالمن محل پرنده را با استفاده از یک گاوسی یکتا با مرکز مانع پیش‌بینی می‌کند.

استدلال احتمالاتی در امتداد زمان

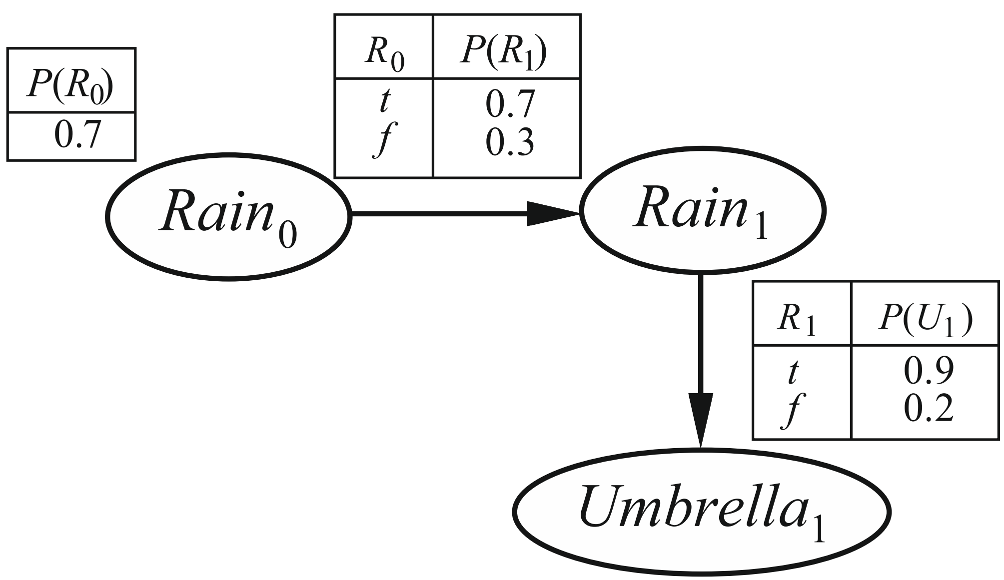
۵

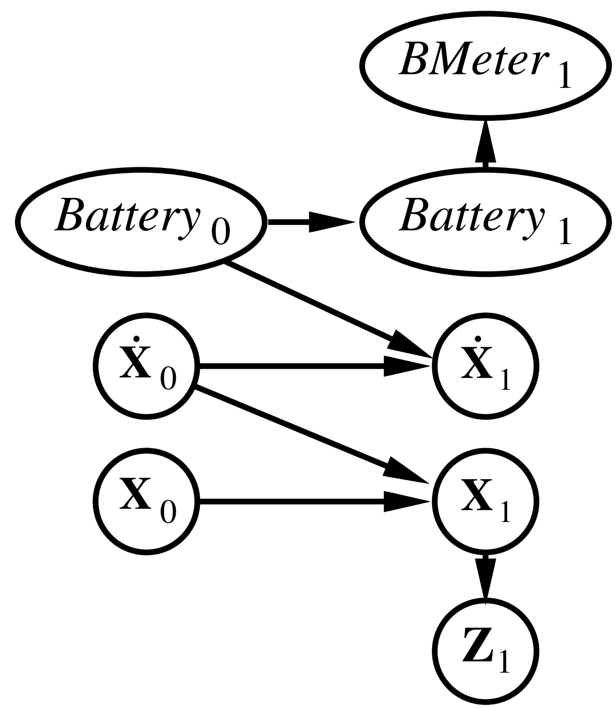
شبکه‌های بیزی پویا

شبکه‌های بیزی پویا

Dynamic Bayesian Networks

$\mathbf{X}_t$ و $\mathbf{E}_t$ حاوی تعداد دلخواه زیادی متغیر در یک شبکه‌ی بیزی باز شده هستند.



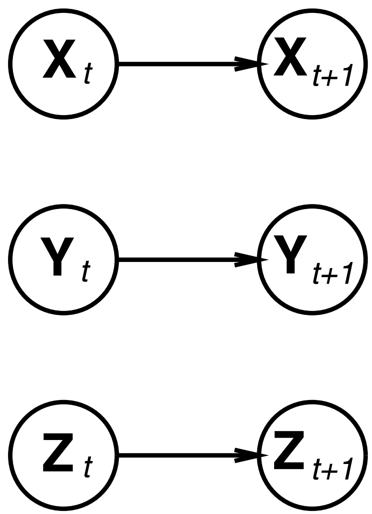


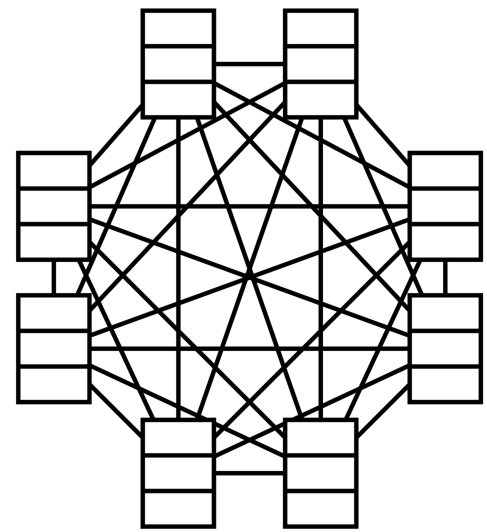
مقایسه‌ی «شبکه‌های بیزی پویا» با «مدل‌های مارکوف مخفی»

DBNs vs. HMMs

هر **HMM** یک **DBN** تک متغیره است.

هر **DBN** گسسته یک **HMM** است.





Sparse dependencies $\Rightarrow$ exponentially fewer parameters;
e.g., 20 state variables, three parents each
DBN has $20 \times 2^3 = 160$ parameters, HMM has $2^{20} \times 2^{20} \approx 10^{12}$

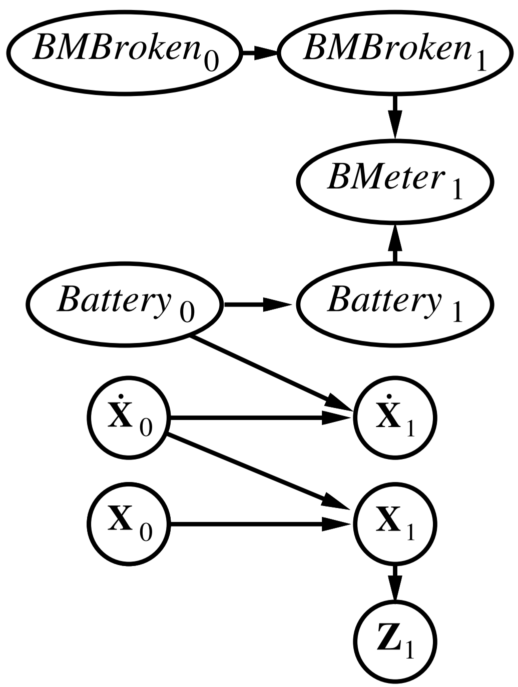
مقایسه‌ی «شبکه‌های بیزی پویا» با «فیلترهای کالمن»

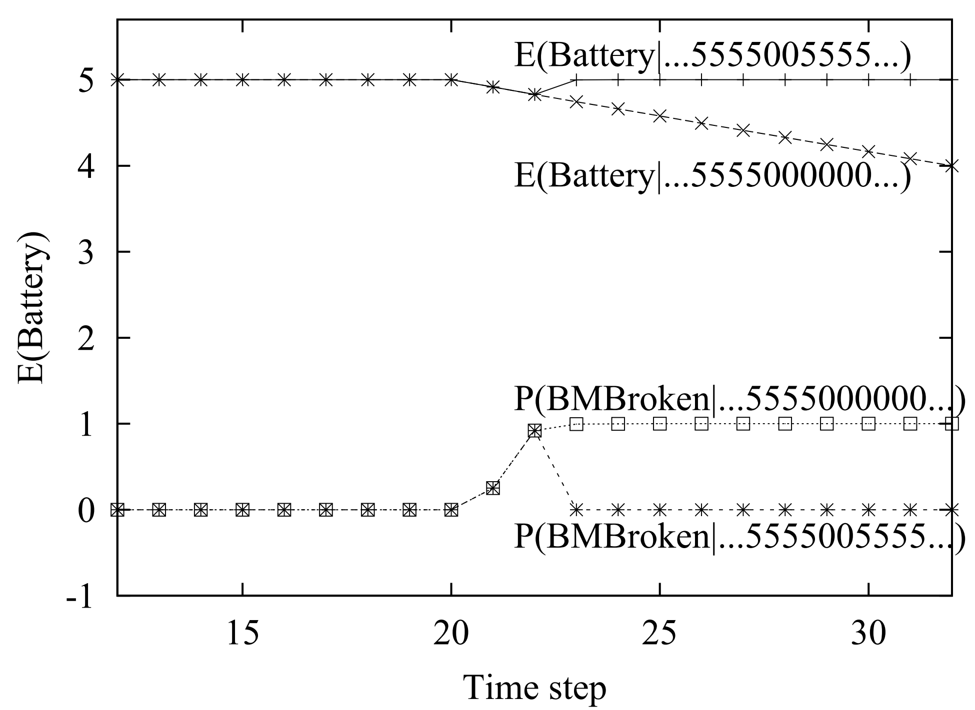
DBNs vs. Kalman Filters

هر مدل **فیلتر کالمن** یک **DBN** تک متغیره است.
اما **DBN**های کمی هستند که **فیلتر کالمن** باشند.

(دنیای واقعی به توزیع‌های پسین غیرگاوسی نیاز دارد)

مثال: میزان شارژ باتری چه‌قدر است؟



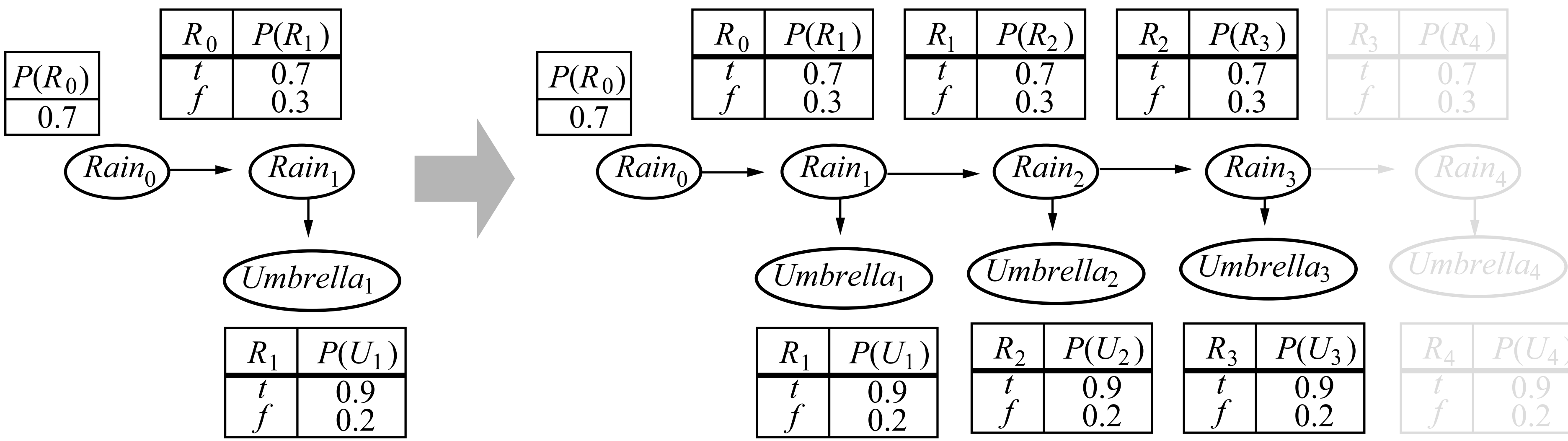


شبکه‌های بیزی پویا

استدلال دقیق در شبکه‌های بیزی پویا

EXACT INFERENCE IN DBNs

روش ساده: شبکه را باز کنید (unroll) و یک الگوریتم استدلال دقیق را اجرا کنید.



Problem: inference cost for each update grows with t

Rollup filtering: add slice $t+1$, "sum out" slice t using variable elimination

Largest factor is $O(d^{n+1})$, update cost $O(d^{n+2})$
(cf. HMM update cost $O(d^{2n})$)

Prepared by Kazim Fouladi | Spring 2017 | 2nd Edition

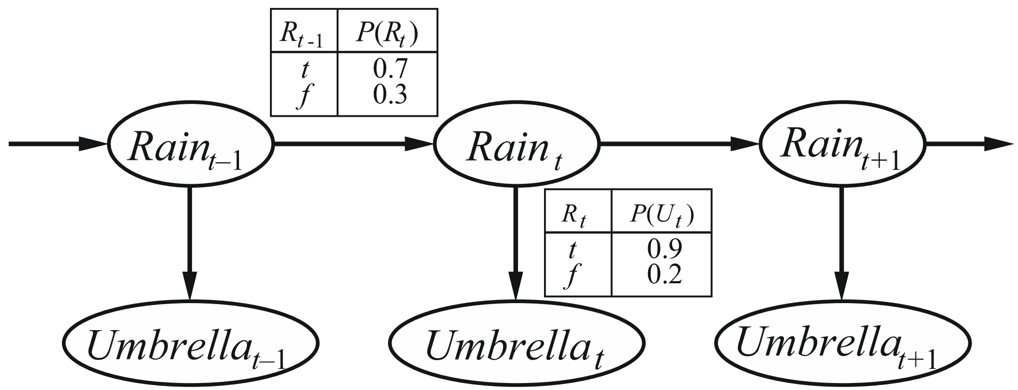
کاظم فولادی

شبکه‌های بیزی پویا

وزن‌دهی درست‌نمایی برای شبکه‌های بیزی پویا

Likelihood Weighting for DBNs

مجموعه‌ی نمونه‌های وزن‌دار، حالت باور را تقریب می‌زنند.



نمونه‌های «وزن‌دهی درست‌نمایی» به شواهد هیچ توجهی نمی‌کنند!

$\Leftarrow$ کسر «توافق» با گذر زمان به صورت نمایی افت می‌کند.

$\Leftarrow$ تعداد نمونه‌های لازم با گذر زمان به صورت نمایی رشد می‌کند.

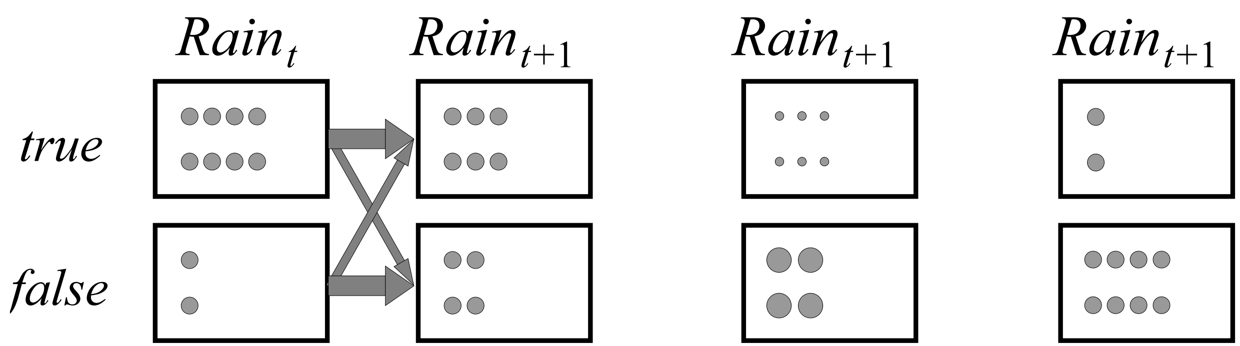
شبکه‌های بیزی پویا

فیلتر کردن ذره‌ای

PARTICLE FILTERING

ایده‌ی اصلی: اطمینان از اینکه جمعیت نمونه‌ها (ذرات: particles) ناحیه‌های دارای شانس بالا در فضای حالت را دنبال می‌کنند.

ذرات متناسب با درست‌نمایی برای $\mathbf{e}_t$ تکثیر می‌شوند.



(a) Propagate (b) Weight (c) Resample

مورد استفاده‌ی گسترده برای ردیابی سیستم‌های غیرخطی، بخصوص در بینایی
همچنین مورد استفاده در مکان‌یابی و نقشه‌برداری در ربات‌های متحرک (فضای حالت صدهزار بعدی)

PARTICLE FILTERING

Assume consistent at time t: $N(\mathbf{x}_t|\mathbf{e}_{1:t})/N = P(\mathbf{x}_t|\mathbf{e}_{1:t})$

Propagate forward: populations of $\mathbf{x}_{t+1}$ are

$$N(\mathbf{x}_{t+1}|\mathbf{e}_{1:t}) = \Sigma_{\mathbf{x}_t} P(\mathbf{x}_{t+1}|\mathbf{x}_t) N(\mathbf{x}_t|\mathbf{e}_{1:t})$$

Weight samples by their likelihood for $\mathbf{e}_{t+1}$:

$$W(\mathbf{x}_{t+1}|\mathbf{e}_{1:t+1}) = P(\mathbf{e}_{t+1}|\mathbf{x}_{t+1}) N(\mathbf{x}_{t+1}|\mathbf{e}_{1:t})$$

Resample to obtain populations proportional to W:

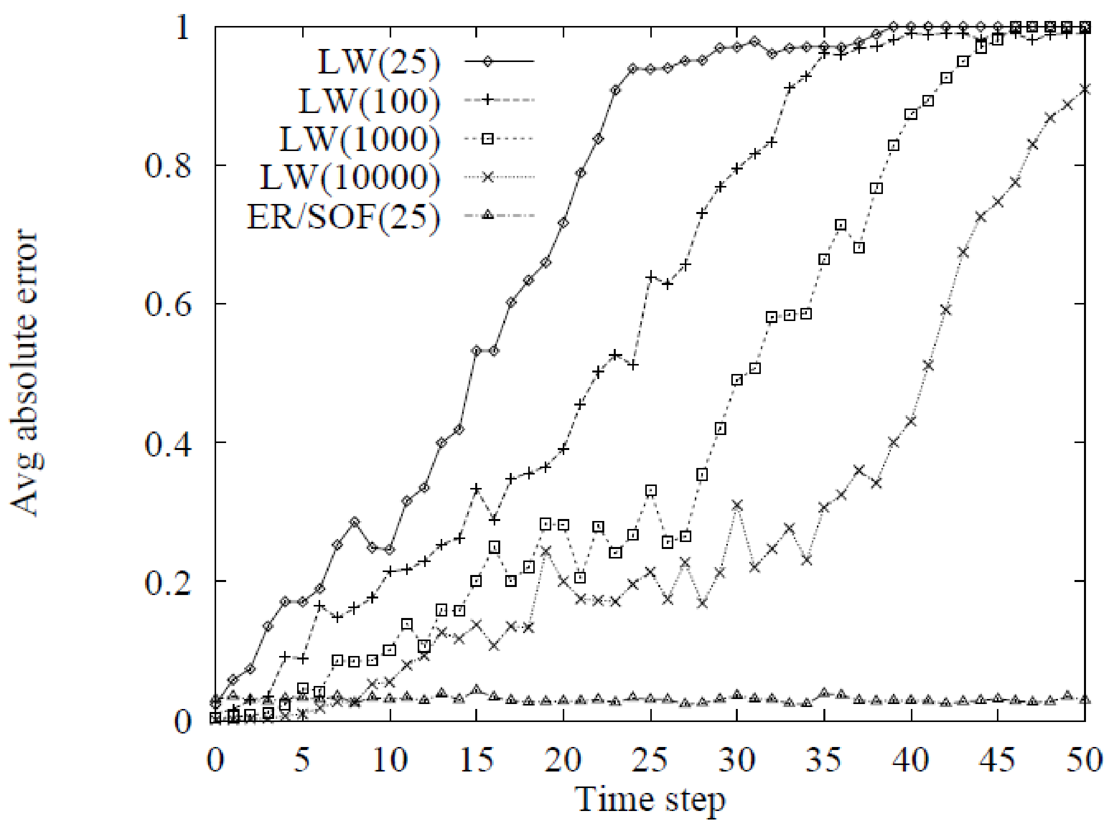
$$\begin{aligned}
N(\mathbf{x}_{t+1}|\mathbf{e}_{1:t+1})/N &= \alpha W(\mathbf{x}_{t+1}|\mathbf{e}_{1:t+1}) = \alpha P(\mathbf{e}_{t+1}|\mathbf{x}_{t+1}) N(\mathbf{x}_{t+1}|\mathbf{e}_{1:t}) \\
&= \alpha P(\mathbf{e}_{t+1}|\mathbf{x}_{t+1}) \Sigma_{\mathbf{x}_t} P(\mathbf{x}_{t+1}|\mathbf{x}_t) N(\mathbf{x}_t|\mathbf{e}_{1:t}) \\
&= \alpha' P(\mathbf{e}_{t+1}|\mathbf{x}_{t+1}) \Sigma_{\mathbf{x}_t} P(\mathbf{x}_{t+1}|\mathbf{x}_t) P(\mathbf{x}_t|\mathbf{e}_{1:t}) \\
&= P(\mathbf{x}_{t+1}|\mathbf{e}_{1:t+1})
\end{aligned}$$

شبکه‌های بیزی پویا

فیلتر کردن ذره‌ای : کارآیی

[Particle Filtering]

خطای تقریبی فیلترکردن ذره‌ای در طول زمان کران‌دار باقی می‌ماند
(حداقل به طور تجربی ــ تحلیل تئوری آن دشوار است)



هوش مصنوعی

استدلال احتمالاتی در امتداد زمان

۶

ردگیری اشیای متعدد

ردگیری اشیای متفاوت

مثال (دوربین‌های مراقبت ترافیکی)

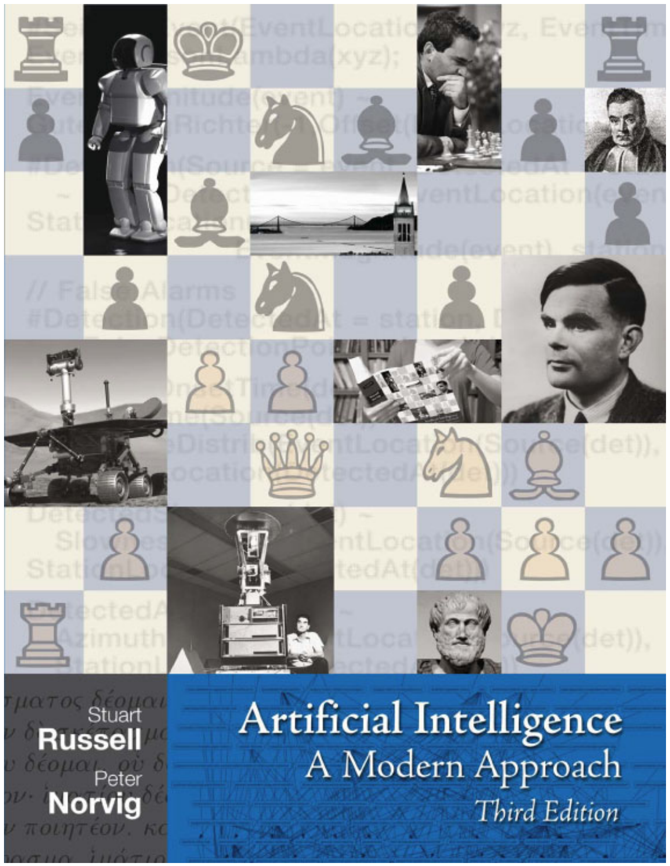
هوش مصنوعی

استدلال احتمالاتی در امتداد زمان

۷

منابع، مطالعه، تکلیف

منبع اصلی



Stuart Russell and Peter Norvig,
Artificial Intelligence: A Modern Approach,
3rd Edition, Prentice Hall, 2010.

Chapter 15

15 PROBABILISTIC REASONING OVER TIME

In which we try to interpret the present, understand the past, and perhaps predict the future, even when very little is crystal clear.

Agents in partially observable environments must be able to keep track of the current state, to the extent that their sensors allow. In Section 4.4 we showed a methodology for doing that: an agent maintains a **belief state** that represents which states of the world are currently possible. From the belief state and a **transition model**, the agent can predict how the world might evolve in the next time step. From the percepts observed and a **sensor model**, the agent can update the belief state. This is a pervasive idea: in Chapter 4 belief states were represented by explicitly enumerated sets of states, whereas in Chapters 7 and 11 they were represented by logical formulas. Those approaches defined belief states in terms of which world states were *possible*, but could say nothing about which states were *likely* or *unlikely*. In this chapter, we use probability theory to quantify the degree of belief in elements of the belief state.

As we show in Section 15.1, time itself is handled in the same way as in Chapter 7: a changing world is modeled using a variable for each aspect of the world state *at each point in time*. The transition and sensor models may be uncertain: the transition model describes the probability distribution of the variables at time t, given the state of the world at past times, while the sensor model describes the probability of each percept at time t, given the current state of the world. Section 15.2 defines the basic inference tasks and describes the general structure of inference algorithms for temporal models. Then we describe three specific kinds of models: **hidden Markov models**, **Kalman filters**, and **dynamic Bayesian networks** (which include hidden Markov models and Kalman filters as special cases). Finally, Section 15.6 examines the problems faced when keeping track of more than one thing.

15.1 TIME AND UNCERTAINTY

We have developed our techniques for probabilistic reasoning in the context of *static* worlds, in which each random variable has a single fixed value. For example, when repairing a car, we assume that whatever is broken remains broken during the process of diagnosis; our job is to infer the state of the car from observed evidence, which also remains fixed.

566

خلاصه

SUMMARY

Temporal models use state and sensor variables replicated over time

Markov assumptions and stationarity assumption, so we need
- transition model $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$
- sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$

Tasks are filtering, prediction, smoothing, most likely sequence;
all done recursively with constant cost per time step

Hidden Markov models have a single discrete state variable; used for speech recognition

Kalman filters allow n state variables, linear Gaussian, $O(n^3)$ update

Dynamic Bayes nets subsume HMMs, Kalman filters; exact update intractable

Particle filtering is a good approximate filtering algorithm for DBNs