Севастопольский государственный университет
Институт информационных технологий

Методы и системы искусственного интеллекта

Бондарев Владимир Николаевич

Предсказание состояния по свидетельствам

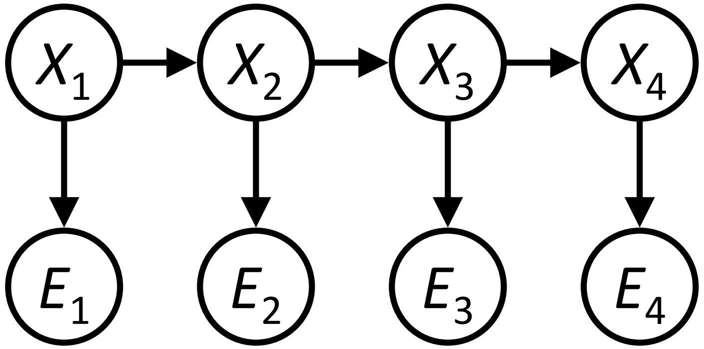
- В каждый момент времени мы получаем свидетельства и мы хотим знать

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

- Идея: начать с $P(X_1)$ и получить B_t с помощью B_{t-1}
  - Эквивалентно: получить B_{t+1} с помощью B_t

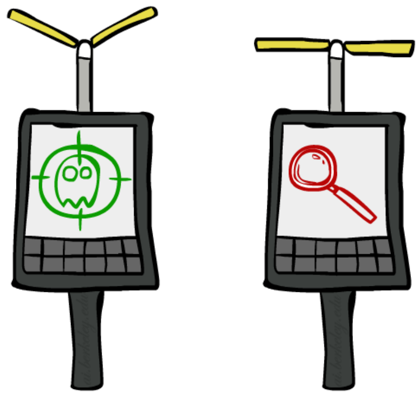
Два шага: переход во времени и наблюдение

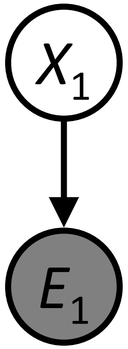
$B(X_t) = P(X_t|e_{1:t})$ $B'(X_{t+1})$

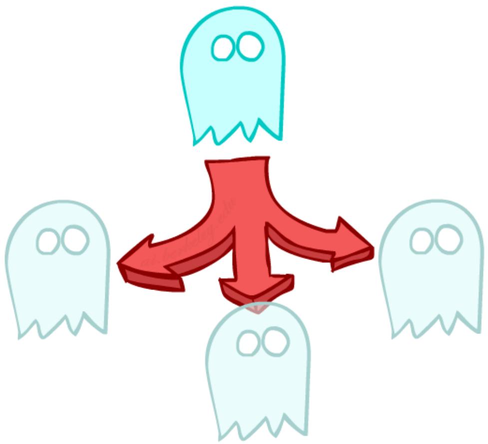


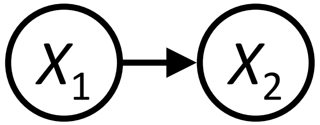
$B(X_{t+1})$

Вывод: базовые случаи









$$P(X_1|e_1)$$

$$P(X_1|e_1) = \frac{P(X_1, e_1)}{\sum_{x_1} P(x_1, e_1)}$$

$$P(X_1|e_1) = \frac{P(e_1|X_1)P(X_1)}{\sum_{x_1} P(e_1|x_1)P(x_1)}$$

$$P(X_2)$$

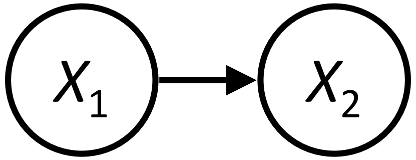
$$P(X_2) = \sum_{x_1} P(x_1, X_2)$$

$$P(X_2) = \sum_{x_1} P(X_2|x_1)P(x_1)$$

Переход во времени

- Пусть у нас имеется текущее доверительное состояние - *P*(*X* | *набор свидетельств*):

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



- Тогда, после одного шага во времени:

$$P(X_{t+1}|e_{1:t}) = \sum_{x_t} P(X_{t+1}, x_t|e_{1:t})$$

$$= \sum_{x_t} P(X_{t+1}|x_t, e_{1:t})P(x_t|e_{1:t})$$

$$= \sum_{x_t} P(X_{t+1}|x_t)P(x_t|e_{1:t})$$

  - Или компактно:

$$B'(X_{t+1}) = \sum_{x_t} P(X'|x_t)B(x_t)$$

- Базовая идея: доверия “проталкиваются” с помощью модели переходов:
  - С обозначением «*B*» мы должны быть осторожны и помнить к какому временному шагу оно относится и какие свидетельства оно включает.

Пример: переход во времени

- неопределённость “нарастает” при переходе по времени

(Модель перехода: призрак обычно совершает переход по часовой стрелке)

<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	1.00	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

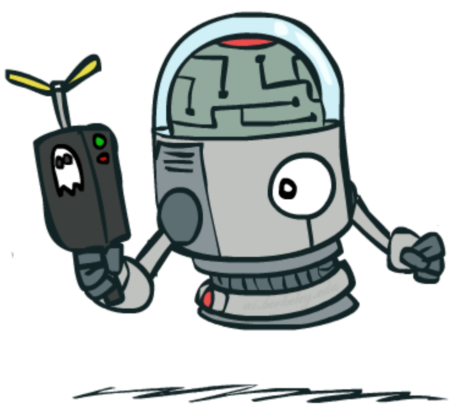
T = 1

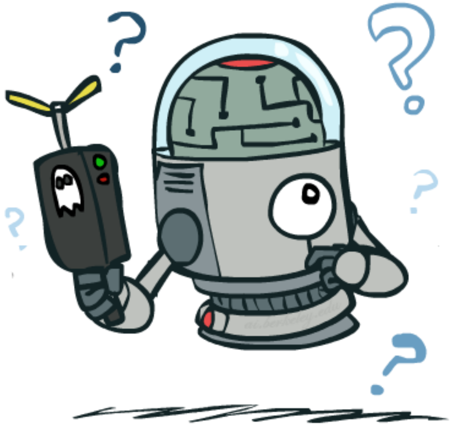
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01
<0.01	0.76	0.06	0.06	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01

T = 2

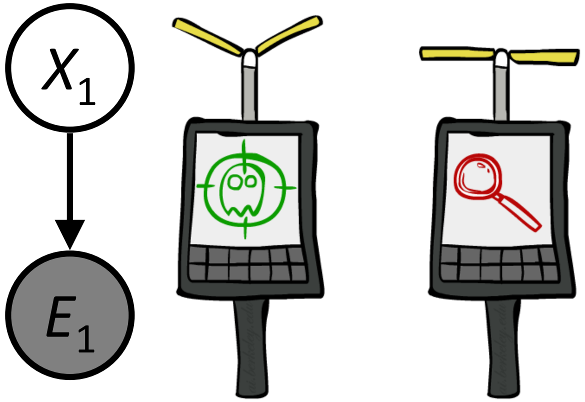
0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

T = 5





Наблюдение



- Пусть имеется текущее доверие *P*(*X* | *пред. свидетельства*):

$$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$

- Тогда, после поступления нового свидетельства :

$$P(X_{t+1}|e_{1:t+1}) = P(X_{t+1}, e_{t+1}|e_{1:t})/P(e_{t+1}|e_{1:t})$$

$$\propto_{X_{t+1}} P(X_{t+1}, e_{t+1}|e_{1:t})$$

$$= P(e_{t+1}|e_{1:t}, X_{t+1})P(X_{t+1}|e_{1:t})$$

$$= P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t})$$

- Или компактно:

$$B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1}|X_{t+1})B'(X_{t+1})$$

- Вывод: доверия “взвешиваются” правдоподобием свидетельства
- В отличие от перехода во времени здесь необходима нормализация

Пример: наблюдение

- По мере того, как мы получаем наблюдения, доверия переоцениваются, неопределенность «уменьшается».

0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

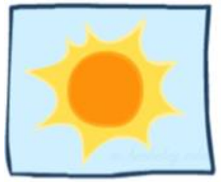
До наблюдения

<0.01	<0.01	<0.01	<0.01	0.02	<0.01
<0.01	<0.01	<0.01	0.83	0.02	<0.01
<0.01	<0.01	0.11	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

После наблюдения

$$B(X) \propto P(e|X)B'(X)$$

Пример: HMM оценки погоды



B(+r) = 0.5
B(-r) = 0.5

B’(+r) = 0.5
B’(-r) = 0.5

B(+r) = 0.818
B(-r) = 0.182

B’(+r) = 0.627
B’(-r) = 0.373

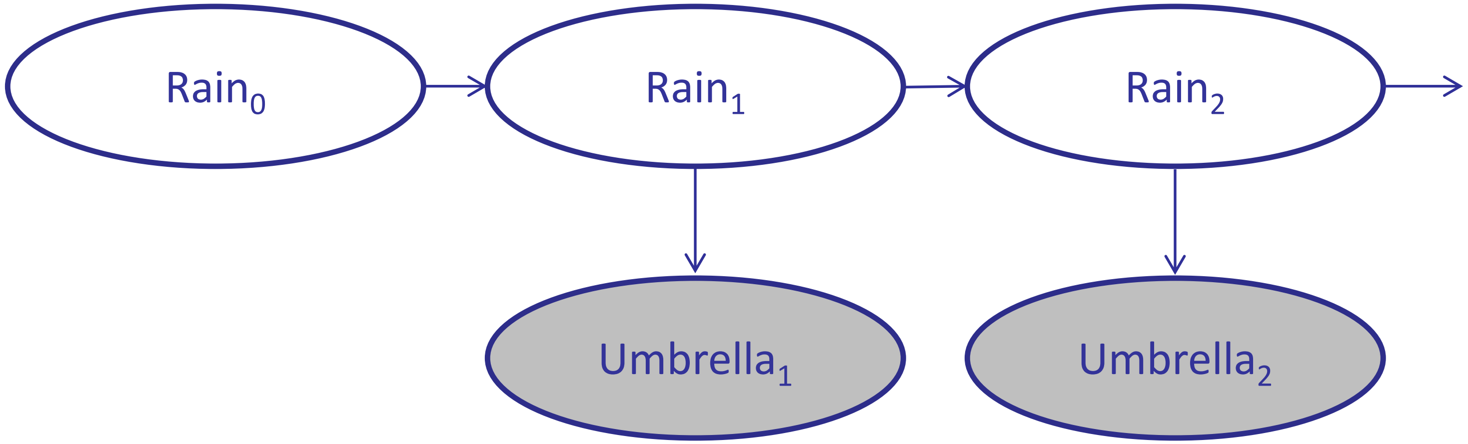
B(+r) = 0.883
B(-r) = 0.117

$$\mathbf{P}(R_1) = \sum_{r_0} \mathbf{P}(R_1 | r_0) P(r_0) =$$

= <0.7, 0.3> * 0.5+<0.3, 0.7> * 0.5 = <0.5, 0.5>

$$\mathbf{P}(R_1 | u_1) = \alpha\ \mathbf{P}(u_1 | R_1)\mathbf{P}(R_1) =$$

= α <0.9,0.2 >* <0.5,0.5>= α <0.45/0.1> =<0.818, 0.182>



u_1 =True

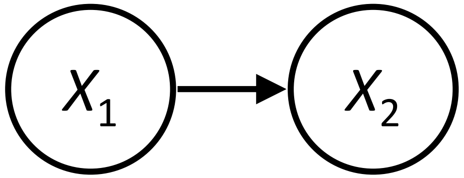
u_2 =True

R_t	R_{t+1}	$P(R_{t+1} \mid R_t)$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

R_t	U_t	$P(U_t \mid R_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

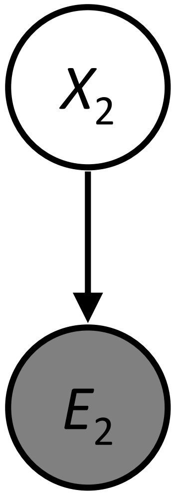
Он-лайн обновление доверий

- На каждом шаге по времени мы начинаем с текущего доверия *P*(*X* | *свидетельства*)
- Выполняем обновление предсказания во времени:



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

- Обновляем с учетом нового свидетельства:



$$P(x_t|e_{1:t}) \propto_X P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$

- Алгоритм прямого распространения выполняет оба действия одновременно (при этом он не выполняет нормализацию на каждом шаге)

Алгоритм прямого распространения

- В каждый момент времени у нас имеются свидетельства и мы хотим определить:

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

- Можно вывести следующее выражение:

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

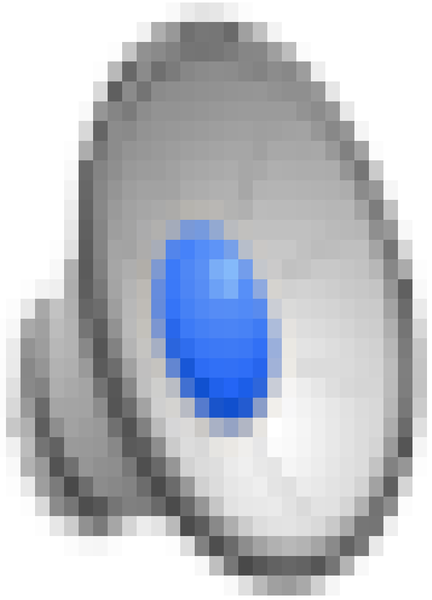
Нормализовать можно по мере вычислений, если хотим иметь P(x|e) на каждом временном шаге или только один раз в конце…

Вместо 2-х отдельных обновлений можно выполнять одно обновление. При этом нормализация (деление на P(e1:t)) может быть выполнена в конце.

Алгоритм напоминает последовательное исключение переменных X1, X2,….Xt

Пакман – Сонар

Демо: Пакман – Сонар (с довериями)

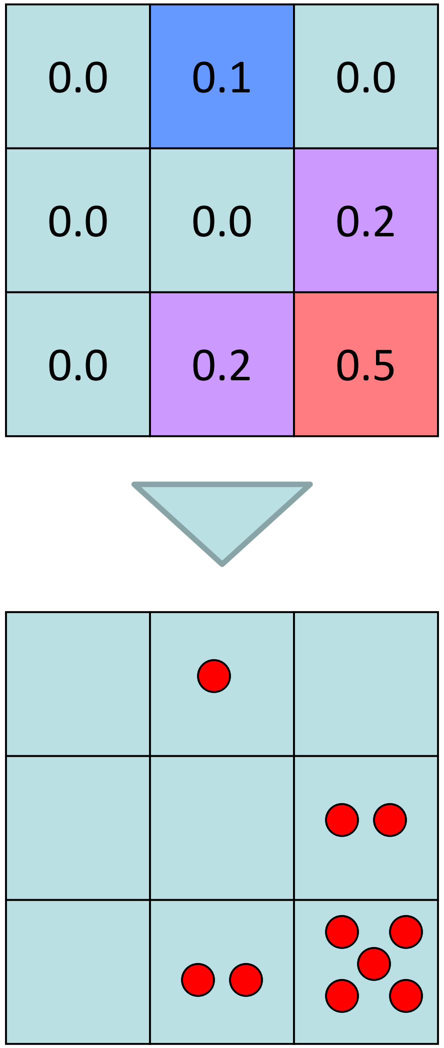


Фильтрация частиц

Фильтрация частиц

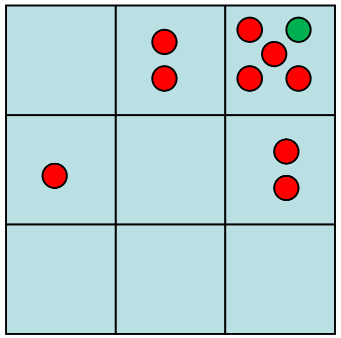
- Точный вывод с использованием алгоритма прямого распространения СММ при большой мощности |X| характеризуется вычислительной сложностью. В этом случае, аналогично сетям Байеса, используют приближенные методы вывода, основанные на формировании случайных выборок из распределений.

- Применение к СММ процедур, аналогичных байесовскому сэмплированию (взятию выборок), называется **фильтрацией частиц** и включает в себя моделирование движения набора частиц через граф состояний для аппроксимации распределения вероятности (доверий) рассматриваемой случайной переменной в требуемый момент времени.



Представление: фильтрация частиц

Частица представляет возможное выборочное значение случайной величины. При этом вместо хранения полных таблиц вероятностей, отображающих каждое состояние в вероятность, хранят **список из *n* частиц**, в котором каждая частица может находиться в одном из *d* состояний. Чем больше частиц, тем выше точность аппроксимации.



На рисунке изображен **список частиц**, представленных координатами клеток, в которых они расположены. Соответственно, для частиц с координатами (3,3) эмпирическая оценка вероятности появления в списке частиц равна $B(3,3)=5/10=0.5$. Таким образом, по списку частиц можно восстановить эмпирическое распределение случайной переменной.

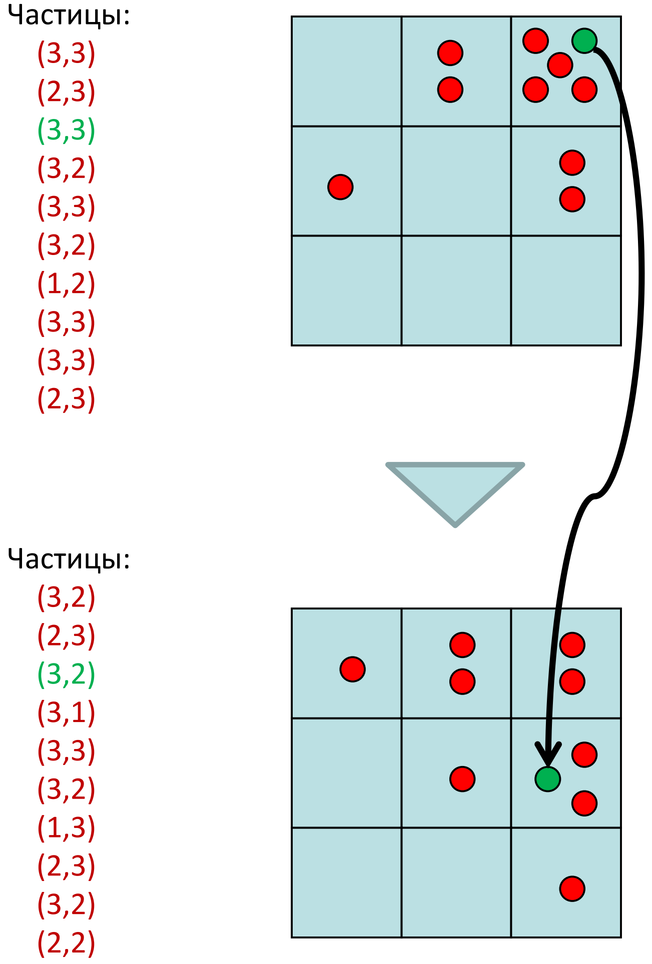
Список частиц:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)

Фильтрация частиц: обновление во времени

Моделирование фильтрации частиц начинается с инициализации частиц. Например, можно выбрать положения частиц случайным образом из некоторого начального распределения. После того, как выбран исходный список частиц, моделирование принимает форму, аналогичную алгоритму прямого распространения в СММ с чередованием обновления распределений во времени и обновления на основе наблюдения на каждом временном шаге.

Обновление во времени (Time Elapse Update).
Для j-ой частицы в состоянии x_t выполняется случайная выборка обновленного значения из переходного распределения $P(X_{t+1} \mid x_t)$.

$$x_{t+1}^{(j)} \sim P(X_{t+1} \mid x_t^{(j)})$$



Фильтрация частиц: обновление во времени

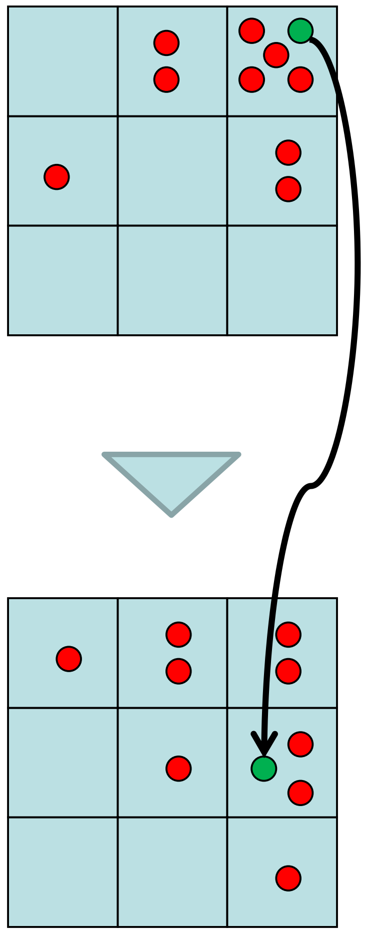
- В ходе сэмплирования каждая частица перемещается в следующую позицию в соответствии с моделью перехода:

$$x' = \text{sample}(P(X'|x))$$

  - Это аналогично выборке из априорных распределений — частоты выборок отражают вероятности перехода;
  - В примере частицы перемещаются по часовой стрелке, но возможны случайные перемещения в других направлениях или пребывание на месте.

- Процесс соответствует переходу во времени:
  - Если выборок много, то получаются частоты близкие к точным значениям вероятностей (согласованность)

Частицы:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)



Частицы:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)

Фильтрация частиц: наблюдение

Обновление на основе наблюдения (Observation Update). Этот этап сложнее. Здесь используется модель сенсора $P(E_t|X_t)$ для взвешивания каждой частицы в соответствии с вероятностью, определяемой наблюдаемым свидетельством и состоянием частицы. В частности, частице в состоянии x_t при свидетельстве e_t, поступающим от некоторого сенсора, присваивается вес $P(e_t \mid x_t)$.

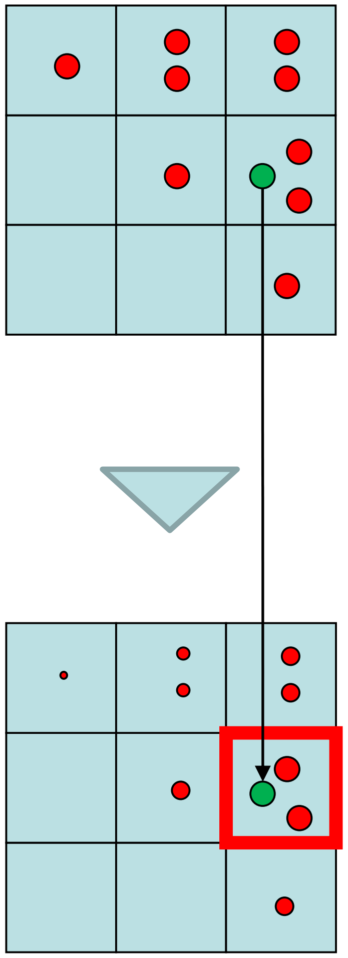
- Т.о., частицы взвешиваются с весом, учитывающим свидетельства:

$$w(x) = P(e|x)$$

$$B(X) \propto P(e|X)B'(X)$$

- Как и раньше, вероятности в сумме не равны единице, так как все они были снижены весом (на самом деле теперь они в сумме аппроксимируют P(e))

Частицы:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)

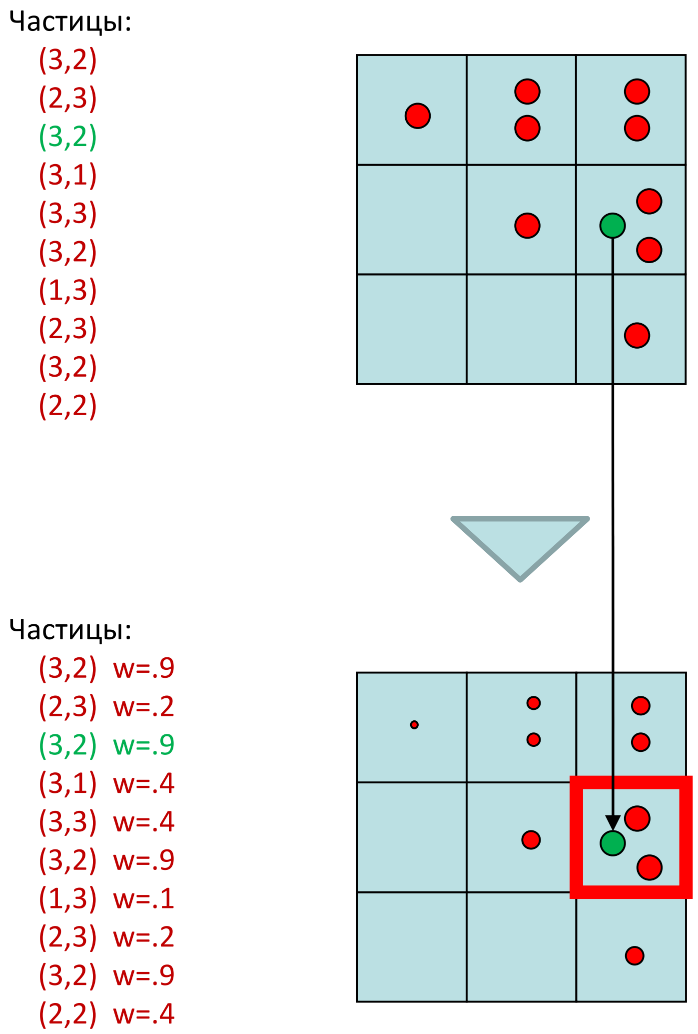


Частицы:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

Фильтрация частиц: наблюдение

Алгоритм обновления на основе наблюдений следующий:

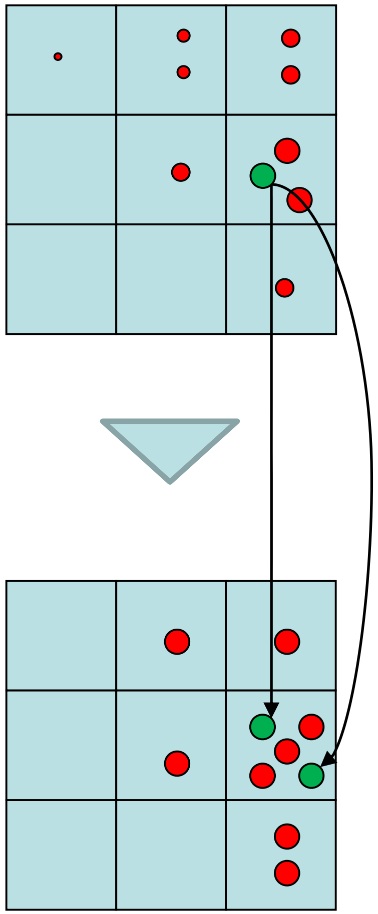
1. Рассчитайте веса всех частиц, как указано выше;
2. Вычислите суммарный вес каждого состояния;
3. Если сумма всех весов во всех состояниях равна 0, повторно инициализируйте все частицы;
4. Иначе нормализуйте распределение по отношению к суммарному весу и выполните выборки из этого распределения (ресэплирование).



Фильтрация частиц: ресэмплирование

- Вместо того, чтобы выполнять оценку взвешенных выборок, мы выполняем очередное **сэмплирование**
- Делаем N выборок из взвешенного выборочного распределения и получаем новый список частиц (делаем выборку с замещением частиц)
- Это эквивалентно ренормализации распределения
- Обновление для текущего временного шага завершается и можно перейти к следующему шагу.

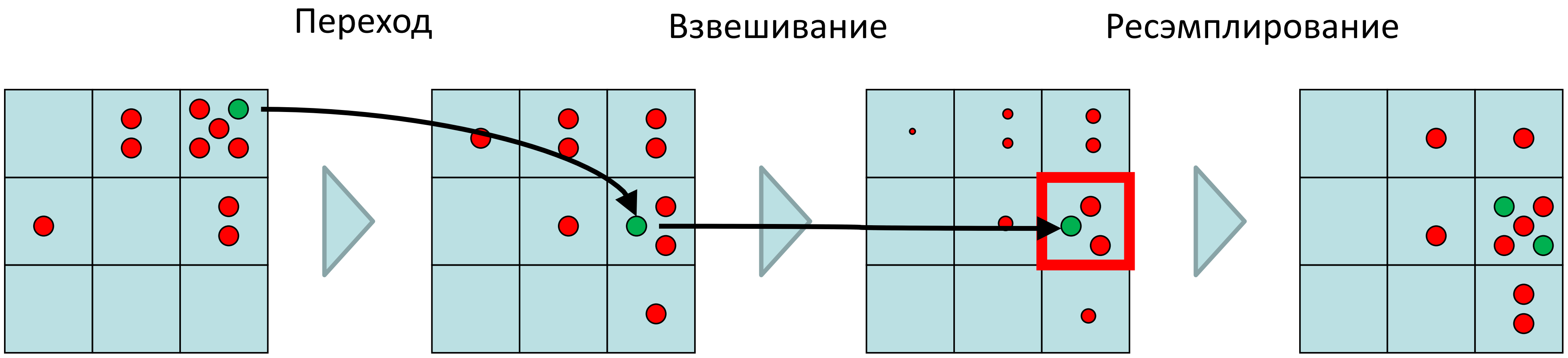
Частицы:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4



Новые частицы:
(3,2)
(2,2)
(3,2)
(2,3)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(3,2)

Резюме: фильтрация частиц

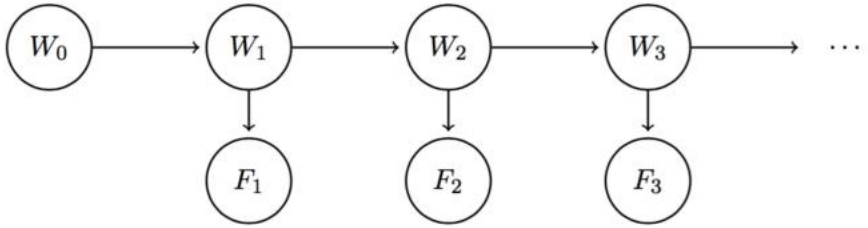
- Вычисляются выборки из состояний, а не явные распределения



Частицы:	Частицы:	Частицы	Новые частицы:
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,3)	(2,3) w=.2	(2,2)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(3,2)	(3,1)	(3,1) w=.4	(2,3)
(3,3)	(3,3)	(3,3) w=.4	(3,3)
(3,2)	(3,2)	(3,2) w=.9	(3,2)
(1,2)	(1,3)	(1,3) w=.1	(1,3)
(3,3)	(2,3)	(2,3) w=.2	(2,3)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,2)	(2,2) w=.4	(3,2)

Пример: фильтрация частиц

- Пример моделирования погоды.



- Пусть, например, имеется список из 10 частиц со следующими значениями температуры:

  [15, 12, 12, 10, 18, 14, 12, 11, 11, 10].

- Соответственно, подсчитав количество различных состояний частиц и поделив эти значения на общее число частиц, получим эмпирическое распределение температуры в момент времени *i:*

T_i	10	11	12	13	14	15	16	17	18	19	20
$B(T_i)$	0.2	0.2	0.3	0	0.1	0.1	0	0	0.1	0	0

Пример: фильтрация частиц

- Определим модель перехода – модель смены температуры: *температура может либо остаться прежней, либо измениться на один градус в диапазоне [10, 20]. При этом вероятность перехода в следующий момент времени к значению, которое ближе к T=15, составляет 0.8, а остальные результирующие состояния равномерно делят оставшиеся 0.2 вероятности между собой.*

  Например, пусть для первой частицы из списка частиц (T_i = 15) модель перехода задается таблицей

T_{i+1}	14	15	16
$P(T_{i+1} \mid T_i = 15)$	0.1	0.8	0.1

- Для формирования выборки для частицы в состоянии T_i = 15 сгенерируем случайное число с распределением, соответствующим модели перехода, в диапазоне [0, 1) и смотрим, в какой диапазон оно попадает. Если случайное число равно r = 0.467, то частица с T_i = 15 попадает в диапазон 0,1 ≤ r < 0,9. Следовательно, в следующий момент времени с учетом таблицы переходных вероятностей частица будет в состоянии T_{i+1} = 15.

Пример: фильтрация частиц

- Допустим, таким способом мы получили список из 10 случайных чисел в интервале [0, 1):

  [0.467, 0.452, 0.583, 0.604, 0.748, 0.932, 0.609, 0.372, 0.402, 0.026]

  С учетом указанной модели перехода и используя эти 10 случайных чисел для формирования выборочных значений наших 10 частиц, получим после полного обновления во времени новый список частиц:

  [15, 13, 13, 11, 17, 15, 13, 12, 12, 10].

- Обновленный список частиц приводит к соответствующему обновленному распределению степеней уверенности $B'(T_{i+1})$:

T_{i+1}	10	11	12	13	14	15	16	17	18	19	20
$B'(T_{i+1})$	0.1	0.1	0.2	0.3	0	0.2	0	0.1	0	0	0

Пример: фильтрация частиц

Теперь выполним обновление на основе наблюдения, предполагая, что сенсорная модель $Pr(F_i|T_i)$ утверждает, что вероятность правильного прогноза $f_i = t_i$ равна 0.8, а остальные 10 возможных состояний предсказываются с вероятностью 0.02. Если наблюдаемый прогноз F_{i+1} = 13, то веса частиц будут следующими:

Частица	1	2	3	4	5	6	7	8	9	10
Состояние	15	13	13	11	17	15	13	12	12	10
Вес	0.02	0.8	0.8	0.02	0.02	0.02	0.8	0.02	0.02	0.02

- После суммирования весов каждого из состояний, получим

Состояние	10	11	12	13	15	17
Вес	0.02	0.02	0.04	2.4	0.04	0.02

- Суммирование значений всех весов дает сумму 2.54, после нормализации

Состояние	10	11	12	13	15	17
Вес	0.02	0.02	0.04	2.4	0.04	0.02
Нормализованный вес	0.079	0.079	0.0157	0.9449	0.0157	0.079

Пример: фильтрация частиц

- Последним шагом является повторная выборка (ресэмплирование) из этого распределения вероятностей с использованием того же метода, который мы использовали для выборки во время обновления во времени. Допустим, мы генерируем 10 случайных чисел в диапазоне [0,1) со следующими значениями:

  [0.315, 0.829, 0.304, 0.368, 0.459, 0.891, 0.282, 0.980, 0.898, 0.341]

  Это дает новый (ресэмплированный) список частиц

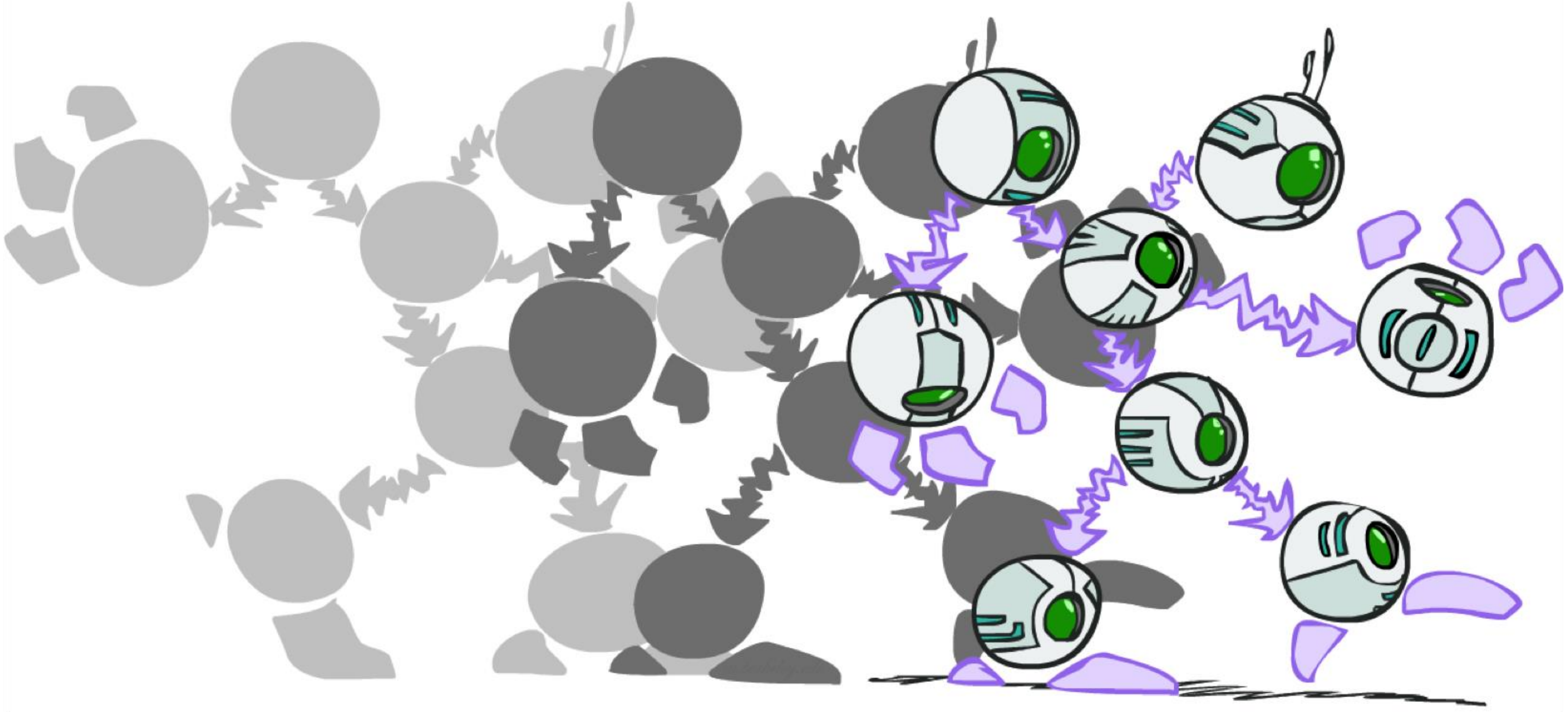
  [13, 13, 13, 13, 13, 13, 13, 15, 13, 13]

  с новым распределением степеней уверенности:

T_{i+1}	10	11	12	13	14	15	16	17	18	19	20
$B(T_{i+1})$	0	0	0	0.9	0	0.1	0	0	0	0	0

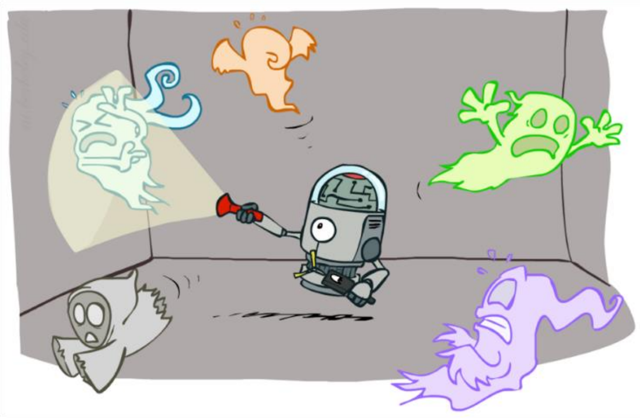
- Обратите внимание, что сенсорная модель предполагает, что наш прогноз погоды весьма точен и характеризуется высокой вероятностью, равной 0,8. Соответственно, наш новый список частиц согласуется с этим: большинство частиц в результате ресэмплирования получают состояние T_{i+1} = 13.

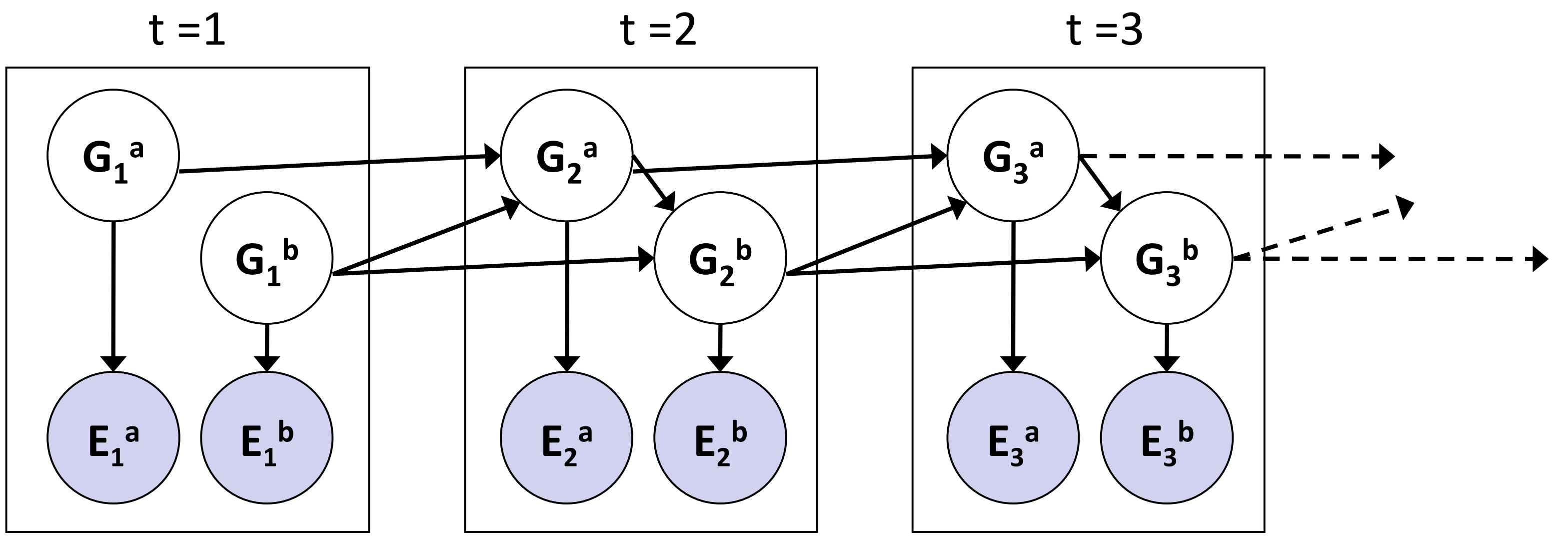
Динамические байесовские сети (ДБС)
Dynamic Bayes Nets (DBNs)



Динамические байесовские сети

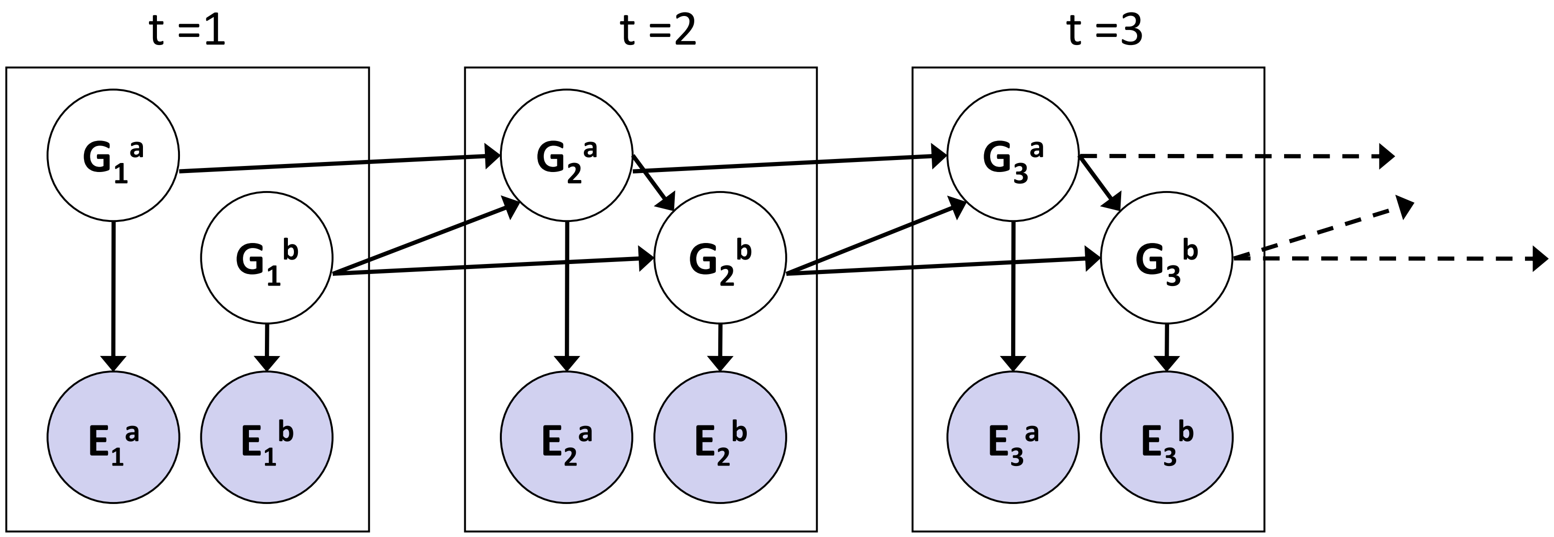
- Каждый временной срез сети DBN может иметь **любое количество переменных состояния и переменных свидетельства.** Переменные и связи между ними точно тиражируются от среза к срезу и сеть DBN представляет марковский процесс первого порядка, так что каждая переменная может иметь родительские переменные только в своем собственном временном срезе или в непосредственно предшествующем временном срезе.





- DBN являются обобщением HMM

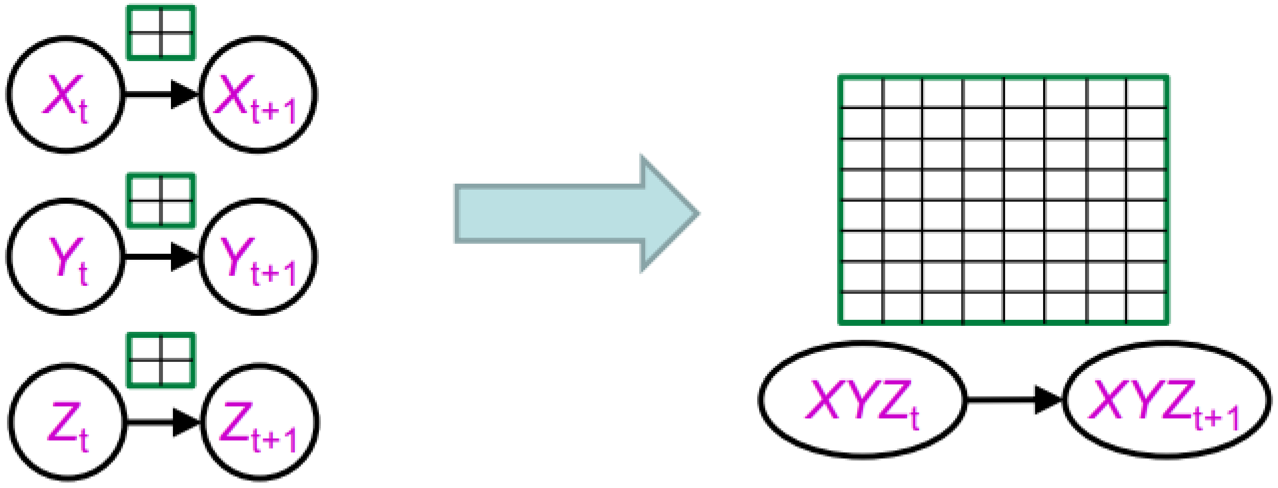
Динамические байесовские сети (DBN)

- Например, в Пакман имеется несколько призраков. Поскольку модели перехода призраков больше не являются независимыми, все призраки должны отслеживаться совместно с использованием динамической сети Байеса, которая является обобщением СММ.
- ДБС с двумя призраками (*a* и *b*) изображена на рисунке. Скрытые переменные *G* представляют положения призраков, а переменные свидетельств *E* представляют собой зашумленные расстояния до каждого из призраков. Представленную структуру ДБС можно распространить на большее количество призраков.
- 

- DBN являются обобщением HMM

ДБС и СММ

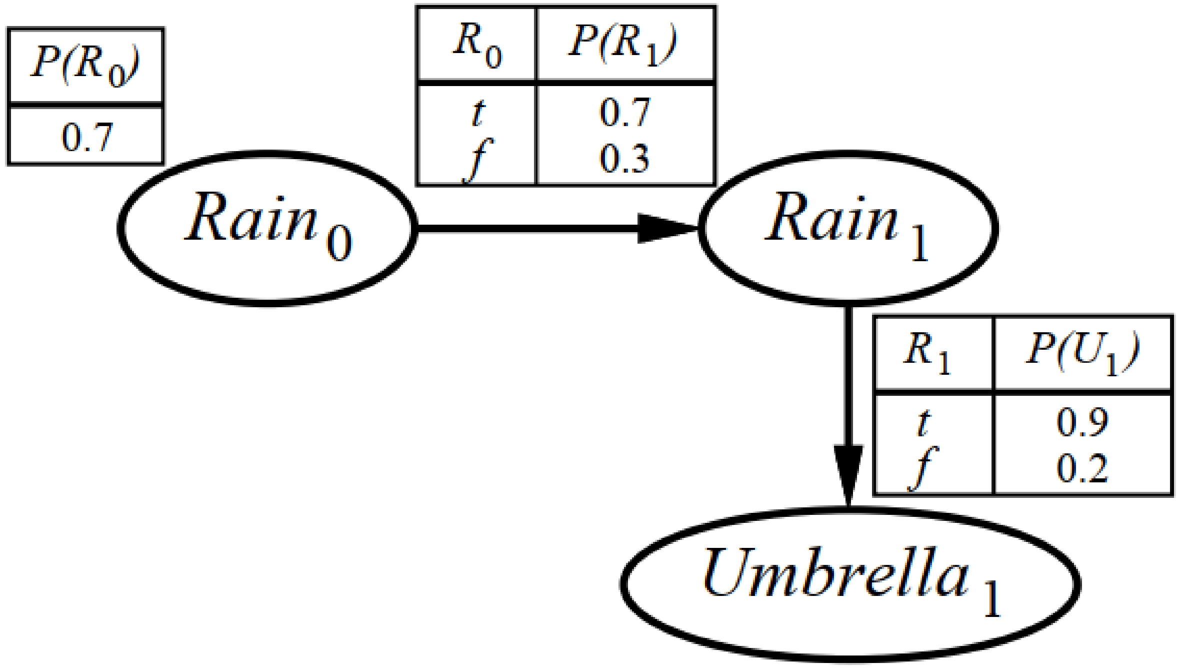
- Каждая СММ – это ДБС с одной переменной;
- Каждая дискретная ДБС – это СММ;
- Состояние СММ – это прямое (декартово) произведение множеств состояний ДБС ;
-



- Благодаря декомпозиции состояния сложной системы на составляющие его переменные сеть ДБС позволяет воспользоваться преимуществами разреженности временной вероятностной модели;
- Например, 20 бинарных переменных, 3 родителя у каждой:
  ДБС будет иметь 20x2^3=160 параметров, СММ имеет (20^20)x(20^20) ~ 10^12 параметров.

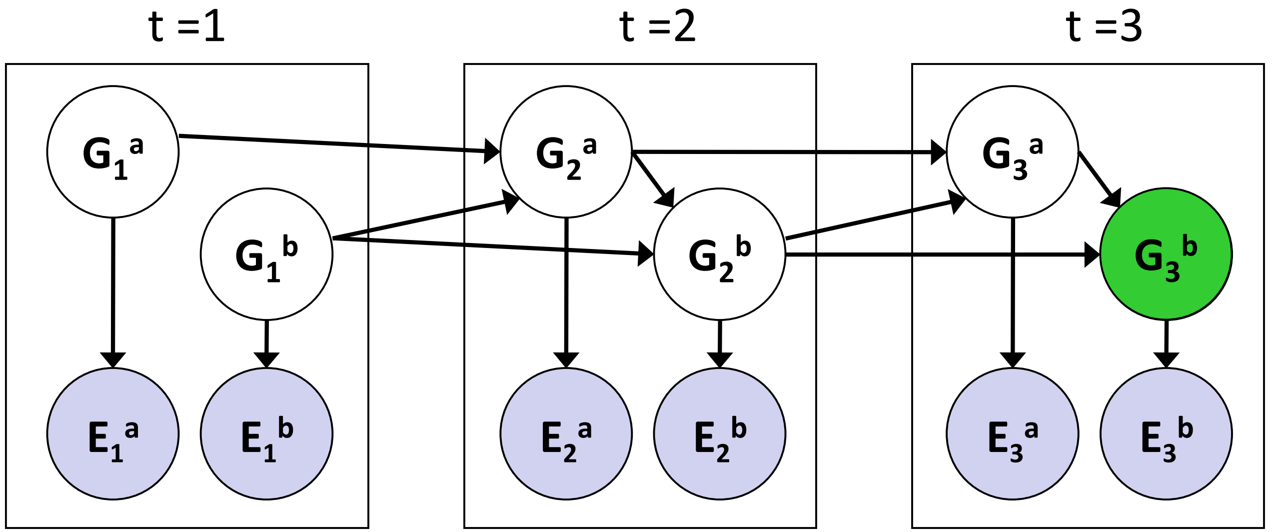
Процедура создания сетей DBN

Чтобы создать ДБС необходимо определить **$P(X_0)$, $P(X_t \mid X_{t-1})$, $P(E_t \mid X_t)$.** Для задания модели перехода и модели восприятия, необходимо определить топологию связей между последовательными срезами, а также между переменными состояния и свидетельствами. Поскольку предполагается, что модели перехода и восприятия являются стационарными (одинаковыми для всех t), удобнее всего задать их для первого среза. На основании этой спецификации по пере необходимости может быть создана полная (полубесконечная) сеть ДБС путем копирования первого среза.
Например, полная спецификация ДБС для мира с зонтиком представляется в виде сети с тремя узлами:



Точный вывод в DBN

- К DBN можно, например, применить процедуру исключения переменных
- Процедура: развернуть сеть на T временных шагов, затем исключать переменные , в процессе вычисления $P(X_T|e_{1:T})$



- Он-лайн обновление доверий: Исключать все переменные предыдущего временного шага; сохранять только факторы текущего времени;
- Проблема: наибольший фактор содержит все переменные для текущего времени

DBN фильтрация частиц

- Частица – полная выборка на временном шаге
- **Инициализация**: Сгенерировать априорные выборки сети Байеса для $t=1$
  - Пример, частицы: $\mathbf{G_1^a} = (3,3)$ $\mathbf{G_1^b} = (5,3)$
- **Переход по времени**: Сделать выборку приемника каждой частицы
  - Пример приемника: $\mathbf{G_2^a} = (2,3)$ $\mathbf{G_2^b} = (6,3)$
- **Наблюдение**: Взвесить каждую полную выборку с учетом правдоподобия свидетельства обусловленного выборкой
  - Правдоподобие: $P(\mathbf{E_1^a} \mid \mathbf{G_1^a}) * P(\mathbf{E_1^b} \mid \mathbf{G_1^b})$
- **Ресэмплирование:** Взять априорные выборки (кортежи значений) пропорционально их правдоподобиям (весам)