

# Before we can find a model, we must forget about perfection

Dimiter Dobrev  
Institute of Mathematics and Informatics  
Bulgarian Academy of Sciences  
*d@dobrev.com*

In Reinforcement Learning we assume that a model of the world exists. We assume that this model is perfect (that is, it describes the world completely and unambiguously). In this article we will show that it makes no sense to look for the perfect model, because this model is too complex and practically cannot be found. We will show that we must forget about perfection and look for event-driven models instead. These models are a generalization of the Markov decision process (MDP) models. This generalization is very important, because without it nothing can be found. Instead of one perfect MDP model, we will be looking for a large number of simple event-driven models, each of which describes some simple dependency or property. That is, we will replace the search for one complicated perfect model with the search for a large number of simple models.

**Keywords:** Artificial Intelligence, Reinforcement Learning, Partial Observability, Event-Driven Model, Definition of Object.

## Въведение

Когато разучаваме непознат град, ние се опитваме да нарисуваме картата на този град. Когато искаме да разберем непознат свят, ние се опитваме да построим модел на тази свят. Моделът е нещо като карта на света и ще представлява един ориентиран граф.

Какъв модел на света търсим? Дали търсим генератор или дескриптор?

Ако търсим генератор, то той трябва да е съвършен и да описва света напълно. За да създадем света (да го генерираме) на нас ни е нужно пълното му описание. Да, но на нас не ни е нужно да създаваме света, защото той вече е създаден и ние трябва само да го разберем.

Ние ще се опитаме да намерим дескриптор, който описва света частично, като казва нещо, без да каже всичко. Светът не може да бъде разбран напълно, ако е достатъчно сложен, затова ще се откажем от намирането на генератор (пълното описание на света) и ще търсим дескриптор (някакво частично описание).

В тази статия ще предпологаме, че съществува модел-генератор, който описва света напълно. Ние ще използваме този модел-генератор с теоретична цел, като чрез него ще дефинираме понятието събитие. Ще използваме този модел, но няма да го търсим, защото ще предпологаме, че този модел е твърде сложен, за да бъде намерен. Няма да търсим този модел, а ще търсим някакви по-прости модели, всеки от които ще описва някаква зависимост или някакво свойство. Тези модели ще ги наречем event-driven модели.

И ние, както повечето автори, ще предполагаме, че моделът-генератор на света е Markov decision process (MDP). Що се отнася до event-driven моделите, те са обобщение на MDP, където вместо действия имаме произволни събития.

Първият опит за въвеждане на event-driven модели, доколкото ни е известно, е в статиите [1] и [2]. Все пак, подхода в тази статия е по-различен от подхода в [1, 2], защото там авторите използват събитията, за да търсят стратегия, докато тук ние ги използваме, за да търсим модел.

Ние ще махнем от MDP нещо, което за момента не ни трябва. Ще махнем целта. Тоест, ще махнем rewards и discount factor. Това, което ще се получи е Reinforcement Learning без reinforcement. Махаме целта, защото в тази статия търсим модел на света, а не търсим стратегия. Тоест, задаваме си въпроса „Какво става?“, а не „Какво да правя?“. Целта е нужна за отговора на втория въпрос, но не и на първия.

Ние ще дефинираме event-driven моделите като обобщение на MDP моделите. За целта ще започнем от най-простия модел и ще го обобщим на няколко стъпки, като минем през MDP моделите и стигнем до event-driven моделите.

Възможно най-простият модел, от който ще почнем, ще бъде Fully observable Markov model (FOMM). Това е най-простият вариант на Markov chain. За FOMM ще видим как той може да предсказва миналото. Ще видим, че за всеки свят може да се намери стандартен FOMM, който върши някаква работа, макар че в общия случай този модел няма да е свършен.

Следващото, което ще направим, е да покажем, че агента се нуждае от динамична памет и че Fully observable model означава модел без памет. Това ще е нашето основание да забравим за Fully observable models. Ние ще направим първото наше обобщение и ще преминем към Partially observable Markov model, който е известен в литературата като Hidden Markov model.

Ще дадем формална дефиниция на понятията факт и събитие. Ще ги дефинираме като подмножество от състоянията и подмножество от стрелките на модела-генератор. Ще покажем, че тези две понятия са близки, макар и различни.

Дали моделът-генератор е единствен? Ще покажем, че не е и че има минимален и максимален модел. Тук не става дума за броя на състоянията, за та това какво „знаят“ тези състояния.

Кой ще е генераторът, който ще използваме, за определянето на фактите и събитията? Отговорът е: „Някой от генераторите.“ Все пак, важно е да съзнаваме, че за всяко събитие трябва да се търси генератор, който го представя. (Едно събитие може да е подмножество от стрелките на един генератор, но да не е подмножество от стрелките на друг.)

Следващото обобщение, което ще направим ще бъде MDP моделът. Това ще бъде Partially observable Markov decision process, в който сме махнали rewards и discount factor. Новото при този модел е, че той отчита действията на агента. В този модел ще се откажем от ограничението събитието да е само едно (т.е. всички стрелки да са еднакви).

Ще забележим, че в MDP модела агентът има свободна воля (тоест може да прави каквото си поиска), докато светът е ограничен от някакви правила, които определят точно какво ще направи.

На базата на това ще направим два варианта на MDP. В първия вариант и светът и агентът ще са лишени от свободна боля и ще са задължени да изпълняват точно определена стратегия. (Този модел ще го наречем MDP Fixed.) Във втория вариант и двамата ще имат свободна воля и ще могат да правят каквото си поискат. Тоест ще могат да избират който си искат от възможните ходове.

Ще направим едно обобщение на тези три модела (на MDP и на двата му варианта). Това обобщение ще го наречем MDP Plus и ще покажем, че този модел е полу-съвършен. Тоест частично ще се откажем от съвършенството, но все още не напълно.

Ще въведем понятието „предпочитание“. Моделът ни казва какво може да се случи, а предпочитанието ни казва какво агентът предпочита да се случи. Предпочитанието ще е това, чрез което агентът ще влияе на света.

Ще направим обратния на MDP Plus модела, който предсказва миналото. Ще видим, че обратният на MDP модела не е MDP модел, което е причината другите автори да не говорят за обратен модел и за предсказване на миналото.

След това въведение и разглеждане на съвършени и полу-съвършени модели, ние напълно ще се откажем от съвършенството като се откажем от свойството на Марков.

Следващата стъпка ще бъде да заменим действията на агента с произволни събития. Това ще бъде най-важното обобщение в резултат на което ще получим event-driven моделите.

За да има смисъл моделът, в състоянията му трябва да се случва нещо специално, което да ги отличава? Това специално нещо ще наречем следа. Ще покажем, че следата също може да е несъвършена.

Зависимост, която се случва от време на време ще наречем явление. Ще въведем следа с памет. Например, едно състояние може да помни, че там се е заселил някой обект. Също така може да помни кое явление агентът е наблюдавал в това състояние предишния път и следващия път пак да му демонстрира същото явление.

Ще въведем обекта като абстракция. Обектът ще има някакви свойства, а свойствата ще представим като event-driven модели.

Ще видим каква е връзката между модела-генератор и event-driven модела. Тя е, че event-driven моделът е фактор множеството на някой от генераторите. Хубавото на фактор множеството е, че така съществено се намалява броят на състоянията. Лошото е, че се губи много от информацията за света, защото състоянията са отъждествени чрез някаква релация на еквивалентност, а тази релация може да е много груба.

По този начин теоретично може да получим event-driven модел от модела-генератор, но ние няма да тръгнем по този път, а ще го постоим директно използвайки реални събития.

Как ще определим събитията? Не става дума за формална дефиниция, а реално да си отговорим на въпроса кога едно събитие се случва. Ще определяме събитията пряко и косвено.

Прякото откриване ще става чрез характеристична функция, а косвеното чрез следата (тоест, по това което се случва, ще заключим, че сме преминали в друго състояние и оттам, че се е случило събитие).

## Литературен обзор

Относно event-driven моделите, най-ранното изследване, което намерихме в литературата, са статиите [1] и [2]. Тук ще обсъдим статията [2], защото там идеите са по-ясни и по-изчистени.

В [2] авторите са забелязали, че моделът-генератор има прекалено много състояния. Оттук следва, че трябва да се потърси модел с по-малко състояния. За съжаление авторите на [2] не са тръгнали по този път, а са се опитали директно да търсят стратегия прескачайки намирането на модел.

Все пак, в [2] е направено най-важното. Там е показано, че в основата на разбирането на света не трябва да са действията на агента, а произволни събития. Авторите на [2] въвеждат понятието „събитие“. Дефиницията на събитие, която ние използваме в тази статия е заимствана от дефиницията дадена в [2] (с известно уточнение).

Има съществена разлика между подхода приложен в статията [2] и подхода, който ние прилагаме. Разликата е в това, че ние търсим модел, а те търсят стратегия.

Когато попаднете в непозната ситуация, кой ще е първият въпрос, който ще си зададете? Дали това ще е въпросът „Какво става?“ или въпросът „Какво да правя?“ Важният въпрос е вторият. Важното е да решим какво ще правим, но първият въпрос, който ще си зададем е „Какво става?“ Ако знаем какво става, лесно ще намерим отговор и на въпроса „Какво да правя?“

Ние в тази статия търсим модел, тоест опитваме се да отговорим на въпроса „Какво става?“ Авторите на [2] търсят стратегия. Тоест те се опитват да отговорят директно на въпроса „Какво да правя?“ пропускайки отговора на първия въпрос.

Тоест авторите на [2] са забелязали проблема и са направили първата стъпка за решаването на този проблем, но не са направили втората стъпка, а тя е въвеждането на event-driven моделите.

## Reinforcement Learning без reinforcement

Както казахме, ние ще търсим отговор само на въпроса „Какво става?“ и няма да си задаваме втория въпрос „Какво да правя?“ Тоест на нас не ни е нужна цел. Повечето автори дефинират MDP специално за Reinforcement Learning, като добавят и цел. Тоест добавят rewards и discount factor.

Ако търсим стратегия, ние не можем без цел, но ако търсим само модел тогава целта не ни е нужна. Тоест ще разглеждаме Reinforcement Learning без reinforcement. Няма да имаме цел и единствената ни цел ще е познанието. За нас целта няма да е важна, защото ако разберем света достатъчно добре, тогава за нас няма да е проблем постигането на произволна цел.

В [6] вече обсъдихме защо discount factor трябва да се махне от дефиницията на MDP. Ако махнем и rewards от дефиницията ще загубим целта, но дали няма да загубим и част от информацията. За да не губим информация ще преместим rewards в наблюдението. Например, в училище целта на децата са оценките. Ако махнем оценките, ще скрием част от информацията, но може да оставим оценките и да кажем на децата, че те не са цел, а са само за сведение.

Единственият проблем ще е това, че при някои модели наблюдението е свързано със състоянието на света, а наградата е свързана с прехода (стрелката). Затова ние ще предположим, че имаме следа (т.е. нещо специално се случва) не само в състоянията, но и при преходите между състояния.

### Какво ни е дадено?

Имаме агент и свят, които си взаимодействат. Това взаимодействие можем да си го мислим като *наблюдение-действие* или като *въпрос-отговор*.

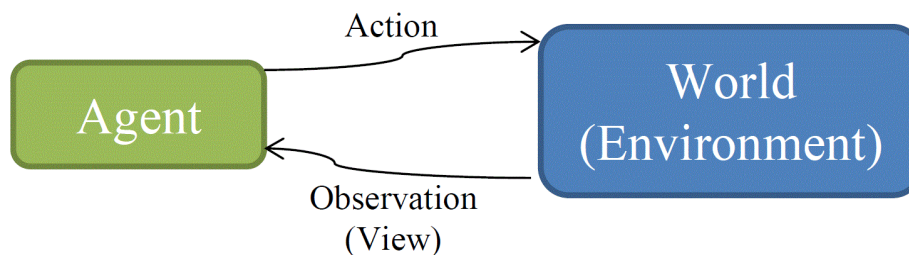


Figure 1

Резултатът от това взаимодействие е редицата:

$$\dots, v_{-2}, a_{-2}, v_{-1}, a_{-1}, v_0, a_0, v_1, a_1, v_2, a_2, \dots$$

Нека множеството на възможните наблюдения е  $\Omega$ , а множеството на възможните действия е  $\Sigma$ . Нека  $\Omega$  и  $\Sigma$  са крайни множества. Можем да си мислим редицата от действия и наблюдения като една дума (крайна или безкрайна) над азбуките  $\Omega$  и  $\Sigma$ .

Тук моментът „нула“ ще бъде текущият момент. Преди този момент ще бъде миналото, а след него ще бъде бъдещето. Повечето автори предполагат, че съществува някакво абсолютно начало, първи момент преди който няма минало. Ние няма да предполагаме съществуването на такова абсолютно начало. В [6] обяснихме, че дори да имаме абсолютно начало, то е твърде назад в миналото и е по-добре да не се занимаваме с него, а да се интересуваме само от текущия момент. При повечето автори терминът начален

момент се използва за означаване на абсолютното начало, а при нас „начален момент“ ще наричаме текущия момент.

## Какво търсим?

Търсим модел на света. Моделът ще има вида на ориентиран граф (фигура 2). Възлите на графа ще наричаме състояния. Едно от състоянията ще наричаме „начално“ или „текущо“. Това е състоянието 2 на фигура 2 и то е отбелязано с по-голям кръг.

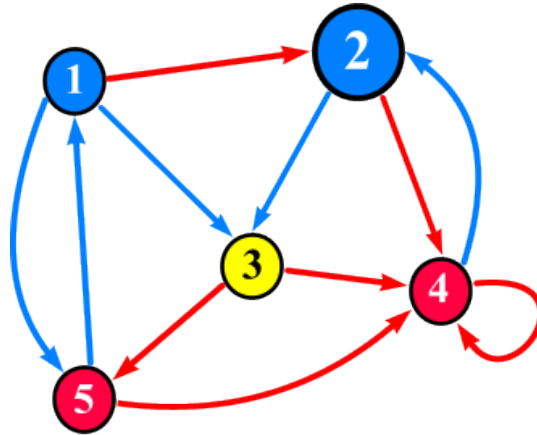


Figure 2

На стрелките ще съответстват някакви причини за промяната на състоянието на света. Тези причини ще наречем „събития“ и на фигура 2 сме отбелязали тези събития с различен цвят на стрелките.

На всяка стрелка ще съответстват две вероятности. Първата ще бъде вероятността да бъде избрана стрелка с този цвят (това събитие да се случи), втората ще бъде вероятността да бъде използвана точно тази стрелка (от няколкото които имат този цвят). Произведението на тези две вероятности ще е равно на вероятността тази стрелка да бъде използвана.

Много често едната от тези две вероятности ще е ясна и тогава на стрелките ще слагаме само една вероятност. Вероятността я разделяме на две, защото изборът на стрелката е разделен на две. Първо се избира събитието и после се избира точната стрелка.

Дотук описахме модела, но той все още нищо не ни казва за света. За да има смисъл един модел, нещо специално трябва да се случва в неговите състояния. Това, което очакваме да се случи в неговите състояния, ще наречем „следата на модела“. На фигура 2 сме отбелязали следата като сме оцветили състоянията с различни цветове. Тоест, ако две състояния са оцветени с един и същи цвят, то в тези две състояния едно и също очакваме да се случва.

Смисълът на модела е да ни каже нещо за бъдещето и за миналото. Съвършен модел ще бъде този, който ни дава съвършено описание на бъдещето, при условие че стратегията на агента е фиксирана. (Бъдещето се определя както от света, така и от агента. Затова не можем да искаме да знаем какво ще се случи без да знаем какво ще направи агентът. Затова предполагаме, че стратегията на агента е фиксирана. Искаме съвършено описание

само на бъдещето, но не и на миналото, защото моделът може да описва съвършено бъдещето без да описва съвършено миналото.)

Трябва да кажем какво е съвършено описание на бъдещето и какво е стратегия.

## Дефиниции

Редицата от действия и наблюдения няма да е еднозначно определена, дори моделът да е даден. Тоест моделът ще има много възможности за минало и за бъдеще.

Редицата от действия и наблюдения преди началния момент ще наречем възможно минало, а редицата от началния момент нататък ще наречем възможно бъдеще. За да не работим с безкрайни думи, ще разгледаме възможните развития на миналото и на бъдещето.

**Дефиниция.** Възможно развитие на миналото ще наричаме всяка крайна дума, която е край на някое възможно минало.

**Дефиниция.** Възможно развитие на бъдещето ще бъде всяка крайна дума, която е начало на някое възможно бъдеще.

Ще дефинираме „съвършено описание на бъдещето“, в което всяко възможно развитие на бъдещето ще се случва с точно определена вероятност.

**Дефиниция.** Съвършено описание на бъдещето ще наричаме множеството *Future* всеки един от елементите, на което има вида  $\langle \omega, p \rangle$ , където  $\omega$  пробягва възможните развития на бъдещето, а  $p$  е вероятността това развитие да се случи ( $p > 0$ ).

$$Future = \{ \langle \omega, p \rangle \mid \omega \text{ е възможно развитие на бъдещето, } p \text{ вероятността } \omega \text{ да се случи} \}$$

**Дефиниция.** Стратегия на агента ще наричаме функция, която на всяко състояние и възможно действие ни дава вероятността това действие да бъде извършено от агента.

$$Policy : S \times \Sigma \rightarrow \mathbb{R}$$

**Дефиниция.** Детерминирана стратегия ще наричаме такава, която дава вероятност единица за една от възможностите и вероятност нула за останалите.

Повечето автори, когато говорят за стратегия, имат предвид детерминирана стратегия. Ако хвърляме ези или тура и така определяме дали да завием наляво или надясно, това е стратегия, но това не е детерминирана стратегия.

**Дефиниция.** Съвършен модел ще бъде този, който ни дава съвършено описание на бъдещето, при условие че стратегията на агента е фиксирана.

## Fully observable Markov model

Това е възможно най-простият модел на света.

**Дефиниция:** Fully observable Markov model е следното:

$S = \Omega$  (множеството на състоянията съвпада с множеството на възможните наблюдения)

$p : S \times S \rightarrow \mathbb{R}$  (вероятността за преход от едно състояние към друго)

$p(i, j) = Pr(V_n=j / V_{n-1}=i)$  (вероятността за преход от състоянието  $i$  към състоянието  $j$ ) Тук  $V_n$  е редицата от наблюдения.

В литературата наричат този модел Discrete-time и Time-homogeneous Markov chain. Discrete-time означава, че процесът е стъпков, а Time-homogeneous означава, че вероятностите не зависят от това на коя стъпка е процесът. На фигура 3 е показана такава верига.

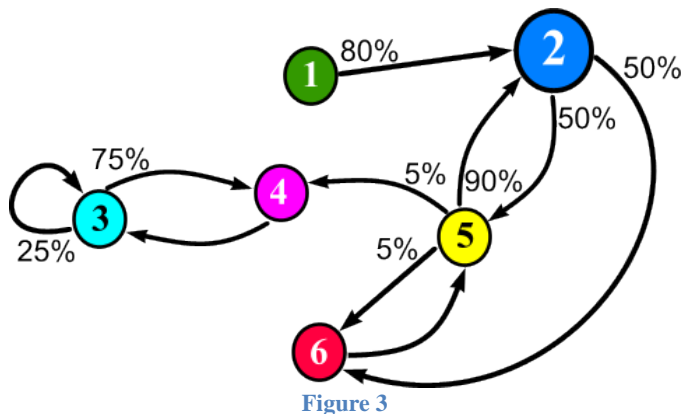


Figure 3

Тук стрелките са само един цвят, защото има само едно събитие и това е събитието „истина“ (това събитие се случва винаги, т.е. на всяка стъпка).

Тук следата на модела ще бъде тотална. Тоест, във всяко състояние нещо специално ще се случва. Това, което ще се случи в съответното състояние е, че агентът ще види едно точно определено наблюдение (един от елементите на  $\Omega$ ). Ще предполагаме, че имаме Full Observability, тоест че по наблюдението можем да определим кое е състоянието на модела, тоест че няма две състояния оцветени с един и същи цвят (т.е. в които наблюдаваме едно и също).

На всяка стрелка ще съответства една вероятност. (Вероятностите няма да са две, защото вероятността да се случи събитието е единица, защото има само едно възможно събитие. Тоест, остава само втората вероятност. На фигура 3 тази вероятност е поставена само там където има повече от една стрелка, защото когато стрелката е една, вероятността е единица.)

Дали Fully observable Markov model (FOMM) е съвършен модел на света? Да, но само ако изпълнява свойството на Марков.

Ако бъдещето зависи само от това в кое състояние на света сме, но не зависи от това как сме стигнали до това състояние, тогава моделът е съвършен. Свойството на Марков означава, че моделът не може да бъде подобрен. Ако имаше два различни начина да се стигне до едно състояние и тези два начина ни даваха различно бъдеще, тогава ние бихме могли да разделим състоянието на две, по единият начин да отидем в едното състояние, а



по другият в другото. Така бихме подобрили модела, защото тези две нови състояния биха ни дали по-точна прогноза за бъдещето. Ако имаме свойството на Марков, тогава тези две нови състояния ще дадат еднаква прогноза за бъдещето, тоест не бихме подобрили модела по този начин.

## Обратен FOMM

Видяхме, че FOMM ни дава предсказание за бъдещето. Дали можем да обърнем този модел и с него да започнем да предсказваме миналото? Ако тръгнем срещу стрелките, ще получим възможните развития на миналото, но ние искаме да знаем не само кое развитие е възможно и кое е невъзможно, а искаме още да знаем вероятността на всяко от възможните развития.

За всяко състояние и за всяка изходяща стрелка имаме вероятността от това състояние да излезем по тази стрелка. Дали по тези вероятности ще можем да получим входящите вероятности? Тоест, за всяко състояние и всяка входяща стрелка да получим вероятността в това състояние да сме влезли по тази стрелка. Отговорът е: Да можем, ако нямаме „бели върхове“. Първо трябва да кажем какво е бял връх.

**Дефиниция.** Черна дупка ще наричаме множество от състояния, при което няма път от състояние от множество до състояние извън множеството. Ще искаме още множеството да е непразно и да не съдържа началното състояние.

Тоест, това е множество от състояния, в което веднъж ако влезем, повече не можем да излезем от него. Бял връх ще бъде обратното на черна дупка. Това ще е множество, от което веднъж ако излезем, повече не можем да се върнем в него.

**Дефиниция.** Бял връх ще наричаме множество от състояния, при което няма път от състояние извън множество до състояние от множеството. Отново ще поискаме множеството да е непразно и да не съдържа началното състояние.

На фигура 3 има бял връх (множеството  $\{1\}$ ) и черна дупка (множеството  $\{3, 4\}$ ).

Ще предполагаме, че във FOMM може да имаме бели върхове и черни дупки, но дали можем да имаме множества, които са и едното и другото. Тоест, в тези множества да не можем нито да влезем нито да излезем. Ще предполагаме, че нямаме такива състояния, защото тези състояния са излишни. Белият връх не участва в предсказването на бъдещето, а черната дупка в предсказването на миналото. Ако едно множество е двете едновременно, тогава то няма да участва в предсказването нито на миналото, нито на бъдещето. Затова ще предполагаме, че тези излишни състояния са отстранени и просто ги няма. Единственият проблем при отстраняването на тези излишни състояния е, че вътре в белите върхове може да се наруши правилото, че сумата от вероятностите на изходящите стрелки е единица. Такъв е примерът във фигура 3. Там от състояние 1 излиза една стрелка с вероятност 80%. Предполага се, че оставащите 20% отиват в ненужни състояния, които сме отстранили. Това нарушение няма да е проблем, защото изходящите вероятности в белите върхове въобще не са важни, защото те служат за предсказване на бъдещето, а белите върхове не участват в предсказването на бъдещето. Аналогично, горните разсъждения могат да се направят за черните дупки и за входящите вероятности, които ще въведем сега.

**Теорема 1.** За всеки FOMM, в който няма бели върхове, можем на базата на изходящите вероятности да изчислим входящите и по този начин да предскажем миналото. Ако предсказанието на бъдещето е било свършено, то и предсказанието на миналото пак ще е свършено.

**Доказателство:** Идеята е просто да обърнем стрелките и да получим нов FOMM, който предсказва миналото по същия начин, по който старият FOMM предсказва бъдещето. За целта ще трябва да преизчислим вероятностите и да заменим изходящите с входящи вероятности.

В това доказателство ще подходим, не като математици, а като инженери и вместо за вероятности ще говорим за бройки.

Ще предположим, че тръгваме от началното състояние и вървим по стрелките докато не се върнем пак в началното или не попаднем в черна дупка. Нека това сме го направили многократно, например 100 пъти. Ще преброим по всяко ребро колко пъти сме минали. На базата на тези бройки можем да сметнем изходящите вероятности и те ще са тези, които имаме първоначално (с някаква статистическа грешка). Ще можем да сметнем и входящите вероятности (пак със статистическа грешка).

Ако вземем стрелките излизащи от началното състояние и сумираме, по колко пъти сме минали по всяка една от тях, ще получим 100, защото 100 пъти сме тръгнали от началното състояние. Дали сумата от входящите стрелки към началното състояние пак ще е 100? Не, тя може да е по-малко, защото трябва да извадим случаите, когато сме попаднали в черна дупка.

По този начин можем да сметнем входящата вероятност на всички състояния, които не са част от бял връх или черна дупка. Ние предположихме, че в този FOMM няма бели върхове, а черните дупки не участват в прогнозата на миналото и затова при тях входящите вероятности не са важни. Тоест, полученият нов FOMM ще предсказва миналото.

Ако първоначалният FOMM е свършен и не може да се подобри, то и полученият обратен FOMM ще е свършен и няма да може да се подобри. Това твърдение е интуитивно ясно, но се нуждае от строго доказателство.

■

Видяхме, че ако имаме свършено описание на бъдещето, от него можем да получим свършено описание на миналото. Да, това е така, ако нямаме бели върхове. Какво правим, ако имаме. Тогава това вече не е така, защото върху белите върхове можем да сложим каквито си поискаме входящи вероятности.

Нека повторим доказателството на Теорема 1, но сега да предполагаме, че имаме и бели върхове. Сто пъти тръгнахме от началното състояние. Нека вземем една стрелка идваща от бял връх. Ще тръгнем от тази стрелка колкото пъти си искаме. Може да тръгнем нула, едно, сто или хиляда пъти. Ако не тръгнем нито веднъж по тази стрелка, това ще ни даде входяща вероятност нула за тази стрелка. Колкото повече пъти тръгнем от нея, толкова по-голяма вероятност ще получим за тази стрелка. Това ще го направим за всички стрелки

идващи от бели върхове. Ще тръгнем и ще вървим докато не стигнем до началното състояние или до черна дупка. По този начин ще се променят входящите вероятности не само на състоянията, в които влизаме непосредствено от бял връх, а ще има промяна при всички състояния, през които сме минали.

Тоест, на стрелките идващи от бял връх ние може да сложим произволна входяща вероятност и това ще промени всичките входящи вероятности в обратния FOMM. Входящите вероятности вътре в самите бели върхове не се определят по този начин, но на тях ние може да поставим произволна вероятност (и тя няма да промени останалите вероятности).

Тоест, видяхме, че в общия случай, ако имаме съвършена прогноза за бъдещето, от това не следва че от нея можем да получим съвършена прогноза за миналото. Може само в частния случай когато нямаме бели върхове.

## Стандартен FOMM

За всеки един свят ние можем да направим модел, който да е FOMM. Това ще стане като вземем множеството на състоянията да бъде множеството  $\Omega$  и стрелките и вероятностите върху стрелките ще поставим на базата на статистика за известен период от време. Тук нямаме проблем да събираме статистика, защото имаме Full Observability, тоест знаем във всеки момент в кое състояние сме. Проблем ще е, ако събираме статистика през един период, а се опитаме да я използваме за модел описващ друг период. Проблемът е, че двата периода могат да дадат различна статистика. Ние ще предположим, че двата периода съвпадат. Тоест, моделът ще описва периода от време, в който сме събирали статистика. Така е сигурно, че статистиката ни ще е адекватна.

Въпросът е доколко адекватен ще е този модел? Това ще е модел, който ни казва нещо за света. Той ще ни даде каква е вероятността за всеки две наблюдения да следват едно след друго. Това ще е средната вероятност, но дали този модел ще е съвършен, тоест дали не може да бъде подобрен.

Да, в един много частен случай моделът ще е съвършен и няма да може да бъде подобрен, но в общия случай ще има и по-добри модели. Тоест, възможно е така полученият модел да има свойството на Марков, но това е много малко вероятно. За да бъде този модел съвършен, трябва статистиката да е върху безкраен интервал от време, защото всяка крайна последователност има модел, който я описва напълно, тоест моделът ѝ може да бъде подобрен.

Дори и безкрайните интервали могат да бъдат описани напълно (само че от безкрайни модели). За да не може моделът да бъде подобрен, трябва да имаме континуум много безкрайни интервали. Само тогава може да е в сила свойството на Марков.

Например нека имаме редицата „черно“ и „бяло“, където всяко следващо наблюдение се определя с хвърляне на монета. Тогава, ако построим стандартния FOMM, той ще ни даде „черно“ с вероятност 50% и това ще е съвършеният модел. Нека света ни дава два пъти „черно“ и два пъти „бяло“, и така нататък. Тогава стандартният FOMM ще е същият, защото след „бяло“ ще има черно с вероятност 50%, но това няма да е съвършен модел,

защото моделът може да се подобри и да се каже с по-голяма точност какво ще видим (в конкретния случай дори съвсем точно).

Вярно е, че за да подобрим модела ще ни е нужна памет, а за целта трябва да се откажем от Full Observability.

## Динамична памет

Каква е разликата между постоянна и динамична памет? В седмицата има седем дни. Това е нещо постоянно, което вие можете да запомните веднъж завинаги, тоест да го сложите във вашата постоянна памет. От друга страна, днес е четвъртък. Това не можете да го запомните веднъж завинаги, защото утре ще е петък и това вече няма да е вярно. Това, че днес е четвъртък трябва да го сложите във вашата динамична памет и да го промените регулярно.

Нужна ли ни е динамична памет? Ако сме в един постоянен свят, в който непрекъснато е четвъртък, винаги е сутрин и винаги сме закусили, през цялото време сме на едно и също място и нищо не се променя, тогава на нас няма да ни е нужна динамична памет. Ако светът се променя, тогава ни трябва динамична памет, за да разбираме какво става.

В постоянната памет ще запомним модела, а в динамичната ще помним кое е текущото (началното) състояние, в което се намираме. В известен смисъл само динамичната памет е памет. ИИ ще трябва да разбере какъв е моделът и да го запомни в своята постоянна памет, но ако направим устройство, което не е ИИ и не е за всеки свят, а е направено специално за конкретния свят, тогава това устройство може да има описание на модела вградено вътре в себе си и ще трябва да помни само текущото състояние, в което се намира. Тоест, ако имаме устройство направено специално за конкретния свят, то няма да има нужда от постоянна памет, а само от динамична.

Колко голяма е динамичната памет на модела? Колко бита? Отговорът е логаритъм при основа 2 от броя на състоянията на модела. По-точно не от броя на състоянията, а от максималният брой състояния с еднакъв цвят, защото на нас не ни е нужно да помним какво виждаме в момента (цвета на състоянието), а само в кое от няколкото възможни състояния сме.

Каква е динамичната памет при Full Observability моделите? При тези модели максималният брой на състояния с еднакъв цвят е едно. Тоест, паметта е нула бита.

Нашето убеждение е, че интересните светове не са постоянни (променят се). Затова ще ни трябват модели с памет. Това е причината, поради която ще се откажем от Full Observability и ще разглеждаме само модели с Partial Observability. Тук се отличаваме от повечето автори, които предпочитат да работят с Fully observable модели и приемат, че тези модели са по-прости. Ние вярваме, че общият случай е по-прост от частния и затова Partially observable моделите са по-прости, по-разбираеми и вършат повече работа.

В момента при ИИ разработките има огромен напредък при разпознаването, но резултатите в разбирането на това, какво става, са много слаби. Например, ИИ програмите разпознават прекрасно лица и гласове, но не могат да проведат елементарен разговор.

Причината за това е, че повечето изследователи използват модели без памет (невронни мрежи и Fully observable модели). За разпознаването динамична памет не е нужна. Ако видите два пъти едно и също лице, от вас се очаква да кажете едно и също. Ако искате да проведете елементарен разговор ще ви потрябва динамична памет, защото ако ви зададат два пъти един и същи въпрос, не се очаква да отговорите едно и също. Би трябвало вие да помните, че този въпрос вече веднъж ви е зададен.

Затова се отказваме от Full Observability и преминаваме към следващите модели.

## Hidden Markov model

Тези модели се различават от FOMMs единствено по това, че тук можем да имаме няколко състояния с еднакъв цвят (т.е., в които виждаме едно и също наблюдение).

**Дефиниция:** Hidden Markov model е следното:

$S$  (множеството на състоянията)

$Trace : S \rightarrow \Omega$  (какво виждаме във всяко състояние)

$p : S \times S \rightarrow \mathbb{R}$  (вероятността за преход от едно състояние към друго)

$p(i, j) = Pr(s_n=j | s_{n-1}=i)$  (вероятността за преход от състоянието  $i$  към състоянието  $j$ )

Тук  $s_n$  е състоянието на  $n$ -тата стъпка.

Hidden Markov model (HMM) е модел с памет (за разлика от FOMM). След като моделът има памет, следва да си зададем въпроса какво помнят състоянията на модела. В тази статия ще говорим за това какво знаят състоянията за миналото и за бъдещето. Няма да използваме глагола „помня“, защото той предполага, че имаме едно единствено минало и че ние сме го запомнили, тоест записали сме някъде какво се е случило. В случая миналото няма да е запомнено, но на базата на модела ще можем да кажем някои неща за миналото и за бъдещето. Най-вече ще кажем кои развития са възможни и кои са невъзможни. Евентуално, за някои възможни развития ще можем да кажем каква е тяхната вероятност.

Какво може да знае състоянието за миналото и за бъдещето? То може да знае някакви факти. По друг начин казано, ако състоянието е текущото, какво тогава бихме могли да кажем за миналото и за бъдещето? Тогава бихме могли да кажем някакви факти.

Първо трябва да кажем какво е факт и какво е събитие.

## Факти vs. събития

Фактите са нещо, което понякога е истина, а понякога е лъжа. Събитията са нещо, което понякога се случва, а понякога не се случва. В [6] дефинирахме събитието като булева функция на времето. По същият начин можем да дефинираме и фактите и тогава събитията и фактите ще са едно и също нещо от теоретична гледна точка.

Все пак, за нас събитието е нещо което е истина от време на време, докато фактите са истина в определени интервали от време. Това е една неформална представа за това, че става дума за различни обекти, но ако ги разглеждаме като булеви функции те няма да са различни.

В тази статия ще променим дефиницията на събитие. Ние ще заимстваме дефиницията дадена в [2] и ще дефинираме събитието като множество от стрелки. Фактите ще ги дефинираме като множества от състояния. По този начин събитията и фактите ще са формално различни обекти, макар че пак ще са много близки.

**Твърдение 1.** Фактът и събитието са доста близки.

**Обосновка:**

Ще покажем, че всеки факт може да се представи като събитие. Нека вземем множеството от стрелките излизачи от състоянията на този факт. Това ще е събитие и то ще се случва в моментите, в които е верен фактът. (Ако бяхме взели стрелките, които влизат в състоянията, тогава щяхме да получим същото събитие изместено с една стъпка напред.)

Аналогично, можем да представим събитието  $E$  като факт, но няма да е факт в същия модел, а ще бъде факт в друг модел, който е еквивалентен на първия. Новият модел ще построим като удвоим състоянията на първия. Всяко състояние  $s_i$  ще заменим с  $s'_i$  и  $s''_i$ . Всяка стрелка  $s_i \rightarrow s_j$  ще заменим с две стрелки:

$$s'_i \rightarrow s''_j \text{ и } s''_i \rightarrow s''_j, \text{ ако } (s_i \rightarrow s_j) \in E$$
$$s'_i \rightarrow s'_j \text{ и } s''_i \rightarrow s'_j, \text{ ако } (s_i \rightarrow s_j) \notin E$$

Фактът ще е множеството  $S''$ , където  $S'' = \{s \mid \exists i : s = s''_i\}$ .

Тази конструкция не е много добра, защото фактът ще е истина не на същата стъпка, в която се е случило събитието, а на следващата. Освен това не е съвсем ясно, кое трябва да е новото начално състояние. Ако начално състояние е било  $s_0$ , то дали новото трябва да е  $s'_0$  или  $s''_0$ ? Ние обаче ще си затворим очите за тези несъвършенства на конструкцията. ■

Защо променяме дефиницията на събитие? Представете си, че имате статуетка, в която има пукнатина, която искате да опишете. Представете си, че не разполагате с цялата статуетка, а само с един резен (slice) от нея. Тогава ще опишете пукнатината само на базата на този резен. Това ще е едно непълно и неточно описание. Ако пукнатината я няма във вашия резен, това не значи, че въобще я няма. Все пак, ако разполагате само с един резен, това ще е всичко, което знаете за статуетката. Останалата част вие реално няма да виждате, но ще можете да си я представите. Затова е по-добре пукнатината да бъде описана като тримерен обект, който е част от една въображаема статуетка.

Аналогично е със света и конкретния живот. Ако ние имаме само един живот и решим да опишем събитието на базата на този конкретен живот, тогава то би имало вида на булева функция. Ако събитието не се е случило през нашия живот, това не значи, че в този свят това събитие е невъзможно. Ако бяхме изживели живота си по друг начин, то събитието можеше да се случи.

Затова ще дефинираме събитието като множество от стрелки в модела-генератор. (Ние предположихме, че съществува модел-генератор, който съвършено описва света.)

Проблемът е, че този модел съвсем не е единствен. Има много модели, които са еквивалентни помежду си.

## Минимални модели

**Дефиниция.** Два модела на света ще наричаме еквивалентни, ако ни казват едно и също за миналото и за бъдещето.

Тук не говорим за тривиална еквивалентност, където моделите са еднакви с точност до изоморфизъм. Ние ще разгледаме модели, при които състоянията знаят повече и такива, при които знаят по-малко.

Минимален модел ще бъде този, при който състоянията знаят минималното. При тези модели две състояния ни казват едно и също за миналото т.т.к. (iff) те казват едно и също за бъдещето. (Ако две състояния знаят едно и също за миналото и за бъдещето тези две състояния са еквивалентни.)

Ако говорим за миналото, в минималния модел състоянията няма да „помнят“ нищо излишно. „Излишно“ е някакъв факт, от който не зависи бъдещето. Сами разбирате, че щом от този факт не зависи бъдещето, е излишно да го помним. (Това означава, че ако две състояния ни казват едно и също за бъдещето, то те ще казват едно и също за миналото, защото излишните факти не се „помнят“.)

Аналогично, ако говорим за бъдещето, ще получим, че състоянията не знаят за бъдещето нищо което не следва от миналото. (Тоест, състоянието не може да знае нещо, което няма от къде да се знае.) Това означава, че ако две състояния ни казват едно и също за миналото, то те ще казват едно и също за бъдещето. (Ако възможните им развития на миналото съвпадат, то съвпадат и възможните им развития на бъдещето.)

Кога може да се наруши минималността? Кога може едно състояние да знае повече? Има две възможни причини:

Първата е недетерминираността. Когато тръгнем от началното състояние след няколко стъпки достигаем до множество от възможни състояния. (Ако множеството има и вероятности за различните състояния, тогава това се нарича „belief“.) Ако в множеството има много състояния с различно бъдеще, тогава те знаят повече от минималното, защото ние знаем, че сме в едно от състоянията от „вряването“, но няма от къде да знаем в кое точно състояние сме. Ако знаем точно в кое състояние се намираме, тогава знаем за бъдещето нещо, което няма откъде да го знаем.

Втората причина е когато имаме две състояния с различно минало, но с еднакво бъдеще. Тогава, ако ние знаем в кое от тези две състояния се намираме, ще знаем нещо излишно за миналото, нещо от което бъдещето не зависи.

Ако детерминираме модела ще отстраним първата причина, а ако го минимизираме, ще отстраним втората. Тоест, минималният модел е детерминиран и минимизиран в двете посоки (по посока на стрелките и в обратна посока).

В [5] разгледахме подробно въпроса за минималните модели. Пак в [5] описахме алгоритъм за намиране на минимален модел, при който от което и състояние да тръгнем (и в двете посоки) получаваме минималност и детерминираност. За съжаление алгоритъмът в [5] е грешен. Може да се види, че отстраняването на недетерминираността напред води до недетерминираност назад и обратното и затова алгоритъмът в [5] не работи.

В общия случай няма минимален модел, при който да можем да тръгнем от което и да е състояние. Все пак има минимален модел, при който ако тръгнем от текущото състояние напред ще имаме минималност и детерминираност. Също така, ако тръгнем назад, ще имаме същото. Този модел може да се получи на три стъпки:

1. Построяваме модел, който е минимален и детерминиран тръгвайки от началното състояние напред. (Детерминираният модел в литературата е известен като belief MDP. Построяваме този модел и го минимизираме, като сливаме състоянията, които имат еднакво бъдеще. Т.е. сливаме състоянията, при които съвпадат множествата от възможни изходящи последователности от действия и наблюдения и тези последователности имат еднакви вероятности.)
2. Построяваме обратния модел, който е единствен, ако нямаме бели върхове (ако имаме, тогава взимаме един от възможните обратни модели.). Детерминираме и минимизираме този модел (както направихме на стъпка 1). После от това, което сме получили правим обратен модел (така ще получим модел, който е детерминиран и минимален назад).
3. Накрая сглобяваме от тези два модела един. За целта правим нови начални състояния на двата модела и съединяваме двата модела в новите състояния.

Трите стъпки са илюстрирани в трите части на фигура 4.

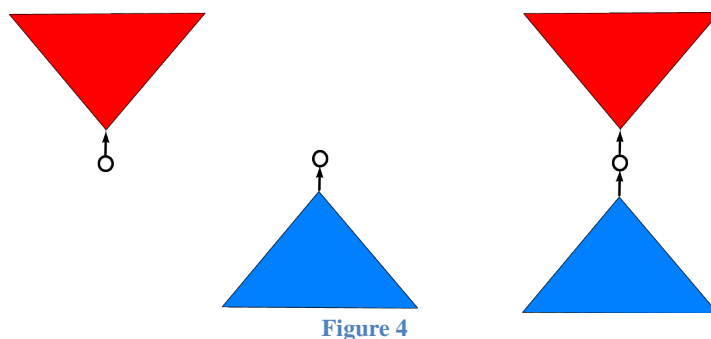


Figure 4

Гарантирано е, че бъдещето ще се определя единствено от първия модел, а миналото единствено от втория. Това е благодарение на факта, че направихме нови начални състояния. Така е гарантирано, че излезем ли веднъж от началното състояние повече не можем да влезем в него. (Тоест, първият модел е черна дупка, а вторият е бял връх.)

Полученият по този начин модел е минимален в смисъл, че състоянията знаят възможно най-малкото, но не е минимален в смисъл, че състоянията са възможно най-малко на брой. Ако искаме да получим минимален модел, който е с най-малко състояния ще трябва да комбинираме двата модела и да използваме някои от състоянията и за миналото, и за бъдещето.

В [5] говорихме за минимален модел в случая, когато нямаме вероятности на стрелките. Хубавото в този случай е, че от краен модел получаваме пак краен модел. Когато минимизираме MDP, тогава от краен модел може да получим безкраен. Причината за това е, че подмножествата на крайно множество са крайно много, но „вярванията“ са безкрайно много. (Тоест, минимизацията не намалява броят на състоянията, дори може съществено да ги увеличи.)



## Кой ще е генераторът

Предположихме, че съществува модел генератор, който съвършено описва бъдещето. Видяхме, че този модел не е единствен. В [5] видяхме, че имаме минимален модел където състоянията не знаят нищо излишно и че имаме максимален модел, където състоянията знаят всичко за миналото и за бъдещето.

Бихме могли да допуснем съществуването на модели, в които състоянията знаят дори повече от тези в максималния модел. Какво повече от всичко, ще попитате вие? Биха могли да знаят някакви несъществени факти, от които не зависи нито миналото, нито бъдещето. Например „Има ли живот на Марс?“. Да допуснем, че отговорът на този въпрос е факт, от който не зависи нито миналото ви, нито бъдещето ви. В този случай това е един несъществен факт. Ще предположим, че нас несъществените факти не ни интересуват и затова няма да разглеждаме модели с такива факти. (Несъществените факти безсмислено усложняват модела. Например фактът „Има живот на Марс“ може да се представи като моделът се удвои. Ще получим два еднакви модела и ще сме в първия, ако има живот на Марс и съответно във втория, ако няма. Тоест, моделът ще знае отговора на този несъществен въпрос, но нищо няма да последва от това, защото в двата еднакви модела миналото и бъдещето са еднакви. Освен несъщественото удвояване на модела може да се получи и несъществено разклонение. Например, появява се живот на Марс и това в нашия модел съответства на едно несъществено разклонение. Ако тръгнем наляво, ще знаем че вече има живот, ако тръгнем надясно, ще знаем че още няма. Това знание отново ще е несъществено, ако от него не следва нищо за миналото и за бъдещето ни.)

Добре, нека да изберем генератора на базата на който ще дефинираме факт и събитие. Първият подходящ кандидат е минималният модел, но в този модел не се помни нищо излишно. Защо да не си позволим да помним излишни неща? Ние не знаем предварително кое е полезно и кое е излишно и затова помним много излишни неща. Например, в нашият свят е много важно да знаем кой е денят по модул 7, защото това са дните от седмицата. Никой обаче не се интересува кой е денят по модул 2, защото това в нашия свят е без значение. Все пак, нека не забраняваме да се помни кой е денят по модул 2, защото това в един момент може да вземе да се окаже полезно.

Освен това, минималният модел не е единствен. Вярно, ако кажем „минималния модел, който има най-малко състояния“, тогава този модел ще е единствен, но състоянията му може да са доста много, защото минимизацията може да увеличи броят на състоянията. (Например, от краен модел чрез минимизация можем да получим безкраен.) При минималния модел състоянията знаят минималното, но това не значи, че броят им е минимален.

По тези причини няма да изберем минималния модел. Дали да не вземем максималния? Не, защото този модел е твърде сложен. Началното му състояние е „belief“ с континуум много състояния. През всяко едно от тези състояния минава една нишка, която съответства на един възможен живот. Ако вземем само един от възможните животи, тогава събитието ще е булева функция, но това, както казахме, не ни харесва.

Затова решаваме да не казваме кой точно е генераторът. Нека да е някакъв генератор, който е между минималния и максималния модел. Събитието ще е подмножество от стрелките на този генератор.

Тази дефиниция е добра, защото ние няма да я използваме. Ние искаме да имаме формална дефиниция на понятието събитие и го определяме като част от стрелките на някакъв генератор. Не казваме, кой точно е този генератор, но предварително казахме, че ние генератора няма да го търсим. Затова нека да е, който и да е от генераторите.

Това е уточнението, което ще направим към дефиницията на събитие дадена в [2]. Там се казва, че събитието е подмножество от стрелките на генератора, но не се уточнява кой генератор. Може би авторите на [2] предполагат, че генераторът е единствен, а това не е така.

## MDP моделите

Недостатък на FOMM и на HMM е, че в тези модели действията на агента не се взимат под внимание. Възможно е светът да е такъв, че действията на агента въобще да не влияят на това какво ще се случи. Възможно е действията на агента да влияят, но моделът да описва някаква зависимост, която не зависи от тези действия. Тоест, FOMM и HMM може да са полезни, но в общия случай не са достатъчни. Ние бихме искали да имаме модел, който може да отчете действията на агента и затова преминаваме към следващите модели. Това са MDP моделите.

Нека да отбележим, че става дума за Partially observable, а не за Fully observable модели. Ние ще изпускате определението Partially observable, защото ще смятаме, че когато не е казано дали е Partially observable или Fully observable, то се подразбира Partially observable. Тук ние се различаваме от повечето автори, които смятат, че когато това не е казано, се подразбира Fully observable.

Разликата между HMM и MDP моделите е, че при HMM имаме само едно възможно събитие (това е събитието „истината“), а при MDP възможните събития са  $\Sigma$  (действията на агента).

Както при HMM така и при MDP моделите можем да дефинираме следата като конкретно наблюдение, което задължително виждаме в това състояние или като множество от възможни наблюдения, всяко от които има точно определена вероятност. Двете дефиниции са еквивалентни и затова при HMM избрахме първата, защото е по-проста, а при MDP ще изберем втората, защото тя дава модел с по-малко състояния. (При минималните модели ще предполагаме че следата дава точно едно наблюдение, защото ако дава няколко възможни наблюдения, това ще е недетерминираност, а характерна особеност на минималните модели е тяхната детерминираност.)

**Дефиниция:** MDP model е следното:

$S$  (множеството на състоянията)

$Trace : S \times \Omega \rightarrow \mathbb{R}$  (вероятността да видим конкретно наблюдение в конкретно състояние)

$Agent : S \times \Sigma \rightarrow \mathbb{R} \times \mathbb{R}$  (Вероятността агентът да избере определено действие, по-точно интервалът, в който тази вероятност се намира. Този интервал винаги ще е  $[0, 1]$ .)

$World : S \times \Sigma \times S \rightarrow \mathbb{R}$  (вероятността за преход от едно състояние по дадено действие към друго състояние)

$World(i, a, j) = Pr(s_n=j \mid s_{n-1}=i, a_{n-1}=a)$  (вероятността за преход от състоянието  $i$  към състоянието  $j$  при действието  $a$ )

Тук  $s_n$  е състоянието на  $n$ -тата стъпка, а  $a_n$  е действието на  $n$ -тата стъпка.

В дефиницията функцията *Agent* е напълно излишна. Не ни трябва функция, която винаги връща интервала  $[0, 1]$ . Защо ни е функция, която е константа? Добавихме тази функция, само заради двата варианта на MDP модела, които ще направим (MDP Fixed и SMDP). В тези варианти тази функция няма да е константа.

В дефиницията не сме сложили rewards и discount factor, но както обяснихме по-горе, те са нещо, което за момента не ни трябва.

**Забележка.** Важно е да се отбележи, че за всеки свят съществува MDP модел, който го описва еднозначно и напълно. Тоест има съвършен MDP модел. Може да няма краен MDP модел, но безкраен MDP модел задължително има. Това означава, че можем да предпологаме, че света има модел генератор и този модел е MDP. (Същото може да се каже за НММ, ако ограничим световите до такива, в които действията на агента не влияят на света.)

## Свободна воля и ограничение

MDP моделът е съвършен, защото той ни казва всичко за бъдещето, при условие че знаем каква стратегия ще следва агентът. Защо MDP моделът не фиксира стратегията на агента? Защото предпологаме, че агентът не е част от света и че той има свободна воля (тоест, може да прави каквото си поиска), докато светът е ограничен от някакви правила, които определят точно какво ще направи (определят с точност до някаква вероятност).

Тоест, ние предпологаме, че агентът има свободна воля, а светът няма. Ако предположим, че вие сте задължен да хвърлите монета и да завиете наляво или надясно според това, какво покаже монетата, това означава, че вие нямате свободна воля, а сте длъжен да се подчините на монетата.

Нека си мислим, че агентът и светът са двама играчи, които играят една игра или че са два субекта, които си беседват. Тогава ще забележим, че в MDP модела тези два субекта не са равноправни. Агентът може да прави каквото си поиска докато светът е задължен да изпълнява една определена стратегия. Агентът може да избере всеки един от възможните ходове с каквато си поиска вероятност (тоест с вероятност в интервала  $[0, 1]$ ), докато светът за всеки възможен ход има точно определена вероятност и трябва да избере този ход точно с тази вероятност. Тоест, агентът има свободна воля, а светът е ограничен до една единствена стратегия.

Може напълно да сте лишен от свободна воля и да сте длъжен да изпълнявате една точно определена стратегия. Има и друг вариант. Може да имате свободна воля, но тя да е ограничена в някакви граници. Тоест, да можете да избирате стратегията си в някакви граници.

**Дефиниция.** Ограничение на агента ще наричаме функция, която на всяко състояние и възможно действие ни дава интервал, в който се намира вероятността това действие да бъде извършено от агента.

$$\text{Constraint} : S \times \Sigma \rightarrow \mathbb{R} \times \mathbb{R}$$

Напълно свободна воля имаме когато интервалът е  $[0, 1]$ . Свободната воля липсва напълно, когато интервалът е с дължина нула (тоест, когато вероятността е точно определена).

Аналогично на стратегия на агента можем да дефинираме стратегия на света:

**Дефиниция.** Стратегия на света ще наричаме функция, която за всяко състояние, действие и ново състояние ни дава вероятността светът след това действие на агента да премине в новото състояние.

$$Policy: S \times \Sigma \times S \rightarrow \mathbb{R}$$

Аналогично на ограничение на агента можем да дефинираме ограничение на света. Функциите *Agent* и *World* от дефиницията на MDP са съответно ограничението на агента и стратегията на света.

Може ли светът да има свободна воля? Представете си, че вътре в света живее един друг агент, който има свободна воля. Действията на този друг агент ще се проявят в поведението на света. Дори този друг агент да може да прави каквото си поиска, това не значи че светът ще прави каквото си поиска. Може светът да има известна свобода, но да си остане ограничен от някакво ограничение.

Когато говорим за света може вместо „свободна воля“ да кажем „непрогнозируема случайност“. В някои случаи знаем точно какво ще се случи. Понякога, не знаем точно какво ще се случи, но знаем с каква вероятност това може да се случи. Ако не знаем дори и вероятността, с която това може да се случи, тогава имаме непрогнозируема случайност. Например, ако в света живее един агент, който прави каквото си поиска, то какво ще направи този агент е непрогнозируема случайност.

## Варианти

Сега ще разгледаме вариант на MDP, при който и светът, и агентът са задължени да следват определена стратегия. Тоест, и двамата нямат свободна воля. Този модел ще наречем MDP Fixed. Единствената разлика с MDP ще бъде това, че функцията *Agent* няма два връща интервалът  $[0, 1]$ , а ще връща конкретна стойност (тоест, ще връща интервал с дължина нула). MDP Fixed моделът също е съвършен, защото казва всичко за бъдещето.

Ще разгледаме друг вариант на MDP, при който свободна воля ще имат и светът и агентът. Тук функцията *Agent* ще връща интервала  $[0, 1]$  или стойностите 0 или 1 (за случаите когато действието е невъзможно или когато има само едно възможно действие.) Аналогично, функцията *World* няма да връща конкретна стойност, а и тя ще връща интервала  $[0, 1]$  или стойностите 0 или 1 (за случаите когато прехода е невъзможен или когато има само един възможен преход.) Този вариант ще наречем State machine decision process (SMDP). Така ще го наречем, защото той много прилича на Nondeterministic finite-state machine (NFSM). Разликата е, че при NFSM се предполага, че състоянията са крайно много, докато при SMDP такова предположение не правим. Друга разлика е, че при NFSM имам два вида състояния (final and not final) докато при SMDP има много видове състояния. (Ако във всяко състояние виждаме конкретно наблюдение, тогава видовете

състояния са  $\Omega$ . Ако във всяко състояние виждаме няколко наблюдения с различна вероятност, тогава видовете състояния са безбройно много). Основната прилика между NFSM и SMDP е, че и в двата случая не пишем вероятности върху стрелките. При SMDP вероятността е интервала  $[0, 1]$  или 1 (в зависимост от това има или няма недетерминираност). Тоест, при SMDP вероятността е ясна и затова не се пише. (По-точно и двете вероятности на стрелката са ясни и затова не се пишат.)

SMDP моделът не е съвършен, но ние ще го наречем полу-съвършен.

**Дефиниция.** Полу-съвършено описание на бъдещето ще наричаме множеството *Future* всеки един от елементите, на което има вида  $\langle \omega, [a, b] \rangle$ , където  $\omega$  пробягва възможните развития на бъдещето, а  $[a, b]$  е най-малкият възможен интервал, в който се намира вероятността това развитие да се случи ( $[a, b] \neq [0, 0]$ ).

**Дефиниция.** Полу-съвършен модел ще бъде този, който ни дава полу-съвършено описание на бъдещето. (Тук не ограничаваме стратегията на агента, освен ако моделът не налага такова ограничение.)

## MDP Plus

Ще обобщим MDP и неговите два варианта. Ще получим един модел, който ще наречем MDP плюс свободна воля и непрогнозируема случайност (за по-кратко MDP Plus).

В този модел и светът и агентът имат свободна воля, но тя не е неограничена (т.е. поставена е в някакви граници).

**Дефиниция:** MDP Plus model е следното:

$S$  (множеството на състоянията)

$Trace : S \times \Omega \rightarrow \mathbb{R}$  (вероятността да видим конкретно наблюдение в конкретно състояние)

$Agent : S \times \Sigma \rightarrow \mathbb{R} \times \mathbb{R}$  (Вероятността агентът да избере определено действие, по-точно интервалът, в който тази вероятност се намира.)

$World : S \times \Sigma \times S \rightarrow \mathbb{R} \times \mathbb{R}$  (Вероятността за преход от едно състояние по дадено действие към друго състояние, по-точно интервалът, в който тази вероятност се намира)

Разликата между MDP и MDP Plus е, че при MDP на всяка стрелка съответства по една вероятност, а при MDP Plus на всяка стрелка съответстват по два интервала. При MDP първият интервал не се пишеше, защото той винаги беше интервалът  $[0, 1]$ . Вторият интервал при MDP е с дължина нула и затова се пишеше само едно число.

MDP Plus моделът е полу-съвършен. Той е обобщение на MDP модела. При него съвършенството се нарушава, защото тук сме дали известна свобода на света. Тоест, светът не е задължен да следва конкретна стратегия, а може да си избира стратегията в рамките на някакво ограничение.

## Предпочитание

Повечето автори, когато говорят за стратегия, имат предвид, че агентът може да си избере която си поиска стратегия. В тази статия говорим освен за стратегия на агента и за стратегия на света. Естествено е да предполагаваме, че агентът може да си избере действието, което ще извърши, но не е естествено да предполагаваме, че агента може да

избере поведението на света. От друга страна не е естествено да предполагаме, че агента няма никаква възможност да влияе на поведението на света.

По-ясни ще станат нещата, когато заменим действията на агента с произволни събития. Естествено е да предполагаме, че агентът може да избере своето действие, но не и че може да избере какво ще е следващото събитие, което ще се случи.

Дори и за действията на агента не е съвсем естествено да предполагаме, че агентът ги контролира напълно. Например, „завършвам университета“ или „уцелвам десетката, при игра на дартс“. Тези действия не са напълно под наш контрол. Затова е по-добре да кажем, кое е нашето предпочитание, а не какво точно събитие ще се случи или кое действие ще извършим.

Какво е предпочитание? Имаме някакво ограничение, което ни дава моделът и вътре в това ограничение имаме една непрогнозируема случайност. Например, моделът ни казва, че вероятността за дъжд е задължително над 10% и под 80%. Тоест, моделът казва, че по никакъв начин не можем да свалим вероятността за дъжда под 10% и да я качим над 80%. Моделът ни казва кой е интервалът, но не ни казва точната вероятност. Ние може да имаме някакво предпочитание. Може да предпочитаме да вали и да полее реколтата или да не вали и да отидем на плаж. Нашето предпочитание може по някакъв начин да повлияе на вероятността. Ние може да извършим действия, които да променят вероятността в посока на нашето предпочитание. Например, може да направим молитва за дъжд или да изстреляме метеорологични ракети, които да предизвикат дъжд.

Това, че предпочитаме нещо да се случи, това не значи, че то ще се случи. Понякога дори е обратното. Колкото повече искаме нещо да се случи, толкова по-малко вероятно е то да се случи. Ние ще влияем на събитията чрез нашите предпочитания. Колко можем да влияем зависи от това доколко е нашата власт. Ако нашата власт е абсолютна ще наречем предпочитанието „шефско“. Например, ако шефското предпочитание е да вали, то ще вали с вероятност 80% (максималната вероятност, която моделът позволява). Ако шефското предпочитание е да не вали, то ще вали с вероятност 10% (минималната вероятност).

Предпочитанието няма да е част от модела. Имаме модел, който ни казва какво може да се случи и предпочитание, което казва какво ние предпочитаме да се случи.

**Дефиниция.** Предпочитание за действието ще наричаме функция, която за всяко състояние ни връща списък от предпочитаните действия. Действията в списъка са наредени от най-желаното към най-нежеланото.

$$Preference : S \rightarrow List(\Sigma)$$

Предпочитанието за действието ще ни даде една детерминирана стратегия (играем най-предпочитания ход). Това е, ако нямаме ограничение на действието. Ако имаме ограничения, тогава ще получим една недетерминирана стратегия, която ще е шефското предпочитание. Т.е. ако ограниченията са  $[a_1, b_1], [a_2, b_2], \dots$  тогава играем първия ход с вероятност  $b_1$ , втория с вероятност  $(1-b_1).b_2$  и така нататък.

По аналогичен начин можем да дефинираме „предпочитание за реакцията на света“ и „предпочитание за следващото събитие“.

## Обратен MDP Plus

Можем ли да обърнем MDP Plus модела и да получим MDP Plus модел, който да предсказва миналото? Отговорът е, че можем. Пак ще предположим, че нямаме бели върхове, защото в противен случай обратния MDP Plus модел няма да е единствен.

Обръщането ще стане по същия начин по който обърнахме FOMM. Ще започнем от обръщането на MDP Fixed. Там вероятността на всяка стрелка напред е фиксирана. Като го обърнем вероятността на всяка стрелка назад пак ще е фиксирана (без бели върхове). За всяко състояние и стрелка влизаща в него ще имаме фиксирана вероятност от тази стрелка да сме влезли. Тази вероятност трябва да се раздели на две вероятности (първата – да имаме това конкретно действие и втората – да е точно тази стрелката при това действие). Това разделяне е тривиално. Така ще получим MDP Fixed модел. Тоест, обратния на MDP Fixed модел пак ще бъде MDP Fixed модел.

Ако обърнем MDP Plus модел, тогава вероятностите на стрелките няма да са точни стойности, а ще са интервали. Как да получим интервалите от вероятности на стрелките в обратна посока? При MDP Plus агентът и светът имат много възможни стратегии. За всяка стрелка от всичките тези стратегии ще изберем тези при които тази стрелка най-рядко се използва. Така ще получим минимума. Максимума ще го получим като изберем тези стратегии, при които най-често се използва тази стрелка. Така за всяка стрелка ще получим интервал от вероятности. Как се разделя този интервал на два интервала? Отново, това е тривиално.

Интересното е, че обратния на MDP модела не е MDP модел (има обратен, но той е MDP Plus модел). Това е причината, поради която другите автори при MDP не разглеждат обратния модел, който предсказва миналото. Нека агентът има напълно свободна воля, а светът е ограничен да спазва една точно определена стратегия (това е MDP модела). Тогава като обърнем стрелките получаваме агент, който има някакви ограничения и свят който има известна свобода (тоест получаваме MDP Plus, а не MDP модел).

## Свойството на Марков

Следващата ни стъпка ще е да се откажем от свойството на Марков. Тоест, ще се откажем от свойството, че моделът не може да бъде подобрен. По този начин ние напълно ще се откажем от съвършенството и полученият модел няма нито да е съвършен, нито да е полу-съвършен.

При FOMM ние вече показахме, че ако имаме свойството на Марков, тогава моделът не може да се подобри. Това свойство предполага, че всички факти, които са съществени за миналото и за бъдещето са отразени (запомнени). Тоест, няма нищо, което да си струва допълнително да се запомни. Всичко, което си е струвало да се помни, вече сме го запомнили. Ако имаше такъв факт, бихме могли да подобрим модела, като добавим този факт към запомнените (това ще го направим като увеличим броя на състоянията).

Отказвайки се от свойството на Марков ние преминаваме от модел-генератор към модел-дескриптор. Вече не казваме всичко за света, а даваме само някакви статистически зависимости, които частично го описват. Например, ако знаете, че „понеделник“ се случва с вероятност  $1/7$ , това е полезно да се знае, но от това не следва, че денят след „неделя“ е „понеделник“.

## Event-driven модел

Сега ще направим най-важното обобщение. Ще заменим действията на агента с произволни събития. Това ще бъде най-същественото отказване от съвършенството, защото вече няма да следим действията на агента, които се случват на всяка стъпка, а ще следим някакви по-епохални събития, някои от които може да се случват много рядко. Моделът вече няма да променя състоянието си на всяка стъпка (при всяко действие), а само когато се случи някое от събитията, които моделът следи.

Така получените event-driven модели ще описват света много грубо, като казват само някои неща за наблюдаваните събития и нищо повече. Например, такъв модел може да ни каже дали сме вечеряли и после сме си измили зъбите или обратното. Това е важна информация, но в нашия свят има още много други неща, които този модел няма да отрази.

**Дефиниция:** Event-driven model е следното:

$S$  (множеството на състоянията)

$E$  (множеството на събитията, които моделът следи)

$Trace : S \times \Omega \rightarrow \mathbb{R}$  (вероятността да видим конкретно наблюдение в конкретно състояние)

$Event : S \times E \rightarrow \mathbb{R}$  (Вероятността определено събитие да се случи.)

$World : S \times E \times S \rightarrow \mathbb{R}$  (Вероятността за преход от едно състояние по дадено събитие към друго състояние)

Тук следата все още е съвършена. По-долу ще я обобщим и ще я направим несъвършена.

За да опростим дефиницията сме сложили функциите *Event* и *World* да връщат вероятност, а не интервал. Всъщност, в повечето случаи ще предполагаме, че тази вероятност не ни интересува, а ще гледаме само дали това може или не може да се случи (т.е. в повечето случаи функциите ще връщат интервала  $[0, 1]$  или стойностите 1 или 0).

При MDP модела описваме само света без да казваме нищо за агента (тоест оставяме го да прави каквото си иска). При MDP Fixed и при MDP Plus налагаме известни ограничения върху действията на агента. При event-driven модела описваме света заедно с агента. Идеята е, че ние сме част от света и когато изучаваме света, ние изучаваме и себе си. Ако едно събитие е невъзможно, причината за това може да е светът, а може причината да сте вие (агентът). Ако нещо не може да се случи, това може да е защото светът не го позволява, а може да е защото вие (агентът) не искате това да се случи или искате, но не можете.

Event-driven моделът е доста по-недетерминиран от MDP модела. Действията на агента са събития, които не се пресичат (не могат да се случат едновременно), докато събитията наблюдавани от event-driven модела спокойно могат да се случват едновременно. Когато това се случи, трябва да изберем по коя стрелка ще тръгнем. Тоест, имаме още една причина за недетерминираност. Освен няколко стрелки с еднакъв цвят може да имаме стрелки с различен цвят, но да не е ясно по коя трябва да тръгнем, защото и двете събития са се случили в този момент. Може event-driven моделът да избегне тази допълнителна недетерминираност, като определи приоритет на събитията и по този начин да разреши подобни колизии. Също така може в такива случаи да се използват и двете стрелки като се премине последователно първо по едната, после по другата (така ще се отчете настъпването и на двете събития). Разбира се, в този случай трябва да се уточни по коя от двете стрелки първо ще се тръгне.



Не трябва да си мислим, че при event-driven модела, ние точно знаем в кое текущо състояние се намираме. Поради недетерминираността на модела ние обикновено няма да знаем отговора на този въпрос, а ще го знаем с някаква вероятност. Затова често ще си задаваме въпроса „Къде съм?“ или „Какво се случва в момента?“ Тези въпроси може да ги сведем до въпроса „В кое от състоянията на модела се намирам?“

## Какво е следата?

Следата е това, което отличава различните състояния. Един модел без следа е безсмислен. Представете си, че всичко е сиво. Тогава има ли значение в кое състояние сме?

Следата има две функции. Тя ни дава смисъл, като ни казва какво очакваме да се случи, но тя още ни помага да разберем къде сме. (Помага ни да разберем кое е текущото състояние. Това е нужно при недетерминизъм.) Тоест, следата е съществена част от дефиницията на модела. Можем да променим само следата, като запазим всичко друго и по този начин да получим съвсем различен модел.

Като пример нека вземем свят, в който имаме едно действие и две възможни наблюдения („червено“ и „синьо“). Нека вземем модела изобразен на фигура 5.

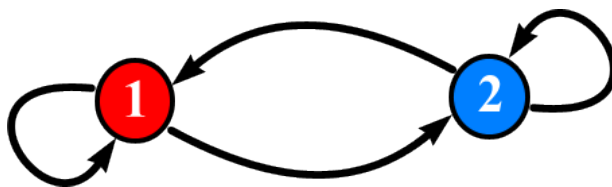


Figure 5

В този модел не сме сложили вероятности на стрелките. Предполагаме, че не знаем каква е вероятността, тоест предполагаме, че вероятността е в интервала  $[0, 1]$ . Това означава, че този модел не е MDP, а е MDP Plus.

Този модел не ни казва нищо за света. Той дори не ни казва кое е текущото състояние, макар че това можем да го разберем по това дали в момента виждаме „червено“ или „синьо“. По-важно е какво ще се случи в бъдеще или какво се е случило в миналото, а това са въпроси, по които този модел мълчи.

Все пак, този модел е интересен. Той е недетерминиран, но ако знаем какво ще видим на следващата стъпка, ние можем точно да кажем в кое състояние ще сме и обратното. Тоест, има две неща, които ни интересуват и от първото можем да получим второто, както и обратното, но няма никакъв начин да научим нито първото, нито второто. Това все пак означава, че моделът е безполезен.

Нека да променим света и наблюденията ни да станат  $\{1, 2, 3, 4\}$ . Нека следата да не е „червено“ или „синьо“, а да стане „четно“ или „нечетно“. (Тази следа е по-обща и излиза извън дефиницията на MDP Plus, която дадохме, но по-надолу ще разгледаме и по-сложни варианти за следа.)

Получаваме нов модел, който ще е в състояние 1 точно тогава когато (iff) наблюдението е четно. Добре, нека да променим следата и тя да стане „по-малко от 3“ или „по-голямо от 2“. Така ще получим съвсем друг модел, където състоянието не се определя от четността на наблюдението, а от това дали е голямо или малко.

## Несъвършена следа

Нека сега обобщим следата на модела и да се откажем от нейното съвършенство. Съвършената следа е тотална и пълна. Тотална означава, че във всяко състояние нещо се случва. Пълна означава, че знаем точно какво се случва. След това обобщение следата вече няма да е навсякъде (във всяко състояние) и няма да казва точно какво ще се случи.

Пълна следа имаме, ако знаем точно какво ще видим в състоянието. Също така, пълна следа имаме, ако имаме няколко възможности и за всяка възможност имаме точно определена вероятност. Пълнотата може да се наруши, ако вместо точно определена вероятност предположим интервал (тоест, ако допуснем известна непрогнозируема случайност). Ако разрешим интервали това ще покрие следата „четно“, която използвахме по-нагоре. В този случай „четно“ ще бъде 2 или 4, всяко с вероятност в интервала  $[0, 1]$ . Тоест, при „четно“ не знаем дали наблюдението е 2 или 4, но важното е че не е 1 и не е 3, защото за тях вероятността е 0.

Заменяйки точните вероятности с интервали ние вече можем да кажем почти всичко за следата, която се случва за една стъпка (за един момент от времето). При MDP моделите ние стоим в състоянието само една стъпка след което се случва поредното действие и преминаваме към следващото състояние. Ние създадохме event-driven моделите където вече не стоим в състоянието само една стъпка, а стоим известен период от време (докато се случи поредното наблюдавано събитие). Тоест, следата вече няма да ни казва какво се очаква да видим в един момент от времето, а какво се очаква да видим в един интервал от време. Когато говорим за интервал, можем да кажем много повече неща, отколкото можем да кажем, когато говорим за една точка. (Например, можем да кажем, че в интервала наблюдението е едно и също и не се променя.)

Най-интересното, което можем да кажем за един интервал от време е, че в този интервал ще се наблюдава някакво явление. Първо да кажем какво е явление.

## Явление

Въведохме event-driven моделите, които не са съвършени и не ни казват всичко, а описват само някаква зависимост (схема). Въпросът е дали тази зависимост се наблюдава непрекъснато или само от време на време. Да вземем зависимостта от фигура 6.

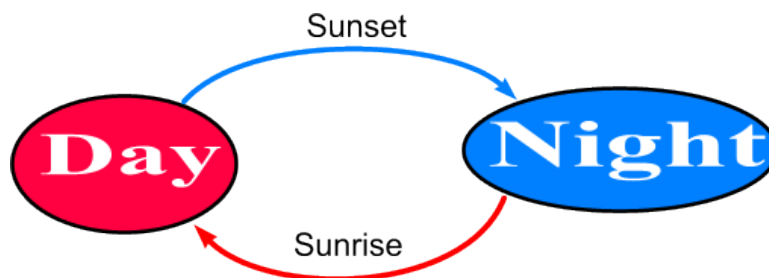


Figure 6

Тази зависимост се наблюдава непрекъснато, защото във всеки един момент или е ден или е нощ. Да предположим, че тръгнем на път към планетата Марс. Тогава слънцето ще грее непрекъснато и ще излезем от схемата ден-нощ. Тоест, има зависимости, които се наблюдават непрекъснато, но повечето зависимости не са такива.

**Дефиниция.** Явление ще наричаме event-driven модел, който не е валиден непрекъснато (през целия живот), а само през определени интервали от време.

Например явлението буря. Ако можем да опишем това явление с event-driven модел, то този модел няма да е валиден непрекъснато, а само когато има буря.

Ако вземем едно конкретно наблюдение от  $\Omega$ , то това също е явление, защото лесно можем да построим event-driven модел, при който постоянно се наблюдава само това наблюдение. (Тоест, наблюдаваме това наблюдение т.т.к. (iff) е валиден този модел.)

Следата на един event-driven модел можем да опишем с някакви явления (тоест с други event-driven модели). Следата може да бъде, че в едно състояние явлението ще се наблюдава (или че може да се наблюдава, или че с определена вероятност ще се наблюдава и т.н.).

## Следа с памет

Казахме, че заменяйки точните вероятности с интервали можем да кажем почти всичко за следата, която се случва в една стъпка. Има обаче неща, които не можем да кажем. Нека следата да има памет и да помни последното наблюдение в това състояние. Нека новото наблюдение в това състояние да е с голяма вероятност същото като предишното.

Като пример нека вземем една къща като модел на света. Нека стаите са състоянията на модела. Събитието ще бъде „преминавам от една стая в друга“. Нека в някои стаи лампата да свети, а в други да не свети. Това дали свети лампата ще е паметта на следата. Когато се връщаме в стая, в която лампата е светила, то тя вероятно пак ще свети, освен ако някой не я е загасил.

Обектите също са нещо, което трябва да го помним. Появата на обект е явление. Естествено е в някои състояния обектът да го има, а в други да го няма. Нека пак вземем къщата като пример за модел на света. Нека вътре има неподвижни обекти като мебелите и хора, които се движат от стая в стая. Тогава за мебелите няма нужда да помним къде са в момента, защото те не се движат. Тоест, мебелите можем да сложим в постоянната си памет (в следата на модела), а хората трябва да ги сложим в динамичната памет (в паметта на следата) и да помним във всеки момент кой къде е. Тоест, паметта на следата няма да е част от модела. Ще имаме постоянна памет, в която се помни моделът и динамична памет, в която се помни текущото състояние на модела и паметта на следата (в коя стая сме и къде са подвижните обекти).

## Обект

Какво е обект? В [6] дефинирахме обект като event-driven модел. Тоест, явлението „наблюдавам обекта“ го асоциирахме с обекта.

Тук ще променим дефиницията на обект. Ще заимстваме от [3], където обектът се описва като нещо абстрактно, което се характеризира с някакви свойства.

**Дефиниция.** Свойство ще наричаме явление, което се случва, когато наблюдаваме обект от групата обекти имащи това свойство.

Тоест, свойството ще бъде event-driven модел, докато обектът ще бъде абстрактно понятие, което се характеризира с някакви свойства.

Дали два обекта, които имат еднакви свойства, съвпадат? Отговорът е не. Да вземем като пример двама братя близнаци. Този пример не е много добър, защото братята не са съвсем еднакви, поне имената им са различни. Да вземе като пример два еднакви кухненски стола. Нека столовете са съвсем еднакви и ние да не можем да ги различим. Въпреки това, това са два различни обекта.

### По-малко състояния

Да се върнем към нашата първоначална цел, която беше да намалим броя на състоянията на модела. Ние създадохме event-driven модела, който описва света грубо, без да казва всичко и който ще има много по-малко състояния от модела-генератор.

Има ли връзка между модела-генератор и event-driven модела? Отговорът е да, можем да представим event-driven модела като фактор множеството на някой модел-генератор спрямо някоя релация на еквивалентност.

Какъв трябва да е този модел-генератор, за да може да го разбием на класове на еквивалентност така, че да получим event-driven модела? Грубо казано в този модел-генератор състоянията трябва да „знаят“ достатъчно. Ако състоянията на event-driven модела „знаят“ нещо което не знаят състоянията на модела-генератор, тогава ще има състояние на модела-генератор, което трябва да бъде едновременно в два различни класа на еквивалентност.

Точното изискване за модела-генератор е следното: Множеството на стрелките на модела-генератор трябва да съдържа всички наблюдавани от event-driven модела събития. Това обаче не е достатъчно. Трябва още събитието „минавам от един клас на еквивалентност в друг“ да бъде покрито от наблюдаваните събития (т.е. да бъде подмножество на обединението на наблюдаваните събития). Ако това събитие не е покрито, тогава ще може да преминем от едно състояние в друго без да се е случило никое от наблюдаваните събития.

По много сложен начин описахме изискванията си към модела-генератор, но тези изисквания не са важни, защото ние този модел няма да го търсим. Ние директно ще търсим event-driven модела, но ще знаем, че всеки такъв модел може да се представи като фактор множество на някой модел-генератор спрямо релацията на еквивалентност „двете състояния отговарят на едно и също състояние на event-driven модела“.

**Теорема 2.** За всяко събитие  $E$  съществува event-driven модел, който наблюдава това събитие.

**Доказателство:** Ще построим event-driven модел с две състояния, който описва факта, че  $E$  се е случило четен брой пъти. Ще вземем модел-генератор, който съдържа събитието  $E$  (такъв модел има, защото така дефинирахме събитие). Този модел в общия случай не можем да го разбием на класове на еквивалентност разделени от  $E$ , но ние ще построим друг модел-генератор.

Аналогично като при доказателството на Твърдение 1 ще построим модел еквивалентен на първия като удвоим състоянията му. Всяко състояние  $s_i$  ще заменим с  $s_i'$  и  $s_i''$ . Всяка стрелка  $s_i \rightarrow s_j$  ще заменим с две стрелки:

$$s_i' \rightarrow s_j'' \text{ и } s_i'' \rightarrow s_j', \text{ ако } (s_i \rightarrow s_j) \in E$$
$$s_i' \rightarrow s_j' \text{ и } s_i'' \rightarrow s_j'', \text{ ако } (s_i \rightarrow s_j) \notin E$$

Двата класа на еквивалентност ще бъдат множествата  $S'$  и  $S''$ , където  $S' = \{s \mid \exists i : s = s_i'\}$  и аналогично  $S''$ . ■

Разбира се, съществува не един, а много event-driven модели, които наблюдават събитието  $E$ .

Показахме каква е връзката между модела-генератор и event-driven модела. Ние можем да получим event-driven модела като фактор множество от някой генератор, но ние няма да тръгнем по този път, а ще го построим директно използвайки реални събития.

## Реални събития

Когато търсим event-driven модел ние трябва да си изберем няколко събития, които ще са наблюдаваните събития в този модел. Трябва да дефинираме тези събития по някакъв начин и да се научим да ги откриваме (да отчитаме когато се случват).

Казахме, че няма да търсим модел-генератор и да го разбиваме на класове на еквивалентност. Вместо това ние директно ще търсим event-driven моделите. За целта трябва да се научим да откриваме събития. Няма да използваме теоретичната дефиниция на събитие, която е неприложима в практиката. Вместо това ние ще търсим събитията по два начина – пряко и косвено.

Пряко ще ги откриваме с характеристична функция, а косвено ще ги откриваме чрез следата (тоест, по това което се случва, ще заключим, че сме преминали в друго състояние и оттам, че се е случило събитие).

Прякото откриване ни дава точния момент, в който събитието се е случило и ни дава възможност да описваме модели с примки (тоест, може да се случи събитие без да се променя състоянието). При косвеното откриване точния момент на настъпване е по-трудно да се определи, защото това че се променя следата може да се забележи веднага, а може и след няколко стъпки. Освен това при косвеното откриване не можем да забележим примка, защото когато състоянието остава същото и следата е същата.

При прякото откриване си избираме една характеристична функция и започваме да търсим следа. Тоест, предполагаме, че знаем кога събитието се е случило и се опитваме да

открием периоди от време преди и след настъпването му, които по някакъв начин да са специфични (т.е. нещо специално да се случва в тези периоди).

При косвеното търсене е обратното. Първо търсим специфични периоди. Тези периоди асоциираме с различни състояния на event-driven модела и търсим събития, които да се случват на границата на тези периоди (т.е. търсим характеристични функции). Не е задължително да намерим характеристична функция, която да опише прехода. Може нашето събитие да си остане „невидимо“, т.е. такова което да може се забележи само косвено.

## Характеристична функция

Класическата характеристична функция връща стойностите 0 и 1. Тук ние ще приемем, че тя връща вероятност, защото би било твърде ограничително да разглеждаме само характеристични функции, които точно да ни казват дали едно събитие се е случило или не. Предпочитаме функцията да може да каже, че събитието се е случило с някаква вероятност.

Ще предполагаме дори, че характеристичната функция връща вероятностен интервал. Например, ако тя ни казва, че събитието се е случило с вероятност по-голяма от  $1/2$ , тогава тя ще ни върне интервала  $[1/2, 1]$ . Ако характеристичната функция нищо не може да ни каже, тогава тя ще върне интервала  $[0, 1]$ .

Какъв ще бъде аргументът на характеристичната функция? От какво ще зависи дали събитието се е случило? Аргументите ще са възможно развитие на миналото (такова, което се е случило) и възможно развитие на бъдещето (такова, което ще се случи).

Защо избрахме характеристичната функция да зависи от миналото и от бъдещето? Не е ли по-добре да зависи само от миналото? Действително, ние предпочитаме, когато събитието се случи, ние вече да знаем, че то се е случило. Тоест, предпочитаме характеристичната функция да зависи само от миналото, но често се случва да имаме събитие, за чието настъпване научаваме по-късно (някъде в бъдещето) и затова дефинираме характеристичната функция да зависи и от миналото и от бъдещето.

Какво ще правим, ако за два различни интервала от време характеристичната функция ни връща различни стойности? Ще предположим, че стойността от по-големия интервал е по-достоверна. Може дори да предположим, че когато интервалът от време се увеличава, характеристичната функция се уточнява (вероятностният интервал се свива). Това последното няма да го предполагаме, защото понякога, когато получим повече информация ние вместо да станем по-уверени, ставаме по-неуверени.

Пример за характеристична функция е тази, която описва едно от действията на агента. Тази функция ще гледа единствено кое е следващото действие на агента и ще върне 1 или 0 в зависимост от това дали действието е било това или някое друго. Тоест, събитията на MDP модела могат да се опишат с характеристични функции.

## Заклучение

Идеята на тази статия е, че светът не може да бъде разбран напълно и че ако искаме да построим модел, ние трябва да се откажем от съвършенството и да търсим прости модели, които описват света частично.

Друга идея на статията е, че един единствен прост модел не може да ни каже всичко за света и затова ние ще трябва да търсим много различни модели, всеки от които описва част от света (някаква зависимост, свойство или явление). Разбира се, многото прости модели, които ще намерим, пак няма да ни кажат всичко, но се надяваме да ни кажат достатъчно.

Ние въведохме event-driven моделите. Това са моделите, които описват малка част от света. Пример за такъв модел е даден на фигура 6. Единственото, което този модел ни казва, е дали е ден или нощ. Това е важна информация, но тя е съвсем недостатъчна, за да разберем целия свят, защото в света освен това дали е ден или нощ има още много други важни неща.

Както казахме, на базата на event-driven моделите ние ще опишем различни зависимости, явления и свойства. По-нататък, на базата на тези явления и свойства ние ще създадем абстракции каквито са обектите и агентите. Както казахме, свободната воля на света си я обясняваме с агенти, които живеят в този свят. Когато в света има човек, за него ще си мислим като за обект и като за агент. Той е обект, когото можем да наблюдаваме и агент, чиито действия можем да забележим.

Казахме, че вместо един съвършен MDP модел ще търсим много простички event-driven модели. Колко простички ще са тези модели? Броят на наблюдаваните от модела събития обикновено ще е в порядъка на едно-две. Броят на състояния на event-driven модела ще е в порядъка на десетина. Въпросът е, как с толкова прости модели ние ще можем да опишем един сложен свят? Отговорът е, че ще строим моделите йерархично. Ще използваме по-прости модели и ще прави от тях по-сложни. Казахме вече, че следата на event-driven модела може да се характеризира с явления, които са други, по-прости event-driven модели. Също така, ако един модел попадне в определено състояние, то това е събитие, което може да бъде наблюдавано от друг модел (тоест, да се използва за създаването на по-сложен event-driven модел).

Ако искаме да намерим модел на света, ние трябва да се откажем от идеята да търсим съвършения модел на света. Ще завършим с афоризма на Волтер „Perfect is the enemy of good“. Оказва се, че тази сентенция е валидна и за Изкуствения Интелект.

## References

- [1] Xi-Ren Cao (2005). Basic Ideas for Event-Based Optimization of Markov Systems. *Discrete Event Dynamic Systems: Theory and Applications*, 15, 169–197, 2005.
- [2] Xi-Ren Cao, Junyu Zhang (2008). Event-Based Optimization of Markov Systems. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, VOL. 53, NO. 4, MAY 2008.

[3] Tatiana Kosovskaya (2019). Isomorphism of Predicate Formulas in Artificial Intelligence Problems. *International Journal "Information Theories and Applications"*, Vol. 26, Number 3, 2019, pp. 221-230.

[4] Dimiter Dobrev (2017). How does the AI understand what's going on. *International Journal "Information Theories and Applications"*, Vol. 24, Number 4, 2017, pp.345-369.

[5] Dimiter Dobrev (2019). Minimal and Maximal Models in Reinforcement Learning. *International Journal "Information Theories and Applications"*, Vol. 26, Number 3, 2019, pp. 268-284.

[6] Dimiter Dobrev (2019). Event-Driven Models. *International Journal "Information Models and Analyses"*, Volume 8, Number 1, 2019, pp. 23-58.