Минимален и максимален модел при Reinforcement Learning

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

Всеки тест ни дава едно свойство, което ще наречем резултата на теста. Продължението на това свойство ще наречем тестовото свойство. Въпросът е, какво представлява това свойство? Дали това не е свойство на състоянието на света? Отговорът е и да и не. Ако вземем произволен модел на света, то отговорът е не, но ако изберем максималния модел на света, то тогава отговорът е да. Имаме различни модели. Минималният модел е този, при който света знае за миналото и за бъдещето минималното, което му е нужно. При максималния модел света знае всичко за миналото и за бъдещето. При този модел, ако хвърлите зар света знае какво ще се падне и дори знае вие какво ще направите. Например, знае дали ще хвърлите зара.

Keywords: Artificial Intelligence, Reinforcement Learning, Partial Observability, Event-Driven Model, Double-State Model, Test Property, Test State.

Въведение

Ние се опитваме да разберем света. За да го разберем, ще използваме тестове. Пример за тест е:

Ако натисна дръжката на врата $\Rightarrow$ вратата ще се отвори.

Тестовете съдържат някакво условие (предпоставка). В случая условието е, че съм натиснал дръжката на вратата. Когато условието е изпълнено, тогава теста е проведен и тогава получаваме резултата на теста, който е истина или лъжа. В случая врата се отваря или не се отваря.

Всеки тест ни дава едно свойство (резултата на теста). В случая свойството е „врата е заключена“. Ние не знаем каква е стойността на това свойство във всеки един момент, а само в моментите когато теста е проведен.

Ние ще предполагаме, че това свойство има смисъл във всеки един момент и ще се опитаме да продължим характеристичната функция на това свойство извън множество на моментите когато сме провели теста. Възможно е това свойство да не е тотално и да не е дефинирано във всеки един момент. Например, ако някой открадне врата, то въпросът дали тя е заключена губи своя смисъл. Тоест, ще се опитаме да продължим характеристичната функция на свойството, но не е задължително да получим тотална функция.

Каква е идеята в продължаването на резултата на теста до тестово свойство? Ако ви дам едно резенче (slice) от краставица, дали бихте могли от резенчето да възстановите цялата краставица? Разбира се, става дума за мислено възстановяване, а не за физическо.

Това възстановяване не е единствено и при него може да се получат различни обекти. Например, от резенчето от краставица можете да получите носорог, като си представяте, че това е резен от рога му. Разбира се, ние ще търсим продължения на резултата, които са максимално прости, естествени и достоверни.

Имайте предвид, че резенчето от краставица е нещо реално, докато самата краставица е нещо въображаемо и измислено. Ако ви дам да видите цялата краставица, то на вас ще е ви е по-лесно да си я представите, но вие може да си представите здрава краставица, а тя да се окаже изгнила отвътре. Тоест, никога вие не получавате цялата информация. Винаги получавате само една част (един резен информация), по който трябва да си представите целия обект.

Нека си зададем въпроса какво представлява това свойство (резултата на теста и неговото продължение). В различни моменти от времето свойството е истина или лъжа. Въпреки всичко, това не е свойство на времето, защото свойството не зависи толкова от конкретния момент, колкото от развитието на историята около този момент. Може би това е свойство характеризиращо състоянието на света (т.е. това е множеството от състоянията на света, при които свойството е истина).

Дали зависи свойството от миналото и от бъдещето? Обикновено тестът не се провежда в един конкретен момент, а в продължение на известно време. Тоест, резултата от теста зависи от времевия контекст (близкото минало и бъдеще в рамките на провеждането на теста). Ако вземем тестовото свойство, то то зависи от по-широк времеви контекст и може да ни казва нещо за далечното минало и бъдеще на света.

Например свойството: „В писмото има добра новина“. Съответния тест е: „Отварям писмото и проверявам какво пише в него“. Това свойство ни казва нещо за бъдещето. По точно, казва ни какво ще прочетем в писмото, когато го отворим. Дали това е свойство на света? Дали света знае какво пише в писмото още преди да сме го отворили. Обикновено си мислим, че знае, но би могъл е да не знае. Например, повечето компютърни игри не си дават труда да изчисляват целия свят, а се грижат само за тази част от света, която играчът вижда в момента. Ако света е такава игра, то той ще реши какво пише в писмото чак когато го отворите. Друг пример, нека живота е телевизионен сериал. Ако в 1354 серия вие получите писмо и десет серии по-късно го отворите, то кога сценариста ще реши какво пише в писмото? Когато пише сценария на серията когато получавате писмото или когато пише серията когато го отваряте? Тоест, виждате, че света може предварително да знае, а може и да не знае предварително какво ще се случи в бъдеще.

Подобен е и въпросът с миналото. Например свойството: „Върнал съм се от почивка“ ни дава информация за миналото. Съответния тест е: Проверявам дали съм на почивка или в командировка и след това се връщам. Предполагаме, че ако сте се върнали от почивка и още сте си в къщи (никъде другаде не сте ходили), то продължавате да сте се върнали от почивка. Тоест, продължихме свойството за моменти, в които то не е тествано.

Нека предположим, че за бъдещето на света няма никакво значение дали сте се върнали от почивка или от командировка. Тогава има ли смисъл света да помни този факт? Въпросът, който вълнува историците е дали света помни миналото? Дали едно историческо събитие е оставило някакви документи или други следи за това, че се е случило? Отговорът е, че миналото може да се помни, а може и да не се помни.

В тази статия ще разгледаме два модела на света – минимален и максимален. При минималния светът помни минималното от миналото и знае минималното за бъдещето. При максималния е обратното, там светът помни всичко от миналото и знае всичко за бъдещето. Там светът знае точно какво ще се случи и дори знае вие (агента) какво ще направите.

Какво търсим?

Какво е дадено и какво се търси? В тази статия ще търсим обяснение на света. При Reinforcement Learning [1] имаме един агент, който живее в някакъв свят. Света е един ориентиран граф подобен на този на фигура 1. Агентът се движи от състояние на състояние по стрелките и събира някакви награди (rewards). За него ще е много важно да разбере света, за да може да намери наградите. Има много статии, където се предполага, че света е даден и се търси стратегия, която би била успешна в този свят (например в [3]). В тази статия ще предполагаме, че света е неизвестен.

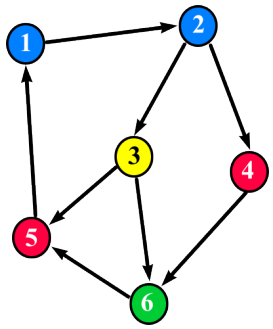

Figure 1

На фигура 1 възможните действия на агента са отбелязани със стрелки, а възможните наблюдения са отбелязани с различни номера и цветове. На всяка стрелка би трябвало да има етикет (label), който да казва на кое действие съответства тази стрелка. На тази фигура етикетите са изпуснати, но се вижда, че понякога има една възможна стрелка, а понякога възможностите са няколко (при състояния 2 и 3). Ще предполагаме, че не всички действия са възможни, както и че имаме недетерминирани преходи. Тоест, ще предполагаме, че при конкретно състояние и конкретно действие може и да няма стрелка с този етикет излизаща от това състояние, както и че може такива стрелки да има повече от една.

Това, че допускаме недетерминирани преходи означава, че допускаме случайност. В [5] показахме, че има два вида случайност – прогнозируема и непрогнозируема. Тук използваме непрогнозируемата случайност (нещо се случва с вероятност в интервала [0, 1]). Тази случайност се използва още и при NFA (nondeterministic finite automaton). Там нещо може да се случи, а може и да не се случи, но не знаем с каква вероятност ще се случи. При POMDP (partially observable Markov decision process) се използва прогнозируемата случайност (нещо се случва с точно определена вероятност). В [5] доказахме еквивалентност между четирите вида модели (детерминирания, моделите с двете случайности и модела с комбинацията от двете случайности). Тоест, можем да работим, с който модел ни е най-удобно и ние сме избрали този.

Казваме, че имаме Full Observability когато можем по това, което виждаме да познаем кое е състоянието. Съответно, казваме че имаме Partial Observability в противния случай. Например, ако виждаме номера на състоянието, тогава имаме Full Observability, но ако виждаме само цвета, тогава не виждаме всичко. Например, ако виждаме „червено“ не можем да кажем дали сме в състояние 4 или в 5.

Ако имаме модела на света, то чрез него бихме могли да предскажем бъдещето. Например ако сме в състоянието 4 можем да предскажем, че следващото състояние ще е 6. Ако сме в червено състояние, можем да предскажем, че следващото ще е зелено или синьо. По същият начин както предсказваме бъдещето можем да предвидим и миналото. Трябва само да обърнем посоката на стрелките и милото става бъдеще. Единственият проблем е, че при обръщане на стрелките от детерминиран граф може да се получи недетерминиран, но ние избрахме да използваме като модели недетерминирани графи.

Какво ни е дадено? Дадена ни е историята до текущия момент. Тоест дадена ни е редицата от действия (outputs) и наблюдения (inputs или view).

$$a_1, v_2, a_3, v_4, \dots, a_{t-1}, v_t$$

Тук номерацията не показва номера на стъпката, а номера на момента. Във всяка стъпка има два момента. В първия извеждаме информация (това е нашето действие), а във втория въвеждаме какво виждаме. Тоест, номера на стъпката ще е номера на момента разделен на две.

Предпочитаме да разделим времето не на стъпки, а на моменти, заради събитийните модели където състоянията ще се променят в определени моменти. За една стъпка състоянието ще може да се промени два пъти, защото в стъпката има два момента.

Моментите ще са два вида – входящи и изходящи, които ще наричаме още четни и нечетни моменти.

Нека отбележим, че входовете и изходите ще са вектори от скалари. Ще предполагаме, че тези скалари са крайни. Бихме могли да се ограничим до булеви вектори, но няма да го направим за да избегнем излишното кодиране (виж [4]).

Към историята ще добавим още и некоректните ходове, които сме пробвали преди да изиграем поредния си ход.

$$bad_1, a_1, v_2, bad_3, a_3, v_4, \dots, bad_{t-1}, a_{t-1}, v_t$$

Тук елемента ***bad*** можем да си го мислим като списък от некоректни ходове или като множество, защото реда, в който сме пробвали некоректните ходове, е без значение. Ще считаме, че некоректните ходове са пробвани в същия момент, когато е изигран съответния коректен ход (затова списъка ***bad*** и следващия след него коректен ход ***a*** имат еднакъв индекс).

Дефиниция: Живот ще наричаме история, която не може да бъде продължена.

Една история не може да бъде продължена, ако е безкрайна или ако завършва със състояние, от което не излизат никакви стрелки. В [6] такива състояния нарекохме „внезапна смърт“.

Двоен модел

Фигура 1 е направена на базата на стандартния модел, който е базиран на стъпки. Ние ще използваме двойния модел, който е базиран на моменти.

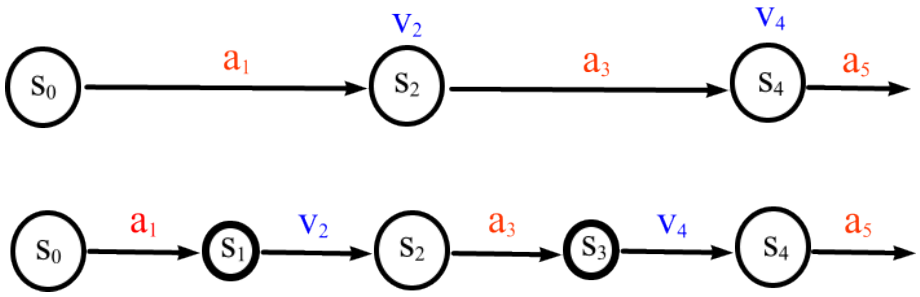

Figure 2

Разликата между стандартния и двойния модел е илюстрирана на фигура 2. Двойния модел се получава от стандартния като всяко състояние бъде заменено от две състояния и стрелка помежду им. Всички стрелки, които са влизали в състоянието, сега ще влизат в първото, а стрелките които са излизали от състоянието, сега ще излизат от второто. Етикета, който е бил на състоянието сега е етикет на стрелката между новите състояния. При двойния модел само стрелките имат етикети, а състоянията нямат. Тук сме отбелязали нечетните състояния с по-малки кръгове, за да подчертаем, че имаме два вида състояния.

Забележка: Тук ще считаме, че на състоянието съответства времеви период, а на стелката съответства само един миг, който е началото на следващия времеви период. При събитийните модели ще считаме, че на състоянията съответстват дълги периоди от време, а на стрелките съответстват съвсем къси периоди (от един момент или малко повече).

Изглежда сякаш в двойния модел състоянията са два пъти повече, но това не е така, защото ако две състояния са еквивалентни от гледна точка на бъдещето, то те мога да бъдат обединени (слети) в едно. В стандартния модел могат да бъдат обединени само състояния, които са еквивалентни от гледна точка на настоящето и на бъдещето. Тоест, в стандартния модел състоянието трябва да помни настоящето, докато в двойния няма такава нужда. Това е една от причините, поради която въвеждаме този модел. В тази статия за нас ще е много важно каква е информацията, която можем да извлечем от състоянието. Тоест, какво може да каже състоянието за миналото и за бъдещето. Настоящето е част от миналото, защото то вече се е случило.

В двойния модел състоянията ще са два вида. Състояния след вход и състояния след изход. Ще ги наричаме още четни и нечетни състояния. По-важни са четните състояния, защото това са състоянията, в които мислим. В нечетните състояния ние не мислим, а само чакаме да видим каква информация ще ни даде света. (Може да предполагаме, че в

нечетните моменти мисли светът. Така се редуваме, в един момент мислим ние, а в следващия мисли светът.)

Да вземем като пример света, в който играем шах срещу въображаем противник. Нашите действия ще са вектори от вида (x_1, y_1, x_2, y_2). Тези вектори описват нашия ход. Ще виждаме хода на противника и оценката (reward). Тоест, входа ще бъде вектор от вида (x_1, y_1, x_2, y_2, R). При двойния модел състоянията ще бъдат позициите на дъската. Четните състояния ще са тези, при които белите са на ход (т.е. ние сме на ход). Когато нашия ход завършва играта (например мат), тогава противника ще трябва да играе някакъв празен ход и да ни даде само оценката на играта. Т.е. вектор от вида (0, 0, 0, 0, R). Този празен ход трябва да премине към началната позиция, когато играта започва отначало.

При стандартния модел нещата ще са много по-сложни. Тогава състоянията ще са само позициите, при които белите са на ход, но ще трябва да се помни още и хода на противника довел до тази позиция. Тоест, състоянията ще са повече, защото позициите когато белите са на ход са два пъти по-малко, но ако всяка от тях може да се получи по средно 100 начина (от 100 различни позиции с 100 различни хода на противника), тогава, като теглим чертата получаваме, че стандартният модел ще има 50 пъти повече състояния.

Както казахме, при стандартния модел настоящето се налага да се помни, а в двойния не се налага. Да видим как стои въпросът с бъдещето. Ако играем шах срещу детерминиран противник, то тогава и двата модела ни дават детерминиран граф. Нека допуснем, че противника е недетерминиран и че за всяка позиция има по няколко възможни хода, които той може да изиграе. Тогава стандартния модел ни дава недетерминиран граф (един наш ход води до няколко различни ответни хода и съответните им позиции). Двойния модел пак си е детерминиран, защото нашия ход води до една определена позиция. От тази позиция излизат няколко възможни стрелки, които съответстват на възможните отговори на противника (ако противникът беше детерминиран щеше да излиза само една стрелка.)

Винаги ли двойният модел ни дава детерминиран граф? Не винаги, но винаги можем да го сведем до детерминиран. Да си представим същата игра, но сега няма да виждаме хода на противника, а само ще виждаме коя фигура е преместил. Тоест, няма да виждаме (x_2, y_2). Сега, когато знаем коя е фигурата, но не и къде е преместена, ще имаме няколко възможни позиции. Естествено би било да представим тази игра с недетерминиран граф, но можем да го направим и с детерминиран граф. Ако състоянието на света не е конкретна позиция на дъската, а е множество от възможни позиции, тогава на конкретния ход ще отговаря една единствена стрелка, която ще води до множество от възможни позиции. Тоест, при първия (недетерминирания) вариант света знае нещо повече за бъдещето, отколкото при втория. При детерминирания вариант света не знае коя точно е позицията. Знае кое е множеството от възможните позиции, а коя е точно позицията ще разбере по-късно, когато това проличи по входа (по наблюденията). Това е като с писмото. В единия случай света знае какво пише в писмо, а във втория случай ще реши какво пише, чак когато ти отвориш писмото.

Минимален модел

Един двоен модел на света ще го наричаме минимален, ако света знае за миналото и бъдещето само толкова колкото му е необходимо. При минималния модел, ако две

състояния са еквивалентни спрямо бъдещето, то те съвпадат. Тоест, не се помни нищо от минало, което не ни е нужно, за да определим бъдещето.

Освен това, минималния модел е детерминиран граф. Това означава, че разклоненията са избутани максимално напред към бъдещето. Тоест, всяко нещо за бъдещето се разбира чак когато му дойде времето (когато то повлияе на наблюдението, но не по-рано).

Минимален модел не значи с най-малко състояния. Спрямо миналото минималността действително намалява състоянията, но спрямо бъдещето по-скоро ги увеличава (защото преминаваме от конкретни възможности към множества от конкретни възможности).

Процедурата по детерминизация винаги може да се приложи и винаги можем да получим детерминиран граф. Това, че графа е детерминиран не означава, че агента е детерминиран или че света е детерминиран. За да бъде агента детерминиран, трябва да няма разклонения в четните състояния. (Тук под детерминиран агент имахме предвид, че е принуден да играе детерминирано, защото има само един коректен ход. Обикновено под детерминиран агент разбираме такъв, който играе детерминирано без да е принуден да го прави.) За да бъде света детерминиран трябва да няма разклонения в нечетните състояния. Това, че графа е детерминиран означава, че състоянията знаят за бъдещето минималното. (Ако две състояния са различни, то те имат различно минало. Т.е. различни са защото имат различно минало, а не защото знаят нещо повече за бъдещето.)

Тотален модел

Когато един ход е некоректен стрелката по този ход просто липса и този ход просто не може да бъде направен. Бихме искали агента да може да пробва некоректните ходови и като ги пробва нищо да не се случи. Т.е. той да получи информация, че хода е некоректен, но да остане в същото състояние.

Тази цел ще я постигнем като добавим към всяко четно състояние (при което има некоректни ходове), по едно допълнително нечетно състояние (виж фигура 3). Всички некоректни ходове от четното състояние ще ги насочим към нечетното. От там ще се върнем обратно с една стрелка с етикет „bad“. Този етикет ще е един специален нов вектор, който сме добавили за целта и който ще получаваме като вход само когато опитаме некоректен ход.

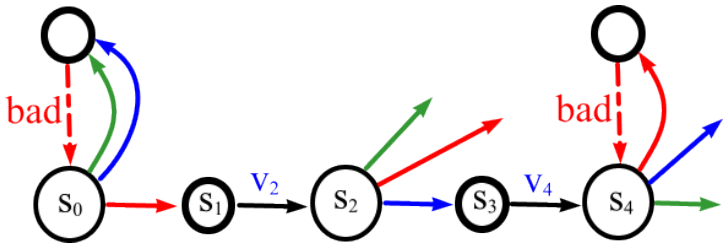

Figure 3

На фигура 3 възможните ходове са отбелязани със стрелки с червен, син и зелен цвят. В състоянието s_0 има два некоректни хода, а в състоянието s_4 има само един. В състоянието

s_2 няма некоректни ходове и затова за него не сме добавили допълнително нечетно състояние.

По този начин ще получим един тотален модел, при който всички ходове могат да се пробват, но коректните ходове реално променят състоянието на света, докато некоректните само дават информация, че са некоректни. Тоест, получаваме един нов тотален модел, който описва същия свят като предишния модел, с тази разлика, че некоректните ходове вече могат да се пробват.

Максимален модел

След като имаме минимален модел, може би има и максимален. Това трябва да е модела, при който състоянието знае всичко за миналото, всичко за това кои са коректните ходове и всичко за бъдещето.

Да знаем всичко за миналото е лесно, просто когато вървим назад (срещу стрелките) не трябва да има разклонения. Тоест, ако модела има формата на дърво, тогава състоянието ще знае всичко за миналото (тръгвайки от него назад ще можем да възстановим цялата история).

Да знае състоянието кои са некоректните ходове също не е трудно, защото това е крайна информация, която лесно ще добавим.

За да знаем всичко за бъдещето, трябва да няма недетерминираност. Тоест, ако хвърлим зар, трябва да знаем какво ще се падне. Ще построим модел, при който цялата недетерминираност ще е концентрирана в началното състояние. След като изберем началното състояние (по недетерминиран начин) нататък всичко ще е детерминирано.

Нека вземем дървото на всички достижими състояния. (Това е дърво, ако има еквивалентни състояния не ги сливаме в едно.) Ще детерминираме това дърво, макар това последното да не е задължително. От така полученото дърво ще получим всички стратегии (policy) на света. Това са поддървета, в които няма разклонения по наблюдението, а разклоненията по действията се запазват. Тези дървета са много (the cardinality of the continuum). От всичките тези дървета правим модел, където началните състояния ще са корените на всичките тези дървета.

Тук думата стратегия не е много подходящо да се използва, защото казваме стратегия, когато имаме някаква цел. Тук предполагаме, че агента има цел, а света няма цел. Ако предположим, че целия свят се опитва да ни помогне или да ни попречи, то това би било твърде егоцентрично. Въпреки това, ще предполагаме, че в света има агенти, които имат своите цели. Тоест, света няма цел, но въпреки това различните поведения на света ще наричаме стратегии.

Следователно направихме модел, който се състои от всички стратегии на света. Още преди да започне живота света избира по случаен начин една от своите стратегиите и я следва до края на живота на агента. Идеята е, още преди да започне играта да намислим как ще играем. Можем да намислим, че ако той ни играе с коня, ние ще отговорим с офицера и т.н. Едно такова намисляне представлява стратегия и се изразява с едно безкрайно дърво.

Тези дървета са неизброимо много. Може да си намислим, че ще играем използвайки определена детерминирана програма, но по този начин ние може да си намислим само някоя изчислима стратегия, а те са много по-малко (само изброимо много).

Полученият модел е еквивалентен на модела, от който тръгнахме, защото всяка история, която е възможна при единия модел е възможна и при другия. При новия модел света се държи детерминирано, с изключение на първия момент, когато избираме началното състояние.

В този свят, ако вземем произволно състояние, то то знае почти всичко за бъдещето, с изключение на това, че не знае какво действие ще избере агента. Бихме искали да направим модел, при който дори и това да се знае от състоянието на света.

Тук има един проблем. Обикновено предполагаме, че света е даден, а агента е произволен. Тоест, няма как света да знае какво ще направи агента, защото той си няма идея кой агент ще му се падне. Сега ще предположим, че агента е фиксиран и че света би могъл да знае нещо за агента. Например, света би могъл да знае, че в определена ситуация агента няма да изиграе определен ход, макар че този ход е коректен и би могъл да бъде изигран. Това, че определен ход няма да бъде изигран от агента ще го отбележим с липсваща стрелка в графа на модела.

Сега ще предположим, че още преди да започне живота света и агента са решили как да играят. Тоест, избрали са своята стратегия, която ще следват до края на живота на агента. Това може да се опише с наредена двойка от две безкрайни дървета или чрез един живот (безкраен път в дървото). Това е така, защото резултата от прилагането на две фиксирани стратегии е един фиксиран път.

Така получихме максималния модел на света. Той се състои от всички възможни животи (това са пътища в дървото на достижимите състояния). Единственото, което ни липсва, това е информацията за некоректните ходове. За целта ще добавим примки (loops), както направихме, когато правехме модела да е тотален. Тук няма да получим тотален модел, защото ще добавим примки само по некоректните ходове, а не по всички липсващи стрелки. Така получаваме модела изобразен на фигура 4. (При s_2 няма примка, защото от това състояние не излизат некоректни ходове)

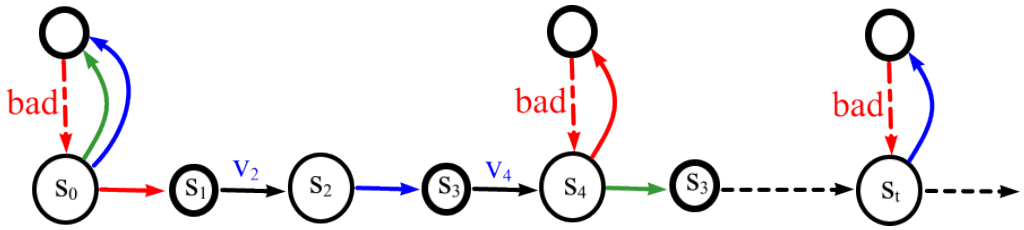

Figure 4

На тази фигура е изобразен само един живот, а не всички възможни животи, които са част от максималния модел. Нас ни интересува само един живот и това е живота, който живеем. Затова може да предполагаме, че максималния модел съдържа само един живот и това е живота, който ни интересува. По този начин ще загубим еквивалентността с първоначалния модел, но получения модел е това, което ни трябва, защото другите възможни животи не са важни.

В така получения максимален модел от всяко състояние може да се възстанови цялата история и дори целия живот. Пътищата напред и назад са без разклонения. Примките, които добавихме, няма да ги броим за разклонения, защото символа *bad* не се среща при наблюденията след коректен ход. Тук състоянието знае кои ходове са некоректни, но не знае кои от тях агента е пробвал. Тази информация не е важна за света, защото от нея не зависи нито миналото нито бъдещето. Разбира се, тази информация е важна за агента, защото той може да не знае кои ходове са некоректни и затова му е полезно да знае кои ходове вече е пробвал.

Заключение

Опитваме се да разберем света, тоест да намерим неговия модел. Проблемът е, че този модел въобще не е единствен. Може да имаме недостижими и еквивалентни състояния, но това не е проблем. Може да има части на света, които никога не сме посещавали и които никога няма да посетим. Например, има ли живот на Марс? След като никога не сме ходили там и никога няма да отидем, то това е без значение (тоест, това няма да се отрази на нашия живот).

Най-съществения проблем е, че имаме модели, в които света знае повече и модели при които света знае по-малко. Кой модел търсим? Отговорът е, че търсим максималния модел. Тоест, ние ще се стремим максимално да разберем миналото и максимално да предскажем бъдещето. Ще се опитаме да предскажем дори и собственото си поведение. Ние сме част от света и за да го разберем трябва да се опитаме да предскажем дори и собственото си поведение.

За да опишем максималния модел ние ще използваме така наречения разширен модел. В този модел ще представим състоянието на света като вектор с огромен брой координати (променливи). Те ще са хиляди или дори безбройно много. От теоретична гледна точка, те ще са безбройно много, но на практика ще изберем само най-интересните от тях. Може би това е модела, за който говори Sutton в [2] (state representation which contains many state variables).

Първите координати (променливи), които ще сложим във вектора описващ разширеното състояние, това ще е това, което виждаме в момента. При двойния модел наблюдението не е функция на състоянието на света, защото, ако състоянието е четно, то може да има много стрелки с различни наблюдения, които да влизат в него. Ако състоянието е нечетно, то съответно може да има много стрелки, които да излизат от него. Тук обаче говорим за максималния модел и при него винаги има само една влизаща и една излизаща стрелка. Тоест, при максималния модел какво виждаме в момента се определя от състоянието на света.

Към това, което виждаме в момента, ще добавим още тестовите свойства на различни тестове. Това не са резенчетата от краставица, които са нещо реално. Това са измислените краставици, които ние сме измислили на базата на реалните резенчета. Тоест, разширения модел няма да е нещо реално, а ще е нещо измислено.

Въпрос: Ако имаме Full Observability, тогава има ли нужда да добавяме тестови свойства към вектора на състоянието на разширения модел? Отговор: Да има нужда, защото при Full Observability ние знаем кое е състоянието на света, но това е състоянието при някой модел, който не е максималният. Ако имаме Full Observability с максимален модел, то света е толкова елементарен, че чак не е интересен.

Как да измислим едно тестово свойство? Помислете си за свойството „заключена ли е вратата“. Пробваме и получаваме ту „заключено“, ту „отключено“. Много трудно е да направим модел и да разберем кога е заключено и кога отключено. Много по-голям успех ще имаме, ако допуснем, че вратите са повече от една (особено, ако те действително са повече). В новият модел трябва да имаме идея пред коя врата сме застанали в момента. Тогава може някои врати да са постоянно отключени, други постоянно заключени, а трети да променят състоянието си по някакви правила. В този случай няма да добавим към разширения модел една променлива, която да отразява едно тестото свойство. Ще добавим много променливи, по една за всяка врата (която да отразява свойството на вратата) и една променлива, която да отразява пред коя врата сме в момента. Това представяне в [5] беше наречено тестово състояние.

Пред коя врата сме застанали в момента се определя от събитийните модели, които ще разгледаме в следващата статия.

References

[1] Richard Sutton, Andrew Barto (1998). Reinforcement Learning: An Introduction. *MIT Press, Cambridge, MA (1998).*

[2] Richard Sutton (2008). Fourteen Declarative Principles of Experience-Oriented Intelligence. www.incompleteideas.net/RLAIcourse2009/principles2.pdf

[3] Johan Åström. (1965). Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications. 10: 174–205.*

[4] Dimiter Dobrev (2013). Giving the AI definition a form suitable for engineers. *April, 2013.* arXiv:1312.5713.

[5] 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.*

[6] Dimiter Dobrev (2018). The IQ of Artificial Intelligence. *Jun 2018. arXiv:1806.04915.*