

Некоректни ходове и проверими състояния

20 април, 2017 г.

Как да опишем невидимото? Нека имаме една последователност: вход, изход, вход, изход ... Зад тази последователност стои някакъв свят и редицата от неговите вътрешни състояния. Ние не виждаме вътрешното състояние на света, а само част от него. За да опишем тази част, която е невидима, ние ще използваме понятието „некоректен ход“ и неговото обобщение „проверимо състояние“. По този начин ние ще сведем задачата за partial observability към задачата за full observability.

Keywords: Artificial Intelligence, Machine Learning, Reinforcement Learning, Partial Observability, Diagnosability.

Въведение:

В областта на Reinforcement learning първата ни задача е да опишем текущото състояние на света (или текущата позиция на играта). Когато виждаме всичко (full observability) тогава този въпрос не стои, защото в този случай входа напълно описва състоянието на света. По-интересен е случая когато виждаме само част от света (partial observability). Тогава ние ще добавим към вектора на входа още координати и по този начин ще получим нов вектор, който вече изцяло ще описва състоянието на света.

За да добавим тези нови координати първо ще въведем понятието „некоректен ход“. Идеята за това, че не всички ходове са коректни не е нова. Други автори предполагат, например, че не можеш да дадеш повече пари от тези, които имаш. Тоест, предполагат че изхода е ограничен от някакъв параметър, който се променя във времето. Новото в тази статия е, че ние ще използваме този параметър за да дадем допълнително описание на състоянието на света.

Ако в един момент ние знаем какво виждаме и какви ходове са коректни в този момент, то ние знаем много за текущото състояние на света, но пак не знаем всичко. Когато обобщим понятието „некоректен ход“ ще получим новото понятие „проверимо състояние“. Ако добавим към вектора на входа стойността на всички проверими състояния, то ще получим един безкрайно мерен вектор, който напълно описва състоянието на света. Тоест, за всеки две различни състояния ще има проверимо състояние, чиято стойност в тези две състояния е различна.

Забележка: Най-близо до понятието „проверимо състояние“ е понятието „Diagnosability“ (виж [1, 2]). Все пак, между тези две понятия има три съществени разлики:

1. Diagnosability отговаря на полу-разрешимо проверимо състояние. Това е така, защото при проверимото състояние ние разполагаме с някакъв експеримент, който, когато може да се проведе, ще ни даде стойността на някакъв параметър. При „Diagnosability“, винаги когато експеримента може да се проведе, тогава параметъра непременно е истина. Когато параметъра е лъжа, експеримента не може да се проведе.

2. При „Diagnosability“ разглеждаме параметър, който само веднъж сменя стойността си (първо failure event не се е случил, после вече се е случил, тоест една единствена промяна от 0 към 1). При проверимото състояние ние разглеждаме параметър, който може многократно да променя стойността си и затова при нас много важен е въпросът, кога точно параметъра е променил стойността си, докато в [1, 2] този въпрос не е съществен.

3. При „Diagnosability“ се тързва от параметъра и се търсят експерименти, които да диагностицират този параметър. При проверимото състояние е обратното. То самото е един конкретен експеримент. По-късно се търси връзка между различните проверими състояния. Например, може да се установи, че две проверими състояния определят един и същи параметър.

Некоректните ходове и проверимите състояния са нещо, което реално съществува, по същия начин както реален е входа, който получаваме на всяка стъпка. Обаче, за разлика от входа, ние не получаваме наготово стойността на проверимите състояния. Например, заключена ли е вратата? За да го проверим трябва да натиснем дръжката и да проверим дали вратата ще се отвори, но това можем да го направим само, ако сме застанали пред вратата. Тоест, проверката изисква допълнително усилие и тя не винаги е възможна (има моменти, в които можем да проверим и моменти, в които не можем). Заключената врата може да се приеме като пример както за некоректен ход, така и като пример за проверимо състояние.

За да опишем некоректните ходове и проверимите състояния ние ще търсим теория, която ни дава това описание. Разбира се, за едно проверимо състояние ние може да имаме много теории, които във времето да се състезават помежду си.

Какво ще представлява теорията? От статистиката сме открили, че при определени обстоятелства дадено проверимо състояние е истина. Определени обстоятелства означава, когато някаква конюнкция е истина. Тази конюнкция може да е свързана само с миналото, а може да е свързана и с миналото и с бъдещето. Например, проверили сме и сме видели, че някаква врата е заключена, но дали това е точно тази врата, която ни интересува? Може да решим, че това е точно тази врата на базата на нещо, което сме видели преди проверката, а може и след проверката, апостериори, да видим нещо, което да ни каже, че това е била точно тази врата.

Друго обобщение на „определени обстоятелства“ ще бъде да разрешим освен зависимостите без памет и зависимости с памет. Зависимост без памет е когато в няколко последователни стъпки се случват някакви събития (тоест това е обикновена конюнкция). Зависимост с памет е когато някакви събития се случват в някакви моменти (които не са последователни) и в периодите между тези моменти определени събития не се случват.

Теорията може да представлява множество от подобни импликации и тези импликации може да бъдат обобщени като краен автомат. Да вземем примера където ние се движим в един замък, в който част от вратите са заключени. Може да имаме правило от вида: „Ако видим бял диван и завием вдясно, то вратата ще е заключена.“ Ако представим картата на замъка като краен автомат, тогава ще

видим, че ако след белия диван сме завили надясно, то ние сме пред вратата X и че тази врата е заключена. Ако знаем крайния автомат ние лесно можем да получим правилата, които той поражда. За съжаление, на нас ни е нужно обратното, от правилата да получим краен автомат, а това е доста по-трудна задача.

Нека сега си представи замък, в който вратите не са постоянно заключени или отключени, а се променят по някакви правила. Тогава нашата теория ще предполага известна устойчивост на проверимите състояния. Например, ако една врата е била заключена в един момент и скоро след това проверим отново, ще предполагаме, че тя отново е заключена, особено ако през това време не са се случвали определени събития (например да сме чули щракане на ключалки).

Следващото надграждане на теорията ще бъде да предположим, че в замъка има някакво същество, което по свое усмотрение отключва и заключва вратите. Тогава ние ще можем да предскажем дали една врата е заключена или отключена предсказвайки поведението на това същество.

След като за едно проверимо състояние ние сме създали една или няколко теории, които го предсказват, ние ще използваме тези теории и ще съберем статистика, която ще ни помогне да предскажем бъдещето, както и да създадем нови теории за други проверими състояния. Например в нашия случай, ако сме забелязали, че зад заключената врата стои принцеса, а зад отключената има тигър, тогава на базата на теорията, която ни казва коя врата е заключена можем да направим теория, която ни казва къде са принцесите.

Статията започва с малко дефиниции, следва един конкретен пример, понятието „некоректен ход“, казваме какво са зависимостите със и без памет, зависещи само от миналото или от минало и бъдеще. Дефинираме понятието проверимо състояние и показваме, че некоректния ход е негов частен случай.

Дефиниции

Разглеждаме една последователност изход, вход, изход, вход ... и целта е да се разбере тази последователност.

Разбира се, тази последователност не е случайна. Може да предположим, че ние играем някаква игра и че това е последователността:

move, observation, move, observation ...

Тогава целта ни е да разберем правилата на играта и каква е текущата позиция на таблото.

Бихме могли да предположим, че се намираме в някакъв свят и тогава последователността би била:

action, view, action, view ...

В този случай целта ни е да разберем правилата на този свят и какво е текущото състояние на света.

Описанието на света се дава от функциите World и View, като е в сила следното:

$$s_{i+1} = \text{World}(s_i, a_{i+1})$$
$$v_i = \text{View}(s_i)$$

Тук действията и наблюденията (a_i и v_i) са вектори от скалари с размерности n и m съответно.

Целта ни е да представим вътрешното състояние (s_i) също като вектор от скалари, в който първите m координати ще са точно v_i . Тоест, функцията View ще бъде проективната функция, която връща първите m координати на s_i .

Всички координати на s_i ще наричаме състояния. Първите m ще ги наричаме видимите състояния. Други координати ще наречем проверими състояния.

Координатите на входа и изхода ще наричаме сигнали. Това са функции на времето, които връщат скалар. Към сигналите на входа и изхода ще добавим и други сигнали, като проверимите състояния. По-точно стойността на проверимото състояние според съответната теория, защото ние няма да знаем точната стойност на тези състояния, а ще ги апроксимираме с някакви теории. За всеки краен автомат ще добавим сигнал, чиято стойност ще бъде равна на състоянието, в което се намира крайният автомат в момента t . Разбира се, ако автоматът е недетерминиран стойността на този сигнал може да не е точна, а приблизителна.

Булевите сигнали ще наричаме събития. Когато булевия сигнал е истина ще казваме, че събитието се случва.

Пример

За да бъдат нещата по-ясни ще разгледаме един конкретен пример. Ще предполагаме, че играем на шах с въображаем противник без да виждаме табло (играем blind). Няма да уточняваме дали въображаемия противник е човек или компютърна програма.

Нашия ход ще е следният 4 мерен вектор:
 X_1, Y_1, X_2, Y_2

Смисъла е, че местим фигура от координати (X_1, Y_1) на координати (X_2, Y_2) .

Това, което ще видим след всеки наш ход е един 5 мерен вектор:
 A_1, B_1, A_2, B_2, R

Първите 4 координати на входа ни показват какъв е отговора на въображаемия противник, а R ни показва каква е непосредствената награда (immediate reward).

Всичките скалари имат стойности от 1 до 8, с изключение на R , чиято стойност е в множеството $\{-1, 0, 1, \text{nothing}\}$. Смисъла на тези стойности е следния: {загуба, реми, победа, партията не е свършила}.

Некоректен ход

Дали да допуснем съществуването на некоректни ходове?

Първото ни предположение е да си изберем такъв свят, че всички ходове в него да са коректни. В примера, който разгледахме, ние не можем да местим офицера като кон и затова е естествено да предположим, че част от нашите ходове са некоректни или невъзможни.

Второто ни предположение е да има некоректни ходове и когато играем такъв ход да предположим, че света ни наказва със загуба. Така много бързо ще се научим да избягваме некоректните ходове, но така се лишаваме от възможността да изучаваме света чрез опипване (проверявайки кой ход е коректен).

Третото ни предположение е, когато играем некоректен ход, състоянието на света да си остане същото, R вместо загуба да ни върне „некоректен ход“, а другите координати на входа да ни върнат „nothing“. Този вариант е по-добър, но по този начин излишно удължаваме историята. (Ще наричаме история редицата $a_1, v_1, \dots, a_{t-1}, v_{t-1}$, където t е текущия момент). Най-малкото, при предположение че състоянието на света си остава същото не е нужно брояча на стъпките да се увеличава.

Четвъртото ни предложение е, когато играем некоректен ход да получим информация, че хода е некоректен, но това да не удължава историята. Новата история ще изглежда така: $u_1, a_1, v_1, \dots, u_t, a_t, v_t$. Тук u_i е множеството от некоректните ходове на i -тата стъпка, които сме пробвали и знаем със сигурност, че са некоректни. По този начин хем историята не се удължава, хем информацията за това, че определени ходове са гарантирано некоректни се записва в историята. Тук предположим, че реда, в който сме пробвали некоректните ходове, няма значение.

Петият вариант, на който ще се спрем в тази статия, ще бъде още по-либерален. В предишния вариант имаме възможността да пробваме дали един ход е некоректен, но, ако се окаже коректен, вече сме го изиграли. Сега ще предположим, че можем да си пробваме различни ходове, колкото си искаме и за всеки да получаваме информация – коректен или некоректен. След това играем ход, за който вече знаем, че е коректен. Знаем, защото вече сме го пробвали. Тук историята е същата като при вариант 5, с тази разлика, че u_i е наредена двойка от две множества. Първото от които е от пробваните некоректни ходове, а второто от пробваните коректни ходове.

Има и шести вариант и той е на всяка стъпка да получаваме пълната информация за това кои ходове са коректни и кои не са. Тоест, получаваме u_i , в което всички ходове участват, все едно че са пробвани. Този вариант не ни харесва, защото това u_i може да е твърде голямо, т.е. това може да е твърде много информация. Освен това предположим, че след известно обучение ще имаме доста точна представа кой ход е коректен и кой некоректен и няма да се налага да правим много опити от този вид.

Зависимости без памет

Ще предполагаме, че текущият момент t е едно доста голямо число и че историята е твърде дълга и не си струва да я помним цялата. Дори и да я запомним би било глупаво на всяка стъпка да я преработваме цялата, за да изберем следващия си ход. Затова ще предполагаме, че вместо да помним цялата история ние сме събрали някаква статистика и ще определяме следващия си ход единствено на базата на тази статистика.

Какво ще представлява тази статистика? Ще разгледаме конюнкциите от литерали. Тук под литерал ще разбираме равенство между сигнал за определено t и конкретна стойност, която този сигнал би могъл да приеме. В конкретния пример, където играехме шах, нека вземем например следната конюнкция:

$$X_1(t)=5, Y_1(t)=2, X_2(t)=5, Y_2(t)=4, A_2(t)=5, B_2(t)=4$$

Това означава, че в момента t сме играли фигура от Е2 на Е4 и въображаемият противник е взел фигурата, която сме играли.

Друг пример:

$$A_2(t-1)=5, B_2(t-1)=4, X_2(t)=5, Y_2(t)=4$$

Това означава, че въображаемият противник е играл на Е4 и ние веднага, още на следващия ход, сме взели фигурата, която той е играл.

Ще предполагаме, че тези конюнкции не са разбъркани, а са подредени:

1. По-старите са преди по-новите (т.е. $t-1$ е преди t).
2. Изхода е преди входа.
3. Координата i на входа (изхода) е преди координата $i+1$.

Ще предполагаме, че тези конюнкции са стабилизирани. Тоест, че времето на най-десния литерал е t . Каква е идеята на стабилизирането? Конюнкцията е едно събитие, което се случва в момента t . Ако положим $t+1$ вместо t ще получим същото събитие, което се случва в момента $t+1$. Тъй като това е едно и също събитие, евентуално изместено във времето, ние ще събираме статистика само за едно от всичките възможни варианти на събитието. Стойността на литералите в момента $t+1$ все още не е известна и затова ще вземем събитието, чиято стойност вече е известна и е станала известна точно в момента t .

За всяка една от тези конюнкции ние ще броим колко пъти тя се е случила (т.е. била а е истина).

Дефиниция: Ход ще наричаме конюнкция, в която всички литерали са за момента t , няма литерали от входа и всичките координати на изхода участват.

Дефиниция: Обобщен ход ще бъде същото като ход, с тази разлика, че няма да е задължително всичките координати на изхода участват.

Тоест, конюнкцията „ход“ ни описва един определен ход, докато обобщения ход ни описва едно множество от ходове.

Дефиниция: Конюнкция, чийто последен литерал е от изхода, ще наричаме: ход при определени условия. Такава конюнкция може да се представи като $A \& M$, където A са определените условия, а M е обобщен ход. (Тук ще предполагаме, че сме разделили на A и M така че M да е максимално.)

За ходовете при определени условия ще броим не само колко пъти са играни, но и колко пъти са пробвани и колко пъти са показали коректен или некоректен ход, когато са пробвани. Тоест, за всеки ход при определени условия ще пазим три брояча:

1. колко пъти е игран,
2. колко пъти е пробван и е коректен,
3. колко пъти е пробван и е некоректен.

Стойността на брояч 2 ще е по-голяма или равна на стойността на брояч 1.

За всяка друга конюнкция ще имаме само брояч 1 (който помни само колко пъти тази конюнкция е била истина). Тук няма да броим колко пъти е имало коректен ход (защото винаги е имало) и няма да броим колко пъти е имало некоректен ход (защото това е информация, която не си струва усилието да я събирате).

Забележка: Празната конюнкция няма да я смятаме за обобщен ход. Тоест, няма да разглеждаме множеството от всички възможни ходове. Ще пропуснем сигнала, който е истина когато всички възможни ходове са коректни. Ще пропуснем този сигнал, защото усилията, които трябва да положим, за да съберем статистика за него не си струват. Има ли случай, когато всички възможни ходове са некоректни? Да, това може да се случи, но това, ако се случи ще е само веднъж. Такава ситуация ще наречем тупик (*cul-de-sac*) или преждевременна смърт. В тази ситуация историята ще приключи, защото няма да има нито един възможен коректен ход.

Всички конюнкции ще ги организираме в едно дърво, където всяка конюнкция ще е един възел от дървото. По този начин ще спестим памет, но най-вече ще спестим време, защото на всяка стъпка ние няма да обикаляме цялото дърво, а само тази негова част където възлите са верни конюнкции (т.е. верни на тази стъпка).

Когато обикаляме дървото ние ще увеличаваме с единица броячите, които са на възлите, през които преминаваме. Важно е да отбележим, че когато имаме няколко коректни хода, то те се представят като поддърво на нашето дърво. Трябва да обиколим това поддърво и да увеличим с единица всички броячи за проверен коректен ход в него. Тези възли, които имат няколко наследника също трябва да се увеличат само с едно. Ако не внимаваме, може да минем няколко пъти през такъв един връх и няколко пъти да го увеличим с едно, което би било грешка. (Това което казахме за няколко коректни хода, същото важи съответно и за няколко некоректни хода.)

Пример: Ако в текущия момент некоректни са обобщените ходове: $X_1(t)=1$, $Y_1(t)=1$ и $X_1(t)=1$, $Y_1(t)=2$ (тоест, не можем да вземем фигура от A1 и от A2), тогава трябва да увеличим с едно броячите на тези два обобщени хода (брояч 3 – за случая, когато в този обобщен ход е имало некоректен ход). Брояч 3 за обобщения ход $X_1(t)=1$ също трябва да се увеличи с едно, защото веднъж се е случило този обобщения ход да съдържа некоректен ход. Нищо, че в този случай той съдържа не един, а два некоректни хода.

С помощта на това дърво ние ще съберем статистика за това, колко пъти е била истина всяка една от наблюдаваните конюнкции. Разбира се, всичките конюнкции са безбройно много и затова ние ще наблюдаваме само част от тях (които са по-къси и които обхващат по-малко стъпки). Ще предположим, че ако една конюнкция я наблюдаваме, то ние наблюдаваме и всички нейни подмножества.

Само информацията за това колко пъти една конюнкция е била истина не ни е достатъчна. Искаме да знаем още кога тя за последен път е била истина. Това лесно може да се постигне, като добавим още една клетка памет към брояча (към трите брояча), които вече отделихме за тази конюнкция. Всеки път когато увеличаваме брояч 1, ще записваме стойността на брояча на времето в тази клетка памет.

Нека поискаме да знаем не само кога за последно конюнкцията е била истина, а да знаем последните 100 пъти, когато е била истина. Това може да се постигне лесно с жертва на памет, но без жертва на компютърно време. Вместо една допълнителна клетка ще отделим сто клетки. В коя клетка ще запишем стойността на брояча на времето? Ще видим колко е брояч 1 по модул сто и това ще ни даде номера на клетката.

По този начин ние ще пазим едни доста стриктни спомени за последните 100 стъпки, а дори и за повече от 100 стъпки, защото за конюнкциите, които рядко са истина, ще знаем какви са били доста назад в миналото.

Това няма да е достатъчно и ще искаме да имаме още информация за времето преди последните сто успеха на събитието. Тази информация няма да е съвсем точна, а ще е приблизителна. За целта ще изберем 100 важни точки, които ще разделят историята на 101 интервала. Ще броим колко пъти се е случило събитието в всеки един от тези 101 интервала. По-точно ще броим колко пъти се е случило от началото (момента на раждането) до края на всеки един от тези интервали, но разликата между две такива числа ще ни даде колко пъти се е случило събитието между произволни две важни точки.

Стотте важни точки няма да са статични, а ще се променят с течение на времето. Идеята е близо до текущия момент те да са по-нагъсто, а назад в миналото да са по-рядко. Ето един примерен алгоритъм за промяна на важните точки:

Слагаме важна точка на всеки 10 стъпки. Когато свършат 100-те точки започваме да разреждаме поставените важни точки, през една, като започваме от началото до текущия момент. Когато свърши първото разреждане, започваме второ, пак от началото и така нататък.

Не е много умно да слагаме важните точки на равни интервали от време. По-хитро би било, ако определим някакви важни събития и слагаме важните точки когато се случи такова събитие. Например, ако се опитваме да намерим краен автомат, чиито преходи са определени събития, то би било умно да слагаме важните точки когато се случват тези събития (преходите). Това би ни помогнало да съберем статистика за този краен автомат и да го постоим по-лесно.

Как ще организираме броенето? Това отново ще бъде с жертва на памет, но без жертва на компютърно време. За всяка конюнкция ще отделим още 100 клетки памет допълнително за стоте важни точки. Така клетките стават 201 (203) общо и то за всяка конюнкция.

Ще си направим една спомагателна таблица. Тя ще бъде със 100 реда и два стълба. В първата колона ще бъдат времената на стоте важни точки (подредени), а във втората ще бъдат номерата на точките (които са между 0 и 99). Тази таблица ще ни помогне да изчислим колко пъти е била истина една конюнкция между две важни точки. Избираме от таблицата двата момента, които определят интересуваният ни интервал. Взимаме от таблицата номерата на съответните важни точки. Проверяваме кога за последен път е била истина конюнкцията и ако двата момента са преди този момент, взимаме разликата. Ако двата момента са след, то отговора е нула. Ако първия момент е преди, а втория след, то взимаме разликата между брояч 1 и клетката на първия момент.

Тази спомагателна таблица ще се използва и при броенето. Ето как ще изглежда алгоритъма на броенето:

Когато една конюнкция е истина взимаме брояч 1 по модул 100 и това е адреса на клетката, която помни кога за последен път е била истина тази конюнкция. Взимаме това число. В следващата клетка (по модул 100) записваме текущия момент. Тръгваме в спомагателната таблица от долу на горе докато времето на създаване на съответната важна точка е след момента, който си помним (когато за последно е била истина конюнкцията). За всяка такава (нова) важна точка взимаме нейния номер и в съответната клетка записваме стойността на брояч 1. Когато приключим, увеличаваме брояч 1 (increment) и сме готови.

Проверими състояния

Проверимото състояние е резултата от някакъв експеримент. Ето два примера:

Вратата е заключена

Ако натисна дръжката \Rightarrow вратата ще се отвори.

Котлона е горещ

Ако пипна котлона \Rightarrow ще се опаря.

Тук е дадено проверимото състояние и експеримента, който трябва да бъде направен, за да бъде получена стойността на това проверимо състояние. Резултата от експеримента може да бъде Да или Не.

От горните два примера може да останете с впечатлението, че експеримента се състои от някакви действия, които трябва да извършим, но това не е така. Може експеримента да е нещо което се случва без наша намеса. Ето пример:

Покрива е за поправка
Ако завали \Rightarrow ще протече.

Тук проверимото състояние се проверява когато завали, а това е нещо, което не зависи от нашите действия.

Дефиниция: Проверимото състояние ще се състои от две твърдения. Първото ще наречем условие, а второто ще наречем заключение. По-долу ще дадем допълнителни изисквания към условието и заключението на проверимото състояние.

Дефиниция: Ще казваме, че едно проверимо състояние е частен случай на друго, ако имат еднакво заключение и ако условието на второто е частен случай на условието на първото.

Забележка: Можем да си мислим за проверимото състояние като за импликация, макар това да не е съвсем точно, защото неговите възможни стойности са три (истина, лъжа и недефинирано – когато условието е лъжа). Ако си представим тази импликация като дизюнкция, то частен случай е тази дизюнкция, която има повече литерали. Съответно, условията ще са конюнкции и частен случай е тази конюнкция, която има по-малко литерали.

Нека да кажем какво е некоректен ход.

Дефиниция: Некоректният ход е проверимо състояние, чието условие е конюнкция, в която всички литерали са от изхода и са за момента $t+1$ и чието заключение ще е „този ход е некоректен“.

Тоест, некоректният ход е проверимо състояние, чието условие е обобщен ход (за момента $t+1$).

За да довършим дефиницията на проверимо състояние, първо ще кажем какво е непосредствено проверимо състояние. Това е проверимо състояние, което, ако може да се провери, то проверката ще се извърши на следващата стъпка.

Дефиниция: Непосредствено проверимо състояние е едно от тези две неща:

1. Некоректен ход.
2. Проверимо състояние, чието условие е конюнкция, в която всички литерали са за момента $t+1$, а заключението ще е един литерал от входа, също за момента $t+1$.

Тук изникват два въпроса. Защо заключението е само един литерал и защо е литерал от входа? На първия въпрос ще отговорим така: Ако заключението е конюнкция на два литерала, то тогава ние можем да представим това проверимо състояние чрез четири други проверими състояния, в които заключението е само от един литерал. Ако това проверимо състояние има вида $A \Rightarrow B_1 \& B_2$, то това проверимо състояние е истина точно когато са истина проверимите състояния $A \Rightarrow B_1$ и $A \Rightarrow B_2$. Това проверимо състояние е лъжа точно когато са истина проверимите състояния $A \& B_1 \Rightarrow \neg B_2$ и $A \& B_2 \Rightarrow \neg B_1$. (Тук отрицанието при булеви сигнали е един литерал, а при небулеви е дизюнкция от няколко литерала.)

Що се отнася до втория въпрос: Ние искаме да опишем света, а не да опишем нашето поведение. Това, че при определени обстоятелства ние винаги сме играли нещо (или никога не сме играли това нещо) не е описание на света, а описание на нашето поведение. Може никога да не сме го играли, защото това е некоректен ход или може винаги да сме го играли, защото това е единствения коректен ход. Това последното вече е описание на състоянието на света, но тази информация се дава от понятието „некоректен ход“.

След като сме казали какво е непосредствено проверимо състояние ще получим понятието проверимо състояние като добавим нещо към условието. Ще добавим предусловие или постусловие или и двете. Предусловието ще е конюнкция от литерали, които са за моменти преди $t+1$, а постусловието ще е конюнкция от литерали, които ще са за моменти след $t+1$.

Тук част от конюнкцията е свързана с миналото (преди $t+1$) и една част е свързана с бъдещето (след t). Проверимото състояние е предсказание за резултата от провеждането на някакъв експеримент, който тепърва ще се проведе. Затова проверимото състояние трябва изцяло да е свързано с бъдещето. В противен случай получаваме нещо, което е частично известно и то или няма да може да се провери (защото известната част е лъжа) или ще бъде проверимо състояние, но друго проверимо състояние (първото ще е частен случай на второто), защото известната част ще е истина и новото проверимо състояние ще зависи само от тази част, която още е неизвестна.

Затова коригираме дефиницията на проверимо състояние като изместваме цялата конюнкция надясно, така че тя цялата да отиде в бъдещето (добавяме някакво k към времето на всички литерали). Трябва да изместим конюнкцията толкова надясно, че най-левия литерал да е за момента $t+1$.

Ако изместим още по-вдясно (например с една стъпка), ще получи друго проверимо състояние, но то ще е прогноза за още по-далечното бъдеще. Така полученото ново проверимо състояние ще е истина, ако старото проверимо състояние е истина на стъпка $t+1$ каквото и да се случило на стъпка $t+1$. Тоест, новото проверимо състояние ще е по-общ случай на старото взето за стъпка $t+1$. Това е така, защото новото може да се получи от старото взето за стъпка $t+1$ като се добави какво се е случило на стъпка t (тоест да се добави конюнкция описваща напълно изхода на стъпка t). Тоест, съвсем естествено получаваме, че проверимото състояние, което гледа по-напред в бъдещето, е по-общ случай от това, което гледа по-наблизо.

В дефиницията на проверимо състояние можем да предполагаме, че предусловието и постусловието не са конюнкции, а са обобщени конюнкция. Тоест, предусловието и постусловието могат да са с памет и да се отнасят за по дълъг период от време (а не само за няколко стъпки).

Да отбележим, че броя на непосредствено проверимите състояния е краен (ако входните и изходните сигнали са крайни функции). Добавяйки предусловия и постусловия ние получаваме безкрайно много проверими състояния, което е естествено, защото не можем да опишем един безкраен свят с крайно много параметри.

Кога проверяваме?

Искаме да съпоставим на всяко проверимо състояние по един сигнал. Тоест, искаме функция на времето, която да е свързана с резултата от провеждането на експеримента. Изниква въпросът, ако експеримента е проведен в някакъв период от време, то кога точно сигнала на проверимото състояние да приеме като стойност резултата от експеримента.

Ако провеждането на експеримента беше ограничено само в една единствена стъпка, то би било лесно, но той може да продължи няколко стъпки. Ако условието на проверимото състояние е обикновена конюнкция, то ние знаем за колко стъпки се провежда този експеримент, но, ако е обобщена конюнкция, дори и това не знаем.

Ще разгледаме няколко варианта:

1. Нека сигнала да получава стойност в момента, в който експеримента е приключил. Тоест, в моментите, в които експеримента приключва сигнала да получава стойност и тя да е равна на резултата от експеримента, а в останалите моменти сигнала да няма стойност. При тази дефиниция стойността на сигнала зависи изцяло от миналото (историята) и въобще не зависи от бъдещето (текущото състояние).

2. Нека стойността се получава в момента, когато се проверява заключението на проверимото състояние. В този момент стойността вече е ясна, но още не е проверено постусловието и то може да бъде истина или лъжа в зависимост от това по кой път ще тръгнем от този момент нататък (какви са бъдещите ни ходове). Ще приемем, че сигнала ще има стойност в този момент, ако постусловието може да бъде изпълнено (при предположение, че играем така, че да го изпълним). При тази дефиниция стойността на сигнала зависи както от миналото (историята) и така и от бъдещето (текущото състояние).

3. Нека изберем момента непосредствено преди започването на експеримента. В този момент ние още не знаем какъв ще бъде резултата от експеримента и дали той въобще ще може да се проведе. Стойността на сигнала ще бъде стойността на прогнозата за резултата от провеждането на експеримента. Това ще е дефиницията,

която ще изберем. Хубавото е, че тази дефиниция зависи само от бъдещето (от текущото състояние).

4. Нека изберем един момент преди започването на експеримента, ама не непосредствено преди, а по-отдалеч. Тогава стойността на сигнала пак трябва да бъде прогноза, но това ще е по-далечна прогноза. Възможните развития гледано по-отдалеч са повече. Например, ако гледано отдалеч се вижда, че експеримента не може да се проведе, то и гледано отблизо пак няма да може, но не и обратното. Тоест, по този начин бихме получили друг, различен сигнал, който отговаря на друго по-общо проверяемо състояние.

Проверимото състояние е нещо реално

Проверимото състояние е нещо съвсем реално, както реален е входа, който получаваме на всяка стъпка. Стойността на това състояние може да разглеждаме като функция, която зависи единствено от текущото състояние на света. Тази функция взима една от четири възможни стойности:



Получаваме стойността „Не може да се провери“, когато експеримента е невъзможен. Кога се случва това? Нека да разгледаме всички възможни развития (всевъзможните ходове, които бихме могли да играем тръгвайки от текущия момент t). Ако при никое от тези развития експеримента не се случва, то тогава за момента t имаме: „Не може да се провери“. Експеримента трябва да се случва веднага, т.е. началото му трябва да е в момента t , а не по-късно. Да не се случва експеримента означава да е лъжа предусловието или постусловието или условието на съответното непосредствено проверяемо състояние.

Съответно, ДА ще бъде, ако понякога може да се повери и винаги когато може да се провери е ДА, аналогично ще бъде за НЕ, а „И да, и не“ ще бъде в останалите случаи.

Кога стойността е „И да, и не“? Това е възможно при хлабав експеримент. Също така това може да се случи при недетерминирана функция World.

Какво значи хлабав експеримент? Това е когато в условието на проверимото състояние са определени само част от сигналите на изхода. Когато са определени всички сигнали на изхода, тогава ще знаем кой точно е хода, който сме играли, и тогава ще наричаме експеримента стегнат.

Пример:

Ако пипна котлона \Rightarrow ще се опаря.

Този експеримент е хлабав, защото не сме казали дали ще пипнем с ръкавица или без ръкавица.

Ако пипна котлона без ръкавица \Rightarrow ще се опаря.

Този експеримент е по-стегнат, защото хода е по-подробно описан.

Забележка: Тук ако предусловието и постусловието са елементарни конюнкции всевъзможните развития ще са крайно много, защото тогава те ще зависят само от няколко стъпки преди и след момента на проверката. Ако те са обобщени конюнкции, то тогава всевъзможните развития може да са безкрайно много, но пак проверимо състояние ще е една добре дефинирана функция, макар че тази функция в този случай може да не е изчислима. Имаме предвид дали можем да я изчислим, ако разполагаме с функцията *World* като подпрограма. Разбира се, ние не разполагаме с функцията *World* и затова тази забележка е чисто теоретична.

Теории

Проверимото състояние не го получаваме наготово. Както казахме това е нещо съвсем реално, но ние не можем директно да го измерим, а трябва да го апроксимираме с теория.

Теорията ще бъде функция, която ще взема като аргумент историята и ще връща сигнал. Този сигнал ще има три възможни стойности: {Да, Не, Не знам}.

Целта на теорията ще бъде да опише сигнала на проверимото състояние. Ще казваме, че теорията е коректна, ако винаги когато тя казва Да, сигнала на проверимото състояние има една от двете стойности: {Да, Не може да се провери}. (Същото за „Не“.)

Ще казваме, че теорията е пълна, ако винаги когато сигнала на проверимото състояние има стойност „Да“, то за този момент теорията казва „Да“. (Същото за „Не“.)

Ще казваме, че теорията е супер пълна, ако е пълна и ако дава прогноза за всички моменти, когато сигнала е „Не може да се провери“. Тоест, за всеки такъв момент теорията да дава „Да“ или „Не“.

Теориите, които ние ще използваме нито ще са коректни, нито ще са пълни, но ние ще се стремим да ги направим максимално коректни и максимално пълни.

Всяка теория за определен момент ни дава някакво предсказание на стойността на проверимото състояние и това предсказание е с някакъв коефициент на достоверност. Тоест, за всеки момент теорията ще ни върне две стойности: предсказание и коефициент на достоверност на това предсказание.

Аксиоми

Дефиниция: Позитивна аксиома ще наричаме проверимо състояние, което винаги когато е можело да се провери е било истина и което е проверявано достатъчен брой пъти.

По аналогичен начин ще дефинираме и негативна аксиома.

Какво значи „винаги когато е можело да се провери“. Това значи, че в историята, която сме натрупали, това е така. Няма гаранция, че това ще е така при всички възможни истории, както и няма гаранция за всички възможни продължения на текущата история. Все пак, ако аксиомата е проверявана достатъчен брой пъти, то с висок коефициент на достоверност можем да предположим, че това ще е така винаги.

Какво значи достатъчен брой пъти? Нека това да е 10 пъти. Ако нещо се е случило само веднъж, то това може да е станало случайно, ако се е случило 10 пъти, то това увеличава коефициента на достоверност, но, разбира се, няма гаранция, че на единадесетия път няма да се случи обратното.

Вместо да броим до 10, по-добре е да искаме да имаме толкова експерименти, че очакването да **не** се случи заключението на условието да бъде поне 10 пъти. Ако заключението има вероятност близка до нула, то тогава да проверим 10 пъти ще е достатъчно. Ако заключението има вероятност около 50%, то тогава ще трябва да проверим поне 20 пъти и т.н.

Заклучението е един конкретен литерал и неговата вероятност може да се вземе от статистиката, като се види колко пъти този литерал е бил истина и се раздели на брояча на стъпките. По-добре е да не се взима вероятността на заключението, а вероятността на малкото проверимо състояние (това което тази аксиома ще опише). Тази стойност пак ще се вземе от статистиката. Колко пъти това малкото проверимо състояние е проверявано и колко пъти е било истина, когато е проверявано.

Теории от аксиоми

За какво ще използваме аксиомите? Ако едно проверимо състояние е позитивна аксиома, то няма какво да го описваме. Неговата теория е тази, която винаги казва „Да“. Интересното е, че с помощта на аксиомите ще описваме по-малките проверими състояния (тези, на които съответната аксиома е частен случай).

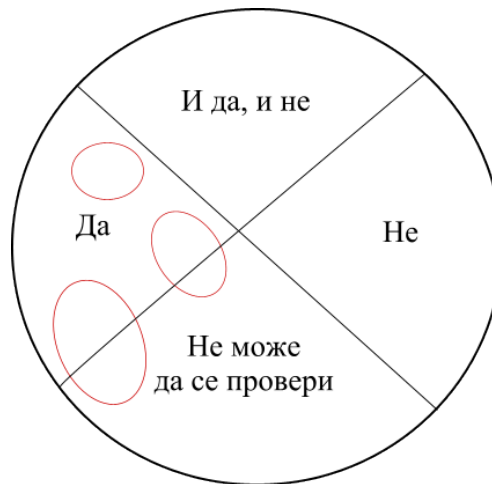
Нека вземем едно проверимо състояние. От статистиката ще намерим аксиоми, които да са частен случай на това проверимо състояние и ще използваме тези аксиоми, за да опишем това проверимо състояние.

Дефиниция: Предпоставка на една аксиома ще наричаме разликата между аксиомата и малкото проверимо състояние, което тя описва. (Тоест, предпоставката зависи от това какво описваме с тази аксиома.)

От всяка позитивна аксиома ще направим теория, която описва малкото проверимо състояние и казва, че то е истина винаги при определени обстоятелства. Определените обстоятелства са, когато е истина предпоставката на аксиомата.

Нека искаме да предскажем стойността на едно проверимо състояние за момента t на базата на информацията, която имаме в момента t . Ще вземем множеството на всички истории с дължина t . Всяка една от тези истории ни води до едно определено вътрешно състояние (може няколко истории да водят до едно и също вътрешно състояние). Разделяме това множество от истории на 4 както разделихме множеството на вътрешните състояния на 4 (фигура 1), като поставяме историята в съответната четвъртина на базата на това до кое вътрешно състояние води тя.

Взели сме три аксиоми, чиито предпоставки са преди момента t (тоест, ако шифтуваме аксиомата наляво, така че малката аксиома да започва от момента t , то цялата предпоставка трябва да е преди t). За всяка от тези три аксиоми рисуваме червен кръг, който огражда историите, при които предпоставката на съответната аксиома е истина.



Забележка: Едно вътрешно състояние може да съответства на две истории и едната да е в червеното кръгче, а другата да е извън него, но и двете непременно ще са в една и съща четвъртинка.

Ще предполагаме, че аксиомите които сме намерили чрез статистиката са коректни. Тоест, предполагаме че техните червени кръгчета са изцяло в четвъртинките „Да“ и „Не може да се провери“. Разбира се, това предположение

може да не е вярно и червеното кръгче на някоя от тези аксиоми да съдържа състояние, което би могло да даде резултат „Не“.

Червеното кръгче може изцяло да е в четвъртинката на „Да“, а може да е там частично. Може ли червеното кръгче да е изцяло в четвъртинката на „Не може да се провери“? Ние тази аксиома в нашата история сме я проверявали достатъчен много пъти. Значи има поне едно вътрешно състояние, което би могло да даде резултат „Да“. Проблемът е, че макар това състояние да е достижимо, няма гаранция, че то може да се достигне с история, която да е с дължина t . По тази причина променяме описанието на фигура 2 и там вече ще участват всички истории, а не само тези, които са с дължина t .

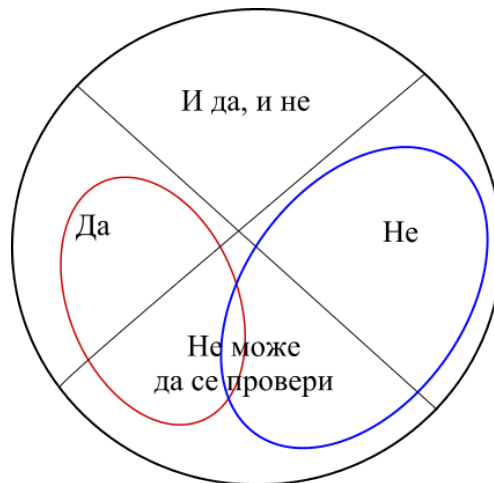
Бихме искали всички истории да завършват в момента t , което би било трудно, ако те са с различна дължина. Тук номерацията на стъпките в историята е в голяма степен условна и затова ние може вместо да ги подравняваме вляво, да ги подравним вдясно. Тоест, вместо номерацията винаги да започва от 1 да я направим винаги да завършва в t . Така по-късите истории ще започнем да номерираме от някое по-голямо число, а по дългите, ще започнем да номерираме от някое отрицателно число. (Нас ни интересува как завършва историята, а не как започва и затова спокойно можем да я номерираме отзад напред.)

След като направихме тази промяна, то във фигура 2 участват вече всички истории и всички достижими състояния (някое може да участва повече от веднъж, защото повече от една история може да види до него). Сега вече можем да кажем, че червеното кръгче не може изцяло да е в четвъртинката на „Не може да се провери“.

Това беше за момента t , на базата на информацията, която имаме в момента t . Ще искаме да можем да предскажем проверимото състояние както в по-ранни, така и в по-късни моменти. Много често късното предсказание е след дъжд качулка, но все пак то ще е полезно при събирането на статистика. Освен това, понякога късното предсказание може да се окаже полезно и да ни даде важна информация. (Става дума за проверимо състояние. Това, че момента е отминал, не значи, че вече знаем каква е била стойността му в този момент. Например, разбираме че на обяд котлона е бил горещ, което означава, че децата са нахранени. Това е полезна информация, получена от стойността на едно проверимо състояние в един минал момент.)

Сега ще направим фигура 2 така, че тя да изобразява аксиомите, които предсказват стойността на проверимото състояние за момента t , на базата на информацията, която имаме в момента $t+k$. Бяхме взели всички истории, които завършват в момента t и на всяка такава история бяхме съпоставили състоянието до което тя ни довежда в момента t . Сега искаме да вземем всички истории, които завършват в момента $t+k$ и на всяка такава история съпоставихме състоянието до което тя ни е довела в момента t . Какво правим когато k е отрицателно? Тогава тази история не стига до момента t . За целта ще вземем всички истории, които завършват в момента $\max(t, t+k)$. Когато k е отрицателно последните няколко стъпки на историята няма да се използват за определянето на това дали предпоставката на аксиомата е валидна, защото тези няколко стъпки ще са още неизвестни.

Получихме една проста картинка с много сложно описание на това, което е изобразено на картинката. Въпреки сложността на това описание, алгоритъма, който следва от тази картинка е съвсем прост. Взимаме всички позитивни аксиоми, които описват определено проверяемо състояние. Правим обединение на червените кръгчета на всички тези аксиоми. Получаваме един червен кръг, които ни описва историите, за които нашата теория казва, че при тях проверяемостта е истина в момента t . По аналогичен начин постъпваме и с негативните аксиоми и получаваме един син кръг. Фигура 3 ни показва как изглежда нашата теория:



Забележете, че червения и синия кръг могат да се пресичат. За случаите, когато теорията ни казва едновременно и Да и Не, тогава ще приемем, че ни каква „Не знам“.

Какъв е алгоритъма? На всяка стъпка проверяваме кои аксиоми излизат (т.е. предпоставката им вече е известна и е истина). Ако до момента теорията ни е казвала „Не знам, защото нямам информация“, то след излизането на аксиомата теорията ни вече знае и вече предсказва нещо. Ако теорията ни вече предсказва „Да“, то новата аксиома, ако е позитивна само ще потвърди това предсказание, а ако е негативна предсказанието не стане: „Не знам, защото имам твърде много информация“.

Да отбележим, че когато аксиомите са обикновени конюнкции, тогава една аксиома ни дава предсказание за един момент, но когато са обобщени конюнкции, тогава една аксиома ни дава предсказание за цял интервал от време.

Заклучение

В областта на AI първата ни задача е да кажем какво търсим, а втората задача е да намерим това което търсим. Тази статия изцяло е посветена на втората задача.

Статиите [3-5], които са посветени на първата задача, ни казват че търсим достатъчно интелигентна програма, която доказва своя интелект като показва, че може да разбере някакъв непознат свят и да се справи достатъчно добре в него.

Ключовото в тази статия е разбирането на света и по-точно разбирането на състоянието на света. Ние твърдим, че ако разберем състоянието на света, то ние сме разбрали и самия свят. Формално погледнато да разберем състоянието на света е все едно да сведем задачата за Reinforcement learning with partial observability към задачата за Reinforcement learning with full observability. Не случайно почти всички статии в областта на Reinforcement learning се занимават със случая на full observability. Това е така, защото това е лесния случай. Единствения проблем при този случай е преодоляването на комбинаторната експлозия. Разбира се, комбинаторната експлозия сама по себе си е достатъчно голям проблем, защото получаваме алгоритми, които биха работили, ако разполагаме с безкрайно бърз компютър и безкрайно дълго време за обучение (дълго в смисъл на брой стъпки). Подобни алгоритми са напълно безполезни за практиката и тяхната стойност е единствено теоретична.

В тази статия представихме състоянието на света като безкрайно мерен вектор от стойностите на всички проверими състояния. Това изглежда достатъчно, за да разберем състоянието на света, но на нас ни е нужно крайно описание и бихме искали това описание да е възможно най-просто. За целта бяха направени още три стъпки. Беше въведен модел на света под формата на краен автомат. Всеки един такъв автомат описва безкрайно много проверими състояния и това е едно много просто описание, с което лесно се работи.

Втората стъпка беше предположението, че проверимите състояния са инертни и се променят само при определени събития. Тоест, ако между две проверки никое от тези събития не се е случило можем да приемем, че стойността на проверимото състояние не се е променила.

Третата стъпка беше въвеждането на агенти, които при определени условия могат да променят стойностите на проверимите състояния. Тази стъпка е особено важна, защото агента скрива в себе си голяма част от сложността на света и по-този начин света става много по-прост и по-разбираем. Самият агент няма да го описваме като система от формални правила, на които той се подчинява. Вместо това ние ще го апроксимираме с предположения от типа на това, че той ни е съюзник и се опитва да ни помогне или че е противник и се опитва да ни попречи.

Без тези три стъпки разбирането на един по-сложен свят би било напълно невъзможно. Формално погледнато за разбирането на състоянието на света са достатъчни само проверимите състояния, чийто предпоставки са прости конюнкции зависещи само от последните няколко стъпки на миналото, но без направените допълнителни обобщения бихме се сблъскали с огромна комбинаторна експлозия.

Acknowledgements

Искам да благодаря на моя колега Dimitar Guelev, който ме запозна с понятието „Diagnosability“ и съответната по въпроса литература.

References

- [1] Meera Sampath, Raja Sengupta, Stephane Lafortune, Kasim Sinnamohideen and Demosthenis Teneketzis. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, sep 1995.
- [2] Xiaowei Huang. Diagnosability in Concurrent Probabilistic Systems. *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.
- [3] Dobrev D. AI – What is this. *PC Magazine – Bulgaria*, November'2000, pp.12-13. <http://www.dobrev.com/AI/definition.html>
- [4] Dobrev D. Formal Definition of Artificial Intelligence. *International Journal "Information Theories & Applications"*, vol.12, Number 3, 2005, pp.277-285. <http://www.dobrev.com/AI/>
- [5] Dobrev D. Comparison between the two definitions of AI. *arXiv:1302.0216*, January, 2013. <http://www.dobrev.com/AI/>