Działania i agenci

Wstęp

Klasyczne podejście do działań oparte na sztucznej inteligencji (AI) zwykle skupiało się na pojedynczych, izolowanych systemach oprogramowania, które działały w stosunkowo nieelastyczny sposób, automatycznie zgodnie z wcześniej ustalonymi regułami. Jednak nowe technologie i aplikacje stworzyły zapotrzebowanie na sztuczne podmioty, które są bardziej autonomiczne, elastyczne i adaptacyjne oraz które działają jako podmioty społeczne w systemach wieloagentowych. W tym rozdziale przedstawiono wprowadzenie i przegląd tej wyłaniającej się sztucznej inteligencji skoncentrowanej na agentach oraz podkreślono znaczenie rozwijania teorii działania, uczenia się i negocjacji w scenariuszach wieloagentowych, takich jak Internet.

Akcja w AI

Historycznie rzecz biorąc, „hipoteza dotycząca systemu symboli fizycznych" w sztucznej inteligencji (Newell i Simon 1976) była osadzona w tak zwanych systemach deliberatywnych. Systemy takie charakteryzują się tym, że zawierają symboliczne modele świata i decyzje dotyczące tego, jakie działania należy wykonać, podejmowane są poprzez manipulację tymi symbolami. Aby system AI zaczął „działać", wystarczy przedstawić mu logiczną reprezentację teorii działania (w jaki sposób systemy podejmują decyzje i zgodnie z nimi postępować) i zmusić go do wykonania małego dowodu twierdzeń. To podejście do działania najlepiej ilustruje problem planowania, w którym systemy wykorzystują manipulację symboliczną, aby ustalić, które działania należy wykonać, aby osiągnąć swoje cele, to znaczy, jak się efektywnie zachować. Zazwyczaj system otrzyma opis stanu świata, w którym się znajduje (stan początkowy) i pożądanego stanu świata (stan końcowy lub cel). System otrzyma także zestaw akcji, którym będzie towarzyszyć lista warunków wykonania akcji oraz lista efektów wynikających z wykonania akcji – które predykaty są usuwane, a które dodawane do opisu akcji. świat. Na przykład wyobraź sobie, że świat składa się z dwóch bloków i stołu, a stan początkowy świata to „blok B na stole, blok A na bloku B, nic na bloku A" lub formalnie {OnTable(B), Włącz(A, B), Wyczyść(A)}; wyobraź sobie również, że celem jest „blok B na stole i blok A na stole", czyli {OnTable(A),OnTable(B)} i że system jest w stanie wykonać dwie akcje: UnStack(x, y) i PutDown(x). Działaniom tym towarzyszy poniższa lista warunków wstępnych i skutków. Dla

Usuń stos (x, y):

Pre {Wł.(x, y), Wyczyść(x)}

Del {Wł.(x, y)}

Dodaj {Trzyma (x), Wyczyść (y)}.

I dla PutDown(x):

Przed {Trzymanie (x)}

Del {Trzyma (x)}

Dodaj {Na stole (x)}.

Jasne, że w tym przykładzie plan składający się z sekwencji działań {UnStack(A, B), PutDown(A)} doprowadzi świat ze stanu początkowego do celu. Na każdym kroku system realizujący algorytm planowania (planista) stara się dopasować warunki poszczególnych działań do opisu świata. Na przykład planista może rozpocząć od próby PutDown(A), ale zakończy się to niepowodzeniem, ponieważ warunek wstępny tej akcji (Holding(A)) nie ma zastosowania. Z drugiej strony warunki

wstępne akcji UnStack(A, B) są spełnione (A jest ułożone na B i jest jasne), więc ta akcja może zostać wykonana. W wyniku wykonania tej akcji do opisu świata dodawany jest Holding(x), warunek wstępny PutDown(A). Po wykonaniu tej drugiej akcji po kolei stan świata przyjmuje postać {OnTable(A), Clear(B), Clear(A), OnTable(B)}, który spełnia cel {OnTable(A), OnTable(B)}. Niestety, biorąc pod uwagę złożoność obliczeniową dowodzenia twierdzeń nawet w bardzo prostych logikach, to podejście do projektowania i wdrażania systemów racjonalnych nie zostało szeroko zastosowane w rzeczywistych scenariuszach. Udowodniono, że nawet wyrafinowane techniki planowania ostatecznie okażą się bezużyteczne w jakimkolwiek systemie ograniczonym czasowo. Jak pokazuje niezwykle prosty przykład powyżej, przeszukiwanie wszystkich możliwych kombinacji w celu wydedukowania celów (twierdzeń) ze zbioru warunków początkowych (przesłanek) zajmuje zbyt dużo czasu. Wyniki te wywarły głęboki wpływ na sztuczną inteligencję, powodując, że niektórzy badacze kwestionowali symboliczny paradygmat sztucznej inteligencji i prowadząc do alternatywnych podejść, w szczególności w architekturach reaktywnych.

System reaktywny to taki, który nie korzysta z symbolicznego modelu świata ani symbolicznego rozumowania przy podejmowaniu decyzji, co dalej. Architektury reaktywne są modelowane jako czarne skrzynki: stosują się do reguł „jeśli-to", które bezpośrednio odwzorowują dane wejściowe na działania. Bez modelu świata i zadania takie systemy są poznawczo elementarne; zachowują się (re) bardziej jak gąsienice niż istoty ludzkie. Być może paradygmatycznym przykładem tego typu systemu jest architektura subsumcyjna, która ustanawia hierarchię konkurencyjnych zachowań, w której niższe warstwy mają pierwszeństwo przed wyższymi (Brooks 1986). Wyobraźmy sobie na przykład reaktywnego robota, który pobiera próbki z, powiedzmy, powierzchni Marsa. Załóżmy, że robot ma następujące zasady (sytuacja → działanie):

1 Jeśli wykryjesz przeszkodę, zmień kierunek.

2 Jeśli przenosisz próbki i u podstawy, upuść próbki.

3 Jeśli przewozisz próbki, a nie w bazie, udaj się do bazy.

4 Jeśli wykryjesz próbkę, pobierz próbkę.

5 Jeśli to prawda, poruszaj się losowo.

Takie zasady tworzą hierarchię, która gwarantuje, że robot obróci się, jeśli napotka przeszkodę; jeśli znajduje się u podstawy i przenosi próbki, zrzuci je, pod warunkiem że nie istnieje bezpośrednie niebezpieczeństwo rozbicia się i tak dalej. Największe zachowanie – losowe przejście – zostanie wykonane tylko wtedy, gdy agent nie ma nic pilniejszego do zrobienia: zakłada się, że jego warunek wstępny „Jeśli jest prawdziwy" zawsze działa. Jest to sposób na zapewnienie, że jeśli zasady (1)–(4) nie mają zastosowania, robot nadal będzie coś robił. Powstałe systemy są, mówiąc obliczeniowo, niezwykle proste, a mimo to mogą wykonywać złożone zadania. Ponadto systemy reaktywne są umiejscowione w rzeczywistych domenach i mogą wykazywać elastyczne zachowanie. W rzeczywistości działania nie są planowane z wyprzedzeniem, lecz są raczej wyłaniającym się rezultatem „zakorzenienia" systemu w konkretnej sytuacji. Jakkolwiek interesujące może być to podejście, stwarza ono kilka problemów. Systemy reaktywne uczą się procedur, ale nie mają wiedzy deklaratywnej; to znaczy, uczą się tylko wartości lub atrybutów, które nie są łatwe do uogólnienia na podobne sytuacje (lub przeniesienia do innych systemów). Poza tym, co być może ważniejsze, właśnie dlatego, że wykazują one właściwości wschodzące, nie ma żadnej opartej na zasadach metodologii budowania takich systemów. Niezależnie od wielu prób łączenia architektur deliberatywnych i reaktywnych w systemach hybrydowych , wydaje się, że ostatecznie pozostaje wybór pomiędzy teoretycznie rozsądnymi, ale niepraktycznymi systemami deliberatywnymi a wydajnymi, ale luźno

zaprojektowanymi systemami reaktywnymi . Może to odzwierciedlać fakt, że każdy typ systemu sztucznej inteligencji został zaprojektowany w celu rozwiązywania powiązanych, ale różnych problemów: symboliczna sztuczna inteligencja była wynikiem wysiłków na rzecz sformalizowania i zmechanizowania rozumowania, które rozkwitły wraz z rozwojem systemów ekspertowych; mając na uwadze, że systemy reaktywne były często motywowane wysiłkami zmierzającymi do rozwiązania numerycznych, nieliniowych problemów, takich jak te związane z koneksjonizmem i sztucznym życiem. W ciągu ostatnich kilkudziesięciu lat badacze byli świadkami ewolucji nowych technologii, takich jak Internet. Wymagają one osobistych, stale działających systemów, dla których starsze koncepcje działania – wynikające albo z kłopotliwego rozumowania symbolicznego, albo z wiecznie adaptacyjnych odruchów – mogą być niewystarczające. Rzeczywiście wielu badaczy uważa, że w XXI wieku, aby systemy sztucznej inteligencji mogły działać „inteligentnie", muszą być w stanie zachowywać się w autonomiczny, elastyczny sposób w nieprzewidywalnych, dynamicznych, typowo społecznych obszarach. Innymi słowy, uważają, że „nowa" sztuczna inteligencja powinna rozwijać agentów (Alonso 2002). Właściwie można postawić tezę, że obecne trendy w tworzeniu i projektowaniu stron internetowych, a także nowe zastosowania w handlu elektronicznym (np. PayPal) i oprogramowaniu społecznościowym (np. Facebook) zostaną w pełni rozwinięte dopiero wtedy, gdy agent przyjmuje się perspektywę.

Trzy zasady sztucznej inteligencji skoncentrowanej na agentach

W tej sekcji szczegółowo zbadano główne funkcjonalności, jakie systemy oprogramowania posiadałyby w społecznościowej, skoncentrowanej na agentach sztucznej inteligencji, czyli innymi słowy, zasady zachowania „nowej" sztucznej inteligencji.

Zachowanie autonomiczne

Badacze autonomii rozumieją zdolność systemów do podejmowania własnych decyzji i wykonywania zadań w imieniu projektanta. Pomysł przekazania części odpowiedzialności systemowi, aby uniknąć żmudnego zapisywania kodu, jest z pewnością bardzo atrakcyjny. Co więcej, w scenariuszach, w których trudno jest bezpośrednio kontrolować zachowanie naszych systemów, niezbędna jest zdolność do autonomicznego działania. Na przykład misje kosmiczne w coraz większym stopniu zależą od bezzałogowych statków kosmicznych i robotów, które samodzielnie podejmują decyzje: zdolność ta jest najważniejsza, ponieważ koszty (czasowe i finansowe) komunikacji między stacją kosmiczną a takimi systemami mogą być wygórowane. To właśnie ta autonomia definiuje agentów. Tradycyjnie systemy oprogramowania wykonują działania (tzw. metody) automatycznie. Wyobraź sobie, że aplikacja internetowa na Twoim komputerze (użytkownik lub klient) żąda dostępu do zawartości strony internetowej przechowywanej w innym systemie oprogramowania w innym miejscu (serwerze lub hoście). Serwer nie może odmówić dostępu do zawartości strony; musi wykonać metodę „wyślij" za każdym razem, gdy zostanie o to poproszony. Natomiast agenci sami decydują, czy zastosować swoje metody zgodnie ze swoimi przekonaniami, pragnieniami i intencjami. Parafrazując Jenningsa, Sycarę i Wooldridge'a (1998): „to, co tradycyjne systemy oprogramowania robią za darmo, agenci robią dla pieniędzy".

Zachowanie adaptacyjne

Po drugie, agenci muszą być elastyczni. Projektując systemy agentowe nie da się przewidzieć wszystkich potencjalnych sytuacji, z jakimi mogą się spotkać i z wyprzedzeniem określić optymalnie ich zachowanie. Na przykład elementy interakcji w Internecie (agenci, protokoły, języki) nie są a priori znane. Agenci muszą zatem uczyć się od swojego otoczenia i dostosowywać się do niego. Zadanie to jest jeszcze bardziej złożone, gdy natura nie jest jedynym źródłem niepewności, ale agent jest umiejscowiony w systemie wieloagentowym (MAS), który zawiera innych agentów o potencjalnie

odmiennych możliwościach, celach i przekonaniach. Poza tym nowe systemy muszą być powszechne. Agent musi mieć kompetencje do pokazania repertuaru działań na tyle ogólnego, aby zachować swoją autonomię w dynamicznych środowiskach. Z pewnością agenta trudno nazwać inteligentnym, jeśli nie jest w stanie dobrze działać, gdy znajduje się w środowisku innym niż (choć pod pewnymi względami podobnym) do tego, dla którego został pierwotnie zaprojektowany. Rzeczywiście, nie ma potrzeby uczenia się czegokolwiek w statycznych, zamkniętych domenach, w których agenci mają doskonałą wiedzę na temat przejść między stanem a akcją. Z kolei inteligencja i uczenie się są ze sobą ściśle powiązane w dziedzinach, w których autonomiczni agenci muszą podejmować decyzje na podstawie częściowych lub niepewnych informacji; to znaczy w dziedzinach, w których agenci uczą się bez nadzoru i bez luksusu posiadania kompletnego modelu świata. Agenci tacy stają przed tak zwanym problemem uczenia się przez wzmacnianie . W takich scenariuszach agent istnieje w środowisku opisanym przez zbiór możliwych stanów. Za każdym razem, gdy agent wykonuje akcję w danym stanie, otrzymuje liczbową nagrodę, która wskazuje bezpośrednią wartość tego przejścia między stanem a akcją – jak „dobre" jest to przejście. Tworzy to sekwencję stanów, działań i nagród. Zadaniem agenta jest nauczenie się polityki, która maksymalizuje oczekiwaną sumę nagród, zazwyczaj z przyszłymi nagrodami dyskontowanymi wykładniczo ze względu na ich opóźnienie. Innymi słowy, im dalej w przyszłość wybiegają przewidywania, tym mniejsze prawdopodobieństwo, że nagrody się zaliczą; rozsądna zasada, ponieważ bardziej odległe nagrody są mniej prawdopodobne. W przeciwieństwie do uczenia się nadzorowanego, takiego jak rozpoznawanie wzorców lub sieci neuronowe, uczącemu się nie mówi się, jakie działania podjąć, ale zamiast tego musi odkryć, które działania przynoszą największą nagrodę, wykorzystując i badając swoją relację ze środowiskiem. Działania mogą mieć wpływ nie tylko na natychmiastową nagrodę, ale także na następną sytuację, a przez to na wszystkie kolejne nagrody. Te dwie cechy, poszukiwanie metodą prób i błędów oraz opóźniona nagroda, to dwie najważniejsze cechy uczenia się przez wzmacnianie. Metodę tę z powodzeniem zastosowano do szeregu problemów organizacyjnych w robotyce, sterowaniu, badaniach operacyjnych, grach, interakcja z komputerem, ekonomia/finanse, złożona symulacja i marketingu.

Zachowanie społeczne

Agenci muszą także wykazywać się postawą społeczną. W środowisku zamieszkanym przez heterogeniczne podmioty agenci potrzebują umiejętności rozpoznawania swoich przeciwników i tworzenia grup, gdy jest to opłacalne. To nie przypadek, że większość platform agentowych zawiera narzędzia wieloagentowe . Niektórzy autorzy rzeczywiście twierdzą, że inżynierię oprogramowania zorientowaną na agenta należy rozwijać właśnie dlatego, że w tradycyjnych systemach oprogramowania nie ma pojęcia struktury organizacyjnej. Ogólnie rzecz biorąc, projektowanie i wdrażanie systemów wieloagentowych jest atrakcyjną platformą dla konwergencji różnych technologii AI. Taka jest filozofia zawodów takich jak RoboCup (www.robocup.org/), podczas których zespoły agentów piłkarskich muszą prezentować swoje indywidualne i zbiorowe umiejętności w czasie rzeczywistym. Co ważniejsze, systemy wieloagentowe odgrywają kilka ról w technologiach informatycznych i telekomunikacji: zapewniają klientom spersonalizowane, przyjazne dla użytkownika interfejsy; jako oprogramowanie pośredniczące są szeroko stosowane do wdrażania rynków elektronicznych i aukcji elektronicznych. Powody tego szczęśliwego mariażu MAS z nowymi technologiami są różne. Gdy domena obejmuje wiele odrębnych systemów oprogramowania, które są fizycznie lub logicznie rozproszone (pod względem danych, wiedzy specjalistycznej lub zasobów), skuteczne rozwiązanie często może zapewnić podejście wieloagentowe. Podobnie, gdy dziedzina jest duża, wyrafinowana lub nieprzewidywalna, ogólny problem można podzielić na szereg mniejszych i prostszych komponentów, które są łatwiejsze w opracowaniu i utrzymaniu oraz które specjalizują się w rozwiązywaniu problemów składowych. Oznacza to, że w większości rzeczywistych zastosowań (pojedynczy) agenci mogą stać się „zbyt duzi", aby dobrze działać, a strategia „dziel i zwyciężaj", w

której wykwalifikowani agenci pracują równolegle, wydaje się bardziej rozsądna. Przykładami mogą być geograficzne rozmieszczenie kamer w sieci drogowej lub zintegrowane podejście wymagane do rozwiązywania złożonych zadań, na przykład współpraca ekspertów (chirurgów, anestezjologów, pielęgniarek) na sali operacyjnej. Podsumowując, w społeczności sztucznej inteligencji powszechnie przyjmuje się, że „nowa" sztuczna inteligencja będzie musiała projektować i wdrażać systemy wieloagentowe, zdolne do działania i uczenia się w szybki i wydajny sposób. Następne dwie sekcje poświęcone są opisowi podstaw zachowań wieloagentowych i uczenia się wieloagentowego.

Zachowanie wielu agentów

Podejścia do zachowań wieloagentowych różnią się głównie stopniem kontroli, jaką projektant powinien sprawować nad indywidualnymi agentami i środowiskiem społecznym, czyli nad mechanizmami interakcji. W systemach rozproszonego rozwiązywania problemów (DPS) jeden projektant jest w stanie kontrolować (lub nawet jawnie projektować) każdego pojedynczego agenta w domenie – zadanie rozwiązania problemu jest rozdzielone pomiędzy różnych agentów, stąd nazwa. Z drugiej strony w MAS jest wielu projektantów i każdy jest w stanie zaprojektować tylko swojego agenta i nie ma kontroli nad wewnętrznym projektem innych agentów. Projektowanie protokołów interakcji jest również ściśle powiązane z kwestią zachęt agentów. Kiedy agenci są projektowani centralnie, zakłada się, że mają wspólny cel ogólny. Dopóki agenci muszą współistnieć i współpracować w jednym systemie, istnieje pewne pojęcie globalnej użyteczności, które każdy agent stara się zmaksymalizować. Agenci tworzą zespoły, które wspólnie przyczyniają się do osiągnięcia ogólnego celu. Natomiast w MAS każdy agent będzie indywidualnie motywowany do osiągnięcia własnego celu i maksymalizacji własnej użyteczności. W rezultacie nie można przyjmować żadnych założeń dotyczących współpracy agentów. Wręcz przeciwnie, agenci będą współpracować tylko wtedy, gdy będą mogli na tej współpracy skorzystać. Badania w DPS rozważają, w jaki sposób pracę związaną z rozwiązaniem problemu można podzielić pomiędzy kilka węzłów, aby zwiększyć wydajność systemu. Oznacza to, że celem jest, aby niezależne węzły rozwiązały globalny problem poprzez spójną współpracę, przy jednoczesnym zachowaniu niskiego poziomu komunikacji. Badacze MAS również są zaniepokojeni skoordynowaną interakcją, ale musi budować agentów, nie wiedząc, jak zaprojektowano ich przeciwników. Głównym problemem badawczym w MAS jest to, w jaki sposób autonomiczni agenci identyfikują wspólną płaszczyznę współpracy oraz wybierają i realizują spójne działania. W szczególności badacze DPS postrzegają negocjacje jako mechanizm przydzielania zadań pomiędzy agentami i alokacji zasobów za pomocą automatycznego kontraktowania. Ponieważ wszyscy agenci mają wspólny cel i mające na celu wzajemną pomoc (w myśl tzw. życzliwości założenie), nie ma potrzeby motywowania agenta, aby zgodził się na wykonanie zestawu działań. Alternatywnie planowanie wieloagentowe to kolejne podejście DPS, które pozwala uniknąć niespójnych i niespójnych decyzji, planując z wyprzedzeniem dokładnie, w jaki sposób każdy agent będzie działał i współdziałał. Planowanie wieloagentowe zostało sformalizowane poprzez rozszerzenie języków i technik planowania jednoagentowego na opisanie złożonych stanów psychicznych – zwykle poprzez zdefiniowanie planów społecznych w kategoriach wspólnych przekonań i wspólnych intencji (Rao, Georgeff i Sonenberg 1992). Z drugiej strony badacze MAS posiadają autonomicznych agentów, którzy wykorzystują negocjacje do dzielenia się pracą związaną z realizacją wcześniej uzgodnionego planu (dla obopólnej korzyści agentów) lub do rozwiązywania jawnego konfliktu. W systemach MAS agenci zazwyczaj zawierają porozumienia w parach w drodze negocjacji dotyczących sposobu koordynacji, przy czym nie ma globalnej kontroli, spójnej wiedzy ani wspólnych celów i kryteriów sukcesu. Zatem głównym celem tego motywacyjnego mechanizmu kontraktowania jest „przekonanie" agentów do osiągnięcia rozsądnych porozumień i zrobienia czegoś w zamian za coś innego. W tym przypadku badacze sztucznej inteligencji oparli się na badaniach dotyczących targowania się z niepełnymi informacjami opracowanymi w ekonomii i teorii gier.

Negocjacja

Ponieważ negocjacje w MAS są prawdopodobnie najpowszechniejszą techniką koordynacji, warto rozważyć ją szczegółowo . W systemie MAS agenci otrzymują mechanizm negocjacyjny składający się z protokołu i zestawu strategii w ramach zestawu transakcji. Negocjacje definiuje się jako proces, podczas którego w każdym momencie jeden agent proponuje porozumienie, a drugi agent albo akceptuje ofertę, albo nie. Jeżeli oferta zostanie przyjęta, negocjacje kończą się realizacją umowy. W przeciwnym razie drugi agent będzie musiał złożyć kontrofertę lub odrzucić ofertę przeciwnika i porzucić proces. Protokół określa zatem, kiedy i w jaki sposób następuje wymiana ofert (tj. jakie działania agenci wykonają lub jakich wstrzymają się od wykonania i kiedy). Na przykład Oferta (x, y, δ_i, t_1) oznacza, że proces negocjacji rozpocznie się w czasie t_1, a agent x oferuje agentowi y umowę δ_i ze zbioru potencjalnych transakcji, zazwyczaj w formie „Wykonam akcję 1 w zamień na działanie 2" lub {Do(x, a_1), Zrób(y, a_2)}. Następnie w kolejnym kroku negocjacji agent y złoży kontrofertę albo z Akceptacją(y, δ_i, t_2), w którym to przypadku etap negocjacji kończy się wdrożeniem umowy, δ_i; lub z Reject(y, δ_i, t_2), tak że negocjacja kończy się niepowodzeniem. Lub, alternatywnie, agent y może wysłać odpowiedź Oferta(y, x, δ_j, t_2), powiedzmy, z δ_j = {Do(x, a_3), Do(y, a_2)}, „Wolałbym, żebyś wykonaj a_3 zamiast a_1", tak aby negocjacje mogły przejść do kolejnego etapu, w którym obowiązuje ta sama procedura. To, jakie konkretne oferty złożą agenci, zależy od ich strategii negocjacyjnej. Jest to funkcja z historii negocjacji do aktualnej oferty zgodna z protokołem. Określa, jaki ruch powinien wykonać agent, aby zmaksymalizować swoją użyteczność, biorąc pod uwagę protokół, negocjacje prowadzone do tego momentu oraz przekonania i intencje agenta. Takie strategie uwzględniają również niechęć do ryzyka agenta; to znaczy, jak niechętnie akceptuje się umowę z niepewnym wynikiem zamiast innej umowy z pewniejszym, ale prawdopodobnie niższym wynikiem. Zwykle wymaga się, aby strategie znajdowały się w równowadze Nasha: oznacza to, że żaden agent nie powinien mieć motywacji do odstąpienia od uzgodnionych strategii. Po przyjęciu strategii, przy założeniu, że agent x ją zastosuje, agent y nie będzie mógł działać lepiej, stosując inną strategię. Aby to zilustrować, rozważmy tak zwany dylemat więźnia. Policja aresztuje dwóch podejrzanych. Policja nie ma wystarczających dowodów, aby wydać wyrok skazujący, więc po rozdzieleniu obu więźniów odwiedza każdego z nich, aby zaproponować tę samą ofertę. Jeśli jeden złoży zeznania (wady drugiego), a drugi będzie milczał, zdrajca wychodzi na wolność, a cichy wspólnik otrzymuje pełny wyrok dziesięciu lat. Jeśli obaj będą milczeć, obaj więźniowie zostaną skazani na zaledwie sześć miesięcy więzienia za drobne zarzuty. Jeśli jeden drugiego zdradzi, każdy otrzyma pięcioletni wyrok. Każdy więzień musi wybrać, czy zdradzi drugiego, czy zachowa milczenie. Podejrzani nie mogą ze sobą rozmawiać w celu osiągnięcia porozumienia. W tym przypadku równowaga Nasha jest taka, że oba świadczą. Każdy podejrzany wie, że jeśli jeden zdecyduje się milczeć, drugi zrobi lepiej, składając zeznania, naruszając w ten sposób równowagę „zachowaj milczenie". Równowaga Nasha jest szczególnie ważną cechą, ponieważ jest postrzegana jako jedyny trwały wynik racjonalnych negocjacji w przypadku braku porozumień możliwych do wyegzekwowania zewnętrznie. Rozwiązanie to ma jednak poważne wady. Po pierwsze, istnieją sytuacje, w których nie ma równowagi Nasha. Na przykład Matching Pennies to przykład gier, w których zysk jednego gracza jest dokładnie równy stracie drugiego gracza. Po drugie, istnieją sytuacje, w których istnieje kilka czystych równowag Nasha. W uproszczonym przykładzie załóżmy, że na wąskiej drodze spotyka się dwóch kierowców. Obaj muszą zjechać z drogi, aby uniknąć czołowego zderzenia. Jeśli obaj zboczą w tę samą stronę, zdołają się wyprzedzić, ale jeśli wybiorą inną stronę, zderzą się. W tym przypadku istnieją dwie czyste równowagi Nasha: albo obie skręcają w lewo, albo obie skręcają w prawo. W tym przykładzie nie ma znaczenia, którą stronę wybiorą obaj gracze, pod warunkiem, że obaj wybiorą to samo. Ponieważ obie strategie są równie dobre, można po prostu rzucić monetą, aby wybrać pomiędzy dwiema alternatywami. Są jednak inne sytuacje, w których nie byłoby takiego wyboru: W grze Battle of the Sexes obaj gracze wolą angażować się w tę samą czynność niż

przebywać w samotności, ale ich preferencje różnią się co do tego, w którą czynność powinni się zaangażować. Gracz 1 woli że oboje imprezują, podczas gdy gracz 2 woli, aby oboje zostali w domu. W tym przypadku istnieją dwie czyste równowagi Nasha, ale nie osiągnięto porozumienia. Wreszcie, akceptując rozwiązanie Nashequilibrium, obaj agenci mogą stracić bardziej zyskowne umowy. Tak jest w przypadku Dylematu Więźnia: równowaga Nasha w tej grze jest rozwiązaniem nieoptymalnym, co prowadzi obu graczy do dezercji, mimo że indywidualna nagroda każdego gracza byłaby większa, gdyby obaj grali wspólnie i milczeli. Zatem zamiast ograniczeń równowagi Nasha oraz w celu zapobiegania irracjonalnym postawom, zazwyczaj przyjmuje się następujące założenia dotyczące racjonalności społecznej: (1) Szczerość: żaden podmiot nie będzie próbował nakłonić innego podmiotu do wiary w twierdzenie, które zna lub które uważa za fałszywe lub twierdzenie, że chce być fałszywe (np. agenci nie mogą zobowiązać się do wykonania działań, których nie są w stanie wykonać). (2) Uczciwość: Agenci muszą działać zgodnie ze swoimi przekonaniami. (3) Fair play: agenci muszą przestrzegać ustalonych umów. (4) Towarzyskość: W przypadku obojętności agenci muszą zaakceptować oferty innych, a transakcje muszą zawsze być indywidualnie racjonalne.

Argumentacja

Założenia dotyczące racjonalności społecznej wymagane do tego, aby poprzednie podejście zadziałało, nie są intuicyjne, a w każdym razie wielu prawdziwych agentów kalkuluje swoje opcje indywidualnie pod kątem własnego interesu, ignorując negocjacje i uzgodnione zobowiązania. W odpowiedzi wielu członków społeczności MAS przyjęło alternatywne podejście do koordynacji MAS. W szczególności przedstawiono kilka badań dotyczących negocjacji opartych na argumentacji jako potężnej techniki współpracy i rozwiązywania sytuacji konfliktowych. W tego typu negocjacjach agenci otwierają przestrzeń porozumienia, wymieniając nie tylko propozycje i kontrpropozycje, ale także powody je wspierające. Ponadto agenci zobowiązują się do zaakceptowania wyników argumentacji, która podlega ścisłym regułom dotyczącym ważności i akceptowalności argumentów oraz ich uporządkowania w typach argumentacyjnych. Wyobraźmy sobie na przykład następującą sytuację: Agent 1 ma młotek, śrubę, śrubokręt i obraz, który zamierza powiesić za pomocą „planu" {młotek + gwóźdź + obrazek}. Agent 2 natomiast posiada lustro i gwóźdź, ma na celu powieszenie lustra i planuje zrealizować plan {młotek + gwóźdź + lustro}. Wyobraź sobie, że agent 1 wie, że agent 2 ma smykałkę i prosi o to. Oczywiście Agent 2 nie może zgodzić się na taką prośbę, gdyż potrzebuje gwoździa do powieszenia lustra. Stosując protokół negocjacji, odrzucenie agenta 2 zakończy odcinek i żaden z agentów nie osiągnie swoich celów. Jeśli jednak pozwolono im się kłócić, agent 2 może wyjaśnić, dlaczego odrzuca ofertę agenta 1 („Potrzebuję gwoździa, żeby zawiesić lustro"), a dzięki tej informacji agent 1 może przekonać agenta 2, że w rzeczywistości istnieje inny sposób zawieszenia lustra, nowy plan wykorzystujący śrubę i śrubokręt zamiast gwoździa. Jeśli agent 2 nie znajdzie błędu w argumencie agenta 1, jest zmuszony go zaakceptować. Skoro tak się wydaje, agenci zgadzają się na wymianę gwoździa na śrubę i śrubokręt, w wyniku czego obaj osiągają swoje cele. Argumentacja opiera się na założeniu, że agenci będą honorować rozumowanie leżące u podstaw umów – założenie to jest trudne do utrzymania w praktyce. W ramach alternatywnych podejść zbadano, w jaki sposób sprawić, by mechanizmy koordynacji MAS były „wykonalne" poprzez prawa socjalne, instytucje, konwencje, a nawet prawa (Alonso 2004). Na tym kończy się nasz opis głównych zagadnień i technik związanych z zachowaniem wieloagentowym. Jednakże, zachowanie w złożonych, dynamicznych scenariuszach, takich jak MAS, nie jest jednorazowym zadaniem, ale procesem udoskonalania, dzięki któremu agenci dostosowują swoje strategie do siebie nawzajem. Dlatego też zajmowanie się uczeniem wieloagentowym ma ogromne znaczenie podczas badania zachowań wieloagentowych.

Uczenie się wieloagentowe

Badania nad uczeniem maszynowym były w większości niezależne od badań nad agentami i dopiero niedawno poświęcono im uwagę w powiązaniu z agentami i systemami wieloagentowymi. Jest to w pewnym sensie zaskakujące, ponieważ zdolność uczenia się i adaptacji jest prawdopodobnie jedną z najważniejszych cech inteligencji. Jak omówiono powyżej, inteligencja implikuje pewien stopień autonomii, który z kolei wymaga umiejętności uczenia się podejmowania niezależnych decyzji w dynamicznych, nieprzewidywalnych dziedzinach, takich jak te, w których współistnieją agenci. Kluczowe kwestie związane z uczeniem się wieloagentowym dotyczą tego, jaką rodzinę technik należy zastosować i czym w istocie jest uczenie się wieloagentowe. Z jednej strony agenci i systemy wieloagentowe można postrzegać jako kolejną domenę zastosowań systemów uczenia maszynowego, która wprawdzie wiąże się z własnymi wyzwaniami. Badania przyjmujące ten pogląd ograniczają się głównie do zastosowania istniejących algorytmów uczenia się jednego agenta mniej więcej bezpośrednio do MAS, tak że uczenie się wielu agentów jest postrzegane jedynie jako wyłaniająca się właściwość. Chociaż mogłoby to być interesujące z punktu widzenia MAS, nie wydaje się zbyt interesujące w przypadku badań nad uczeniem maszynowym. Niemniej jednak jest to kierunek, w którym podąża większość badań nad uczeniem się MAS. Istniejące algorytmy uczenia się zostały opracowane dla pojedynczych agentów uczących się oddzielnych i niezależnych zadań. Alternatywnie, systemy wieloagentowe stwarzają problem uczenia się rozproszonego, to znaczy wielu agentów uczy się oddzielnie, aby wykonać wspólne zadanie. Gdy proces uczenia się zostanie rozdzielony pomiędzy kilka podmiotów uczących się, takie algorytmy uczenia się wymagają rozległych modyfikacji lub trzeba opracować zupełnie nowe algorytmy. W nauczaniu rozproszonym agenci muszą współpracować i komunikować się, aby skutecznie się uczyć; Zagadnienia te są szeroko badane przez badaczy MAS, ale jak dotąd nie poświęcono im zbyt wiele uwagi w obszarach uczenia się. Jeśli chodzi o techniki uczenia się, metody uczenia się nadzorowanego nie są łatwe do zastosowania w scenariuszach wieloagentowych, ponieważ zazwyczaj zakładają, że agentom można zapewnić odpowiednie zachowanie dla dane sytuacji. Dlatego większość badaczy stosowała metody uczenia się przez wzmacnianie do tego stopnia, że problem uczenia się wieloagentowego można zredefiniować jako problem uczenia się przez wzmacnianie dla systemów wieloagentowych. W szczególności najprostszym sposobem rozszerzenia algorytmów uczenia się pojedynczego agenta na problemy wieloagentowe jest po prostu sprawienie, aby każdy agent uczył się niezależnie. Agenci uczą się „jakby byli sami". Nie chodzi zatem o komunikację czy wyraźną koordynację – współpraca i rywalizacja to nie zadania do rozwiązania, a jedynie właściwości otoczenia. Podobnie agenci nie mają modeli stanów psychicznych innych agentów ani nie próbują budować modeli zachowań innych agentów. Niezależnie od tego, jak proste może być to podejście do uczenia się wieloagentowego, założenie, że agenci mogą uczyć się skutecznych polityk w środowisku MAS niezależnie od działań wybranych przez innych agentów, jest nieprawdopodobne. Intuicyjnie najbardziej atrakcyjną alternatywą jest nauczenie agentów strategii równowagi Nasha. Jednakże, koncepcja równowagi Nasha jest problematyczna, a metody sformułowane przy użyciu takiego podejścia obarczone są mnóstwem trudności technicznych, które sprawiają, że ich zastosowanie jest raczej ograniczone.

Wyzwania

Aplikacje agentowe odniosły znaczny sukces w produkcji, kontroli procesów, systemach telekomunikacyjnych, kontroli ruchu lotniczego, zarządzaniu ruchem i transportem, filtrowaniu i gromadzeniu informacji, handlu elektronicznym, zarządzaniu procesami biznesowymi, rozrywce i opiece medycznej . Niemniej jednak jednym z kluczowych problemów był podział na pracę teoretyczną i praktyczną, które w dużej mierze rozwinęły się różnymi ścieżkami. W rezultacie projektantom brakuje systematycznej metodologii jasnego określania i strukturyzacji swoich aplikacji jako systemów (wielo)agentowych. Większość aplikacji agentowych została zaprojektowana w sposób ad hoc, albo zapożyczając metodologię z bardziej tradycyjnych podejść, albo projektując system w oparciu o intuicję

i (koniecznie ograniczone) doświadczenie. W każdym razie, jeśli agenci i systemy wieloagentowe mają stać się standardem w rozwoju nowych aplikacji internetowych – tak jak uważają ich zwolennicy, powinni – konieczne będą pewne istotne zmiany w metodologiach i technologiach zorientowanych na agenty. Po pierwsze, należałoby zbudować język modelowania agentów, który umożliwiłby określanie, wizualizację, modyfikowanie, konstruowanie i dokumentowanie systemów (wielo)agentowych. Twórcy agentów nadal charakteryzują swoje systemy jako rozszerzenia tradycyjnych systemów, dlatego też Unified Modeling Language (UML) jest de facto standardowym językiem w projektowaniu i specyfikacji agentów i systemów wieloagentowych. Wada ta polega na braku odpowiednich metod i technik weryfikacji systemów agentowych. Po drugie, chociaż niektóre funkcje programowania, takie jak abstrakcja, dziedziczenie i modułowość, ułatwiają zarządzanie coraz bardziej złożonymi systemami, Java i inne języki programowania nie mogą zapewnić bezpośredniego rozwiązania w zakresie implementacji agentów. Jak dotąd programy zorientowane na agenty były używane głównie do testowania pomysłów, a nie do opracowywania jakichkolwiek realistycznych systemów. Po trzecie, konieczne będzie ustanowienie standardów interoperacyjności pomiędzy agentami. Debata nie powinna skupiać się wyłącznie na zaletach i wadach różnych języków i protokołów komunikacji agentów, ale także na ontologiach – to znaczy na tym, jakie typy bytów i pojęć definiują domenę agenta oraz jakie są ich właściwości i relacje. Obecnie ontologie są często określane nieformalnie lub są ukryte w implementacji agenta. Aby zapewnić prawdziwą interoperacyjność, agenci będą potrzebować jawnie zakodowanych i współdzielonych ontologii. Czwartą kwestią jest możliwość ponownego użycia. Jeśli systemy wieloagentowe mają być zrównoważone, konieczne będzie opracowanie technik określania i utrzymywania modeli i oprogramowania wielokrotnego użytku dla MAS, agentów i komponentów agentów. Możliwość ponownego użycia jest również konieczna ze względu na mobilność. Jeśli agenci mają przemieszczać się po sieciach rozległych, takich jak WWW, musi istnieć możliwość ich ciągłego ponownego wykorzystania w różnych scenariuszach. Wreszcie, jeśli ludzie mają się oswoić z pomysłem delegowania zadań agentom, należy uwzględnić kwestie związane z zaufaniem. Należą do nich uwierzytelnianie, prywatność komunikacji i informacje o profilu osobistym użytkownika, audyt, odpowiedzialność i obrona przed złośliwymi lub niekompetentnymi agentami. Podsumowując, chociaż istnieje potrzeba utrzymywania teorii i praktyki w tym samym tempie, sztuczna inteligencja skoncentrowana na agentach zapewniła już dojrzałe i zintegrowane techniki i procedury, które nadają się do wykorzystania. Można twierdzić, że paradygmat agenta służył jako pomost między tradycyjnymi systemami sztucznej inteligencji a aplikacjami, które pojawiły się w ciągu ostatnich kilku dekad. Kiedy przy okazji dwudziestej piątej rocznicy AI Magazine zapytano ekspertów o stan wiedzy na temat sztucznej inteligencji, panowało wspólne przekonanie, że sztuczna inteligencja musi wrócić do budowania inteligentnych systemów o ogólnych kompetencjach. Wydaje się, że agenci i MAS mogą dostarczyć nam koncepcji, metodologii i technik niezbędnych do realizacji pierwotnego celu AI w usługach i aplikacjach oferowanych przez Internet.

Wniosek

Systemy AI muszą podejmować inteligentne decyzje. Ale co najważniejsze, muszą pokazać, że to robią, zachowując się odpowiednio. W tym rozdziale skupiono się na roli agentów w analizie zachowania systemów AI. W końcu tym właśnie zajmują się agenci: działają. Dlatego badanie zachowań i działań w sztucznej inteligencji musi uwzględniać agentów. W rzeczywistości istnieją mocne powody, aby sądzić, że agenci stanowią paradygmat ucieleśniający „nową” sztuczną inteligencję. Mówiąc dokładniej, w erze Internetu i usług sieciowych sztuczna inteligencja zacznie skupiać się na tym, jak zbiory autonomicznych agentów koordynują swoje zachowanie (zachowanie wieloagentowe) i na tym, jak się tego uczą (uczenie się wieloagentowe).