

# Vedenjsko kloniranje z metodami strojnega učenja

V tem prispevku se ukvarjamo z računalniško rekonstrukcijo veščine vodenja sistemov, kot sta žerjav ali letalo. Načrtovanje vodenja takih dinamičnih sistemov po klasičnih postopkih iz teorije vodenja zahteva apriorno znanje o objektu vodenja, najpogosteje v obliki matematičnega modela. Pri tem lahko nastopijo težave, npr. če objekta vodenja ne poznamo dovolj dobro, če bi bil razvoj natančnega matematičnega modela predolgotrjen ali predrag, če je sistem močno nelinearen ali pa se s časom spreminja. Poleg tega je včasih predrago, predolgotrajno ali sploh nemogoče sproti zagotavljati tako natančne numerične vrednosti parametrov, kot to zahtevajo klasične metode avtomatskega vodenja.

Hkrati je znano, da so izurjeni operaterji na podlagi izkušenj in kvalitativnega sklepanja sposobni ročno voditi mnoge sisteme, ki so za klasične metode iz zgoraj navedenih razlogov težki. Ena od zanimivih možnosti je torej posnemanje izurjenega operaterja. Pri tem namesto objekta vodenja modeliramo človekovo reševanje naloge z množico pravil, ki so lahko kategorična ali "mehka". Vodenje sistemov s hitro dinamiko ponavadi terja izurjenost, veščino, ki je operativna in jo je mogoče pokazati oziroma opazovati, zelo težko pa jo je natančno opisati. Marsikateri od svojih reakcij se operater sploh ne zaveda, še manj pa jih zna razložiti. To znanje je težko prenosljivo, kar otežuje poučevanje začetnikov, hkrati pa tudi onemogoča njegovo

neposredno “vgradnjo” v napravo, ki bi bila sposobna večino reproducirati. Zaradi težav, ki jih imajo ljudje pri eksplicitnem izražanju tovrstnega znanja, je pri netrivialnih sistemih pogosto težko priti do ustrezne množice pravil za vodenje sistema.

Eno od možnih rešitev je predlagal Donald Michie in jo poimenoval vedenjsko kloniranje (angl. *behavioural cloning*) (Michie in sod. 1990; Michie 1993). Pri tem postopku izhajamo iz posnetkov operaterjevega izvajanja naloge, ki jim pravimo tudi vedenjske sledi. Iz njih poskušamo s *strojnim učenjem* (angl. *machine learning*) (Bratko 1989) priti do operativnih opisov veščine na simbolični ravni. Rezultat je *vedenjski dvojniki*, zato postopku pravimo tudi generiranje vedenjskih dvojnikov.

Običajna formulacija problema generiranja vedenjskega dvojnika s strojnim učenjem je takšna:

Imamo posnetek operaterjevega izvajanja. Posnetek je zaporedje stanj vodenega sistema in operaterjevih akcij, na katere gledamo kot na primere pravilnih odločitev. Vsak primer je par (*Stanje*, *Akcija*), pri čemer je *Stanje* vektor atributnih vrednosti, *Akcija* pa je odločitev, prirejena temu stanju. Pri tej formulaciji lahko uporabimo tehnike strojnega učenja, ki inducirajo funkcijsko relacijo  $Akcija = f(Stanje)$ . Vrednosti atributov vektorja *Stanje* so lahko zvezne ali diskretne, prav tako tudi vrednosti razreda *Akcija*. Naučena funkcija  $f$  predstavlja vedenjskega dvojnika, ki naj bi bil sposoben “oponašati” posnetega operaterja, torej voditi dinamični sistem.

Možno je upoštevati tudi časovni zamik med stanjem in akcijo. Običajna interpretacija se sklicuje na fiziološke omejitve, ki jih ima človek tako pri prepoznavanju situacije kot pri fizičnem izvajanju kontrolnih akcij. Če je zamik ( $t$ , se učimo  $Akcija(t) = f(Stanje(t - \Delta t))$ ). S poskusi je bilo ugotovljeno, da je primerna izbira vrednosti zamika eden od pomembnih dejavnikov uspeha kloniranja. Vedeti pa moramo, da zamik odseva celo vrsto pojavov, da se spreminja od naloge do naloge, od človeka do človeka in celo pri istem človeku od situacije do situacije, zato poenostavljene interpretacije niso primerne, izbira pa je v odsotnosti trdnejših teoretičnih ali eksperimentalnih izsledkov zaenkrat še rezultat sistematičnega poskušanja.

Za generiranje vedenjskih dvojnikov iz posnetkov vodenja lahko uporabimo obstoječe programe za strojno učenje, ki iz dane množice učnih primerov generirajo splošen opis zakonitosti, vsebovanih v podanih primerih. Naučeni opisi imajo lahko obliko odločitvenih dreves, kot npr. pri učnih programih ID3 (Quinlan 1979), C4.5 (Quinlan 1987) in Asistent (Cestnik in sod. 1987), ali pa klasifikacijskih pravil, kot npr. pri programu CN2 (Clark in Niblett 1989). Pri našem delu smo veliko uporabljali programa RETIS (Karalič 1991) in M5 (Quinlan 1993), ki

gradita regresijska drevesa in sta primerna za obravnavo zveznih atributov in razredov.

Ker problemska domena v veliki meri posega v področje implicitnega, dostikrat nezavednega znanja ljudi, se srečujemo še z dodatnimi vprašanji, kot sta problem interpretacije rezultatov strojnega učenja in problem vzajemne povezave med rezultati strojnega učenja in človekovim razumevanjem njegove lastne veččine. Zato imamo pri generiranju vedenjskih dvojnikov dva cilja:

- generiranje vedenjskega dvojnika, ki bo čimbolj *uspešno izvajal* predpisano nalogo vodenja, in
- generiranje vedenjskega dvojnika, ki bo dovolj *razumljiv*, da bo na simbolični ravni kazal, kaj operater pravzaprav počne.

Generiranje vedenjskih dvojnikov je bilo po naših informacijah zaenkrat preizkušeno v naslednjih problemskih domenah:

- vodenje inverznega nihala (Michie in sod. 1990; Zalar 1994),
- pilotiranje letala (Sammut in sod. 1992; Michie in Camacho 1994),
- razporejanje opravil (Kibira 1993),
- vodenje kontejnerskega žerjava (Urbančič 1994; Urbančič in Bratko 1994).

Skupne ugotovitve omenjenih študij so naslednje:

- V vseh do sedaj preizkušenih problemskih domenah je bilo možno s standardnimi tehnikami strojnega učenja generirati vedenjske dvojnike, ki so uspeli rešiti zastavljeno nalogo.
- V vseh omenjenih domenah je bilo možno generirati tudi vedenjskega dvojnika, ki je izvedel nalogo v določenem smislu bolje od človeka, katerega posnetki so bili vir učnih primerov.
- Ljudje se v načinu vodenja pri enako zastavljeni nalogi razlikujejo, zato vsi avtorji odsvetujejo učenje iz primerov, izvirajočih iz posnetkov različnih operaterjev.

Dejstvo, da so bili v vseh omenjenih problemskih domenah generirani dvojniki, ki so rešili isto nalogo kot operater, in to celo v nekem smislu bolje od njega, je obetavno. Vseeno pa moramo omeniti problema zanesljivosti in razumljivosti. Težave z zanesljivostjo se pojavljajo na dveh ravneh. Kot prvo omenimo veliko odvisnost uspešnosti postopka od parametrov učenja. Omejujoče so tudi težave z robustnostjo samih vedenjskih

dvojnikov. V kompleksnejših domenah, kot npr. pilotiranje in vodenje žerjava, so le-ti zelo neodporni za spremembe parametrov problema. Že malo spremenjeno začetno stanje je lahko za dvojnika hud problem. Eden od razlogov je verjetno v tem, da učni primeri slabo pokrivajo problemski prostor. Drugi, po našem mnenju pomembnejši, pa se navezuje na dejstvo, da dvojniki izvajajo nalogo izrazito reaktivno, ne da bi karkoli "razumeli" o domeni, ciljih in podciljih operaterja, vzročnih povezavah ipd. Prav to je tudi razlog za težave, ki jih lahko imamo, preden doumemo, kaj dvojniki pravzaprav počne. Na problema robustnosti dvojnikov in njihove razumljivosti zato gledamo povezano.

Ena od najpomembnejših do sedaj zapostavljenih nalog pri generiranju vedenjskih dvojnikov je po našem mnenju primerjava dvojnika, generiranega iz posnetka operaterja, in navodil za vodenje, kot jih opiše isti operater. Ta primerjava naj bi pomagala poiskati odgovore na vprašanje o transparentnosti generiranih dvojnikov ("Kaj dvojniki pravzaprav počne?"), njihovi robustnosti in splošnosti ("Ali je možno dvojnika izboljšati z upoštevanjem operaterjevih navodil?"), prav tako pa tudi na vprašanje, ali je možno operaterjeva besedna navodila izpopolniti na podlagi informacije, ki jo vsebuje dvojniki.

Na osnovi primerjave ugotavljamo, da lahko posnetki operaterjevega izvajanja in operaterjeva navodila ob primerni uporabi *obeh* pripomorejo k izpolnitvi zastavljenega cilja, to je vedenjskega dvojnika, ki bo hkrati *operativen* in *razumljiv*. Operaterjeva navodila predstavljajo eksplicitni del njegovega znanja o problemski domeni, ki ga lahko izkoristimo kot predznanje za avtomatsko učenje (izbira atributov ipd.). Hkrati pa sam ta del operaterjevega znanja ni dovolj za avtomatsko vodenje. Avtomatsko generirani vedenjski dvojniki lahko pomagajo zapolniti vrzeli v operaterjevih navodilih in natančneje določiti numerične vrednosti, ki jih operater poda bolj "po občutku". Zato smo predlagali ciklično metodološko shemo za generiranje vedenjskih dvojnikov, pri kateri operaterjevo sicer nepopolno eksplicitno znanje pomaga pri učenju dvojnikov, hkrati pa rezultat strojnega učenja izkoristimo za dopolnitev operaterjevih besednih navodil (Urbančič in Bratko 1994).

Kombinacija refleksov, usmerjenega opazovanja, planiranja, kreativnosti v poskušanju, mehanizmi prehajanja učenja s sub-simbolnega na simbolno raven in obratno, vse to omogoča, da se ljudje izurijo v reševanju zelo kompleksnih nalog vodenja. Strojno učenje nam omogoča, da z generiranjem vedenjskih dvojnikov dobimo vsaj delen vpogled v tako pridobljeno človekovo veščino. Rezultat je po našem mnenju vse, kar se človek iz tega nauči o sebi, in vse, kar prispeva k izpopolnjenemu delovanju človeka ali naprave pri reševanju določene naloge.

## LITERATURA

- BRATKO, I. (1989): *Machine Learning*, v: K. J. Gilhooly (ur.): **Human and Machine Problem Solving**, Plenum Press, New York, str. 265-287.
- CESTNIK, B., KONONENKO, I., BRATKO, I. (1987): Assistant-86: A knowledge-elicitation tool for sophisticated users, v: I. Bratko, N. Lavrač (ur.), **Progress in Machine Learning**, Sigma Press, Wilmslow.
- CLARK, P., NIBLETT, T. (1989): The CN2 induction algorithm, **Machine Learning 3**, 261-283.
- KARALIČ, A. (1991): *Avtomatsko učenje regresijskih dreves iz nepopolnih podatkov*, Magistrsko delo, Fakulteta za elektrotehniko in računalništvo, Univerza v Ljubljani.
- KIBIRA, A. (1993): *Developing an expert controller of a black box simulation of a telephone line using machine induction*, Neobjavljeno poročilo, AI Laboratory, University of New South Wales.
- MICHIE, D. (1993): *Knowledge, learning and machine intelligence*, v: knjigi L. S. Sterling (ur.), **Intelligent Systems**, Plenum Press, New York.
- MICHIE, D., BAIN, M., HAYES-MICHIE, J. (1990): *Cognitive models from subcognitive skills*, v knjigi: M. Grimble, J. McGhee, P. Mowforth (ur.), **Knowledge-Based Systems in Industrial Control**, Stevenage, Peter Peregrinus, str. 71-99.
- MICHIE, D., CAMACHO, R. (1994): *Building symbolic representations of intuitive real-time skills from performance data* v knjigi: K. Furukawa, S. Muggleton (ur.), **Machine Intelligence and Inductive Learning**, Oxford University Press (v tisku), Oxford.
- QUINLAN, R. (1979): *Discovering rules by induction from large collection of examples*, v: D. Michie (ur.), **Expert Systems in the Micro-electronic Age**, Edinburgh University Press.
- QUINLAN, R. (1987): *Simplifying decision trees*, **International Journal of Man-Machine Studies**, vol. 27, št. 3, 221-234.
- QUINLAN, R. (1993): *Combining instance-based and model-based learning*, **Proceedings of the 10th International Conference on Machine Learning**, Morgan Kaufmann, 236-243.
- SAMMUT, C., HURST, S., KEDZIER, D., MICHIE, D. (1992): *Learning to Fly*, v: D. Sleeman, P. Edwards (ur.), **Proceedings of the Ninth International Workshop on Machine Learning**, Morgan Kaufmann, str. 385-393.
- URBANČIČ, T. (1994): *Avtomatizirana sinteza znanja za vodenje sistemov*, Doktorska disertacija, Fakulteta za elektrotehniko in računalništvo, Univerza v Ljubljani.
- URBANČIČ, T., BRATKO, I. (1994): *Reconstructing Human Skill with Machine Learning*, v: A. Cohn (ur.), **Proceedings of the 11th European Conference on Artificial Intelligence**, 498-502. John Wiley & Sons, Ltd.
- ZALAR, A. (1994): *Avtomatsko učenje vodenja dinamičnega sistema*, Diplomaska naloga, Fakulteta za elektrotehniko in računalništvo, Univerza v Ljubljani.