

Konstruktivna indukcija v strojnem učenju

Constructive induction in machine learning

Marko Robnik
Univerza v Ljubljani
Fakulteta za elektrotehniko in računalništvo
Tržaška 25, 61000 Ljubljana
tel.: + 386 61 1768-386
e-mail: rmarko@fer.uni-lj.si

Povzetek

S tehniko konstruktivne indukcije skušamo razširiti dani jezik za opis znanja z novimi konstrukti. To lahko pripomore k uspešnosti učenja ter h kompaktnosti in razumljivosti naučenega, lahko pa tudi pripomore k razumevanju učnega problema.

Podajamo pregled razvoja konstruktivne indukcije na različnih področjih strojnega učenja. Področja odločitvenih pravil, odločitvenih dreves in induktivnega logičnega programiranja opišemo tudi s primeri aplikacij, medtem ko učenje z Bayesovim klasifikatorjem, nevronskimi mrežami in razpoznavanjem vzorcev predstavimo le z nekaj idejami.

ABSTRACT:

Constructive induction expands given description language with new constructs and thereby tries to improve accuracy, compactness and comprehensibility of learned theories. It can also aid humans to better understand the learning problem.

The article shows current state of constructive induction research in various fields of machine learning. Decision rules, decision trees and inductive logic programming are presented also with examples of applications, while Bayesian classifier, neural networks and pattern recognition are described only briefly with some ideas.

Ključne besede: konstruktivna indukcija, učenje iz primerov, strojno učenje, umetna inteligenca, odločitvena pravila, odločitvena drevesa, induktivno logično programiranje, razpoznavanje vzorcev, nevronske mreže, Bayesov klasifikator

Keywords: constructive induction, learning from examples, machine learning, artificial intelligence, decision rules, decision trees, inductive logic programming, pattern recognition, neural networks, Bayesian classifier

1. Uvod

Algoritmi, ki se učijo opisov konceptov iz primerov, v določeni meri predvidevajo razporeditev primerov v problemskem prostoru (*Utgoff, 1986*) in na podlagi predpostavk identificirajo podprostore, ki predstavljajo iskani koncept. Tako je za uspešno učenje koncepta nujno, da se primeri, ki ga predstavljajo, nahajajo v enem ali več območjih problemskega prostora, ki smo jih sposobni opisati z danim opisnim jezikom.

Problem nastopi, ko primeri niso strnjeni v območjih, ki jih je dani opisni jezik sposoben opisati. Takrat je inducirani opis koncepta zapleten, težko razumljiv, nepopoln, preveč prilagojen učnim primerom in zato nepravilen. Vzroka za to sta dva:

- šum v učnih primerih
- neprimerna opisni jezik in formalizem

S problemem neprimernega opisnega jezika se ukvarja konstruktivna indukcija.

Pojem *konstruktivna indukcija* je prvi uporabil Michalski (*1986*). Po njem konstruktivna indukcija v nasprotju s selektivno indukcijo, ki za indukcijo opisa koncepta izbira le med njegovimi že znanimi lastnostmi, generira tudi nove lastnosti.

Tej definiciji lahko oporekamo, saj ni povsem jasno, kaj so nove lastnosti: le-te bodo nujno izražene s starimi, gre torej zgolj za vprašanje opisnega jezika.

Na konstruktivno indukcije lahko gledamo tudi kot na poskus transformacije originalnega problemskega prostora v prostor, kjer so učni primeri razporejeni bolj urejeno. Iščemo primeren jezik za predstavitev danega učnega problema (*Pfahringner, 1994*).

Kakorkoli, konstruktivna indukcija lahko vsekakor bistveno pripomore k uspehu učenja (*Kibler in Langley, 1988*). Naštejmo nekaj prednosti (*Ragavan in sod., 1993; Robnik, 1993; Yang in sod., 1991*):

- z gradnjo vmesnih konceptov (atributov, jezikovnih gradnikov) pripomoremo k večji kompaktnosti in razumljivosti končnega koncepta

- dobimo izraznejši opisni jezik
- dosežemo večjo točnost klasifikacije

Pri tem pa se nam obetajo tudi težave (*Pfahring*, 1994; *Robnik*, 1993):

- mogoča je prevelika prilagoditev učnim primerom, kar pomeni, da so konstrukti pretirano kompleksni in zapleteni in zato težje razumljivi
- večja računrska kompleksnost učenja, ki sledi iz poskusov generiranja novih koristnih konstruktov

Različna področja strojnega učenja se različno lotevajo konstruktivne indukcije in jo tudi različno poimenujejo.

Drugi in tretji razdelek se ukvarjata s konstruktivno indukcijo pri učenju pravil in odločitvenih dreves. Četrty se dotika tvorbe novih predikatov v induktivnem logičnem programiranju, peti pa predstavi nekaj idej iz drugih pristopov k učenju: Bayesovega klasifikatorja, prepoznavanja vzorcev in nevronskih mrež.

2. Odločitvena pravila

Obstaja več načinov uporabe konstruktivne indukcije pri učenju odločitvenih pravil. V grobem lahko ločimo dva pristopa:

Transformacijska indukcija (*Muggleton*, 1990): množico konjunktivnih pravil preoblikujemo s postopno uporabo transformacijskih operatorjev,

Konstrukcija pravil iz atributov (*Pfahring*, 1994; *Michalski*, 1983): konstrukte gradimo z uporabo operatorjev nad osnovnimi atributi ali že izpeljanimi konstrukti.

Poglejmo si podrobneje oba pristopa, ilustrirana s primeri realnih sistemov.

2.1 Transformacijska indukcija

2.1.1 Razvrščanje konceptov v skupine

Eden prvih in najbolj znanih pristopov h konstruktivni indukciji je vejetno razvrščanje konceptov v skupine (conceptual clustering) (*Michalski in Stepp, 1983*). Uporablja statistično tehniko razvrščanja konceptov v skupine in na ta način zgradi hierarhijo koncepta oziroma konceptov.

2.1.2 INDUCE

INDUCE je sistem za učenje odločitvenih pravil (*Dietterich in Michalski, 1983; Michalski, 1983*), ki poleg selektivne uporablja tudi konstruktivno indukcijo. Vsebuje naslednje operatorje za konstrukcijo:

- **Konstruktivna generalizacija:** Obstoječi pravili

$$X \leftarrow A \wedge B$$

$$Y \leftarrow B$$

zamenjamo z

$$X \leftarrow A \wedge Y$$

- **Operatorja preštevanja:** konstruirani atribut pomeni število primerov, ki zadoščajo nekemu pogoju, pri tem so primeri lahko zapisani kot vrednosti atributov ali kot relacije
- **Lastnost verige:** če imamo neko tranzitivno relacijo in tvorijo primeri zaradi tega neko zaporedje (verigo), tvori operator atribut, ki določi posebne elemente v verigi: prvega, zadnjega, srednjega, N-tega, ali pa dolžino verige
- **Odvisnost atributov:** konstrukt opisuje relacije med posameznimi atributi: monotono povezanost, monotono povezanost pri nekem pogoju, približno monotono povezanost, korelacijo, korelacijo pri pogoju, ...

2.1.3 Duce

Duce je del okolja za razvoj ekspertnih sistemov RuleMaster (*Muggleton, 1990*). Vhod v algoritem je množica konjunktivnih pravil, ki jih zgradi sistem v prvi fazi. Duce postopoma uporablja množico operatorjev (transformatorjev), da spremeni podmnožico vhodnih pravil. Išče takšne transformacije, ki čimbolj zmanjšajo število potrebnih simbolov za zapis pravil. Število simbolov, potrebnih za zapis pravila definira kot število konjunktivnih simbolov v telesu pravila plus 1 (za glavo pravila). Za vsak operator lahko napovemo, za koliko se bo z njegovo uporabo zmanjšala dolžina zapisa pravil. Koristne transformacije išče v prostoru vseh mogočih transformacij po načelu najprej najboljši. Ko najde lokalno najboljši transform, ga preveri s prerokom (vpraša uporabnika za potrditev).

Poglejmo si operatorje in njim pripadajoče število $V_{operator}$, ki pove koliko manj simbolov potrebujemo z uporabo operatorja. Pri tem pomeni R' podmnožico množice pravil R , nad katero opravimo transformacijo, in I' skupno podmnožico teles pravil iz R' .

1. **Interkonstrukcija.** Skupino pravil, na primer:

$$\begin{aligned} X &\leftarrow B \wedge C \wedge D \wedge E \\ Y &\leftarrow A \wedge B \wedge D \wedge F \end{aligned}$$

preoblikujemo v

$$\begin{aligned} X &\leftarrow C \wedge E \wedge Z? \\ Y &\leftarrow A \wedge F \wedge Z? \\ Z? &\leftarrow B \wedge D \end{aligned}$$

$Z?$ je najbolj specifična posplošitev pravil za X in Y .

$$V_{Inter} = (|I'| - 1)(|R'| - 1) - 2$$

2. **Intrakonstrukcija.** Gre za distributivni zakon Boolove algebre.

$$\begin{aligned} X &\leftarrow B \wedge C \wedge D \wedge E \\ X &\leftarrow A \wedge B \wedge D \wedge F \end{aligned}$$

preoblikujemo v

$$\begin{aligned} X &\leftarrow B \wedge D \wedge Z? \\ Z? &\leftarrow C \wedge E \\ Z? &\leftarrow A \wedge F \end{aligned}$$

Uporabili bi lahko tudi interkonstrukcijo, vendar je intrakonstrukcija kompaktnejša:

$$V_{Intra} = |I'|(|R'| - 1) - 2$$

3. **Dihotomizacija.** Cepimo pravila, ki delujejo na pozitivnih in negativnih primerih.

$$\begin{aligned} X &\leftarrow A \wedge B \wedge C \wedge D \\ \overline{X} &\leftarrow A \wedge C \wedge J \wedge K \\ \overline{X} &\leftarrow A \wedge B \wedge C \wedge L \end{aligned}$$

zamenjamo z

$$\begin{aligned} X &\leftarrow A \wedge C \wedge Z? \\ \overline{X} &\leftarrow A \wedge C \wedge \overline{Z?} \\ Z? &\leftarrow B \wedge D \\ \overline{Z?} &\leftarrow J \wedge K \\ \overline{Z?} &\leftarrow B \wedge L \end{aligned}$$

Pravila s pozitivno in negativno glavo morajo imeti skupne simbole.

$$V_{Dichot} = |I'|(|R'| - 2) - 4$$

Poleg konstruktivnih Duce uporablja tudi druge operatorje, ki mu pomagajo preoblikovati pravila in pripraviti pogoje za morebitno uporabo konstruktivnih operatorjev.

Duce so uspešno uporabili pri konstrukciji šahovskih končnic in diagnostiki v nevropsihologiji. Zgrajene koncepte so strokovnjaki ocenili za koristne in razumljive. Menili so tudi, da izražajo dejanske odvisnosti v problemski domeni.

2.2 Konstrukcija pravil iz atributov

2.2.1 CiPF

Tudi CiPF (*Pfahring*, 1994; 1994a) uporablja operatorski pristop h konstruktivni indukciji. Operatorje uporablja na že obstoječih atributih in na novo generiranih konstruktih. Za omejitev rasti konstruktov uporablja načelo minimalne dolžine opisa MDL (Minimum Description Length) (*Rissanen*, 1978).

Uporablja naslednje konstruktivne operatorje:

- primerjava atributov istega tipa glede enakosti, neenakosti, relacij večji, manjši (npr.: Ali je razdalja po osi x manjša od razdalje po osi y ?)
- diskretizacija numeričnih atributov v več intervalov z uporabo tehnike Chi-Merge (*Kerber*, 1992) (konstrukt označuje, da lahko vrednost atributa pade v več intervalov)
- združitev vrednosti nominalnih atributov v množice z uporabo tehnike Chi-Merge (*Kerber*, 1992) (konstrukt označuje, da lahko vrednost atributa pade v več množic)
- prešteje, koliko logičnih spremenljivk iz množice ima vrednost 1(0)
- za množico pozitivnih primerov, ki jih pokrije pravilo: izračunaj intervale (podmnožice) atributov, tako da ti natančno pokrijejo dano množico primerov
- 3σ heuristika: za numerične attribute določimo ali pripada vrednost "zdravemu" intervalu $[\mu - 3\sigma, \mu + 3\sigma]$ z odstopanjem 3σ od srednje vrednosti ¹
- kombinacija vrednosti dveh atributov, nam da nov logičen atribut; tako lahko za logične attribute dobimo vse mogoče logične funkcije (npr. konstrukt predstavlja kombinacijo: $A_5 = true$ in $barva = zelena$)

¹Ideja prihaja iz medicine, kjer odstopanje (npr. števila levkocitov v krvi) od srednje vrednosti za več kot trikratno standardno deviacijo dostikrat kaže na bolezensko stanje.

2.2.2 Operatorji kot predznanje

Očitno se nekaterih konceptov brez pravih operatorjev ne bomo mogli naučiti in narobe: nekateri koncepti bodo z uporabo pravih operatorjev preprosto naučljivi. Na to, katere operatorje bomo uporabili za učenje nekega koncepta, lahko gledamo kot na izbiro potrebnega predznanja za dani problem.

3. Odločitvena drevesa

Tudi pri odločitvenih drevesih sta prisotna transformacijski princip in princip neposredne gradnje konstruktov.

Transformacijski princip (*Yang in sod., 1991*) išče v že zgrajenem drevesu ponavljajoče se vzorce. Predvideva, da niso naključni in jih poskuša nadomestiti s konstrukti. Za zdaj se pristop še ni izkazal, kajti predpostavka o pomenu ponavljanja vzorcev marsikdaj ne drži.

Neposredna gradnja atributov poskuša iz obstoječih atributov zgraditi konstrukte, ki se v vozliščih dreves obnašajo kot binarni atributi. S tem dobijo drevesa večjo izrazno moč in kompaktnost. Primer takšnega algoritma je LFC (*Ragavan in Rendell, 1993; Ragavan in sod., 1993*), ki za gradnjo uporablja le operatorje logične konjunkcije, disjunkcije in negacije. V praksi se je algoritem pokazal za obetavnega, njegova reimplementacija (*Robnik, 1993*) pa je pokazala tudi nekatere omejitve (majhen nabor operatorjev, velika računska zahtevnost).

Slika 1 prikazuje različna odločitvena drevesa za problem parnosti.

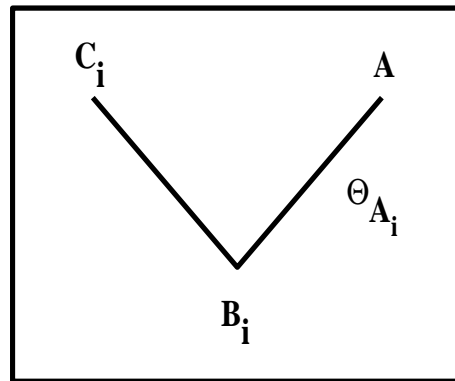
Pod a) je prikazano navadno drevo brez konstrukcije, b) prikazuje drevo, kot ga zgradi LFC (identificirani so posamezni členi v opisu), slika c) pa prikazuje drevo, če imamo na voljo tudi operator primerjave atributov. Pri reševanju problemov, kjer so atributi med seboj odvisni (problemi parnosti, šah, limfografija) je LFC pokazal prednost pred algoritmi brez konstruktivne indukcije in induciral drevesa, ki so natančneje klasificirala nove primere.

Pri odločitvenih drevesih pomeni konstruktivna indukcija pogled naprej. S tem ko iščemo kombinacije atributov, ki jih želimo združiti v konstrukt, simuliramo pregledovanje nekaž vej naprej v še nezgrajeno drevo. Če se

4.1 Preoblikovalni pristopi

Z uvajanjem novih predikatov preoblikujemo že obstoječo teorijo, tako da bi bila le-ta bolj kompaktna in pravilna. Nove predikate iščemo s postopkom inverzne resolucije ali s shemami.

Operator **inverzne resolucije** (ali W-operator ali intrakonstrukcija) obrne postopek deduktivnega sklepanja (slika 2).



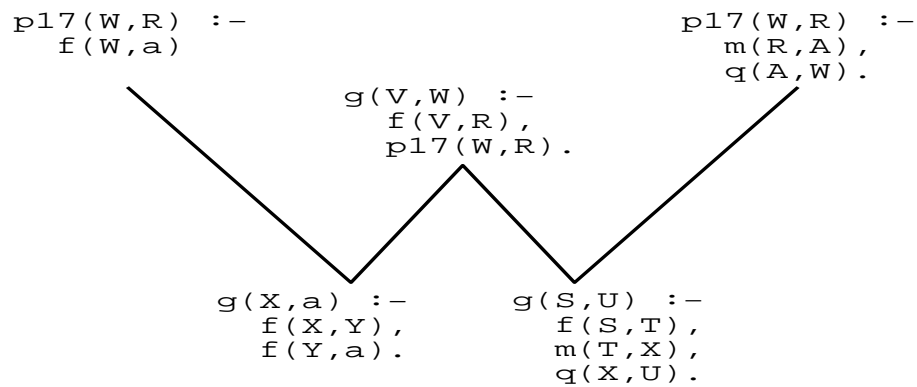
Slika 2: Postopek logičnega sklepanja

Iz množice stavkov $\{B_1, \dots, B_n\}$ intrakonstrukcija skonstruira množico stavkov $\{C_1, \dots, C_n\}$ in stavek A tako, da dobimo stavek B_i iz resolucije A z C_i na nekem literalu $L \in A$.

Primer je na sliki 3. Tvorimo predikat $p17$.

Različni sistemi se med seboj razlikujejo glede argumentov novega predikata ter glede trenutka uporabe in omejitev operatorja; naštejmo nekaj najpomembnejših: CIGOL (*Muggleton in Buntine, 1988*), LFP2 (*Wirth, 1989*), ITOU (*Rouveriol, 1992*) in DREAM (*Banerji, 1991*).

S **shemami** definiramo nove predikate kot kombinacijo že obstoječih literalov v predikatu. Sheme opisujejo koristne kombinacije literalov. Če se med učenjem določena shema izkaže za koristno, jo mora potrditi še prerok. Primera sistemov s shemami sta CIA (*De Raedt in Bruynooghe, 1992*), FOCL (*Silverstein in Pazzani, 1991*). Poglejmo si primer sheme.



Slika 3: Prikaz delovanja W-operatorja

$$S \equiv P(X, Y) \leftarrow Q(X), R(X, Y).$$

Shema S pove, da so koristne kombinacije dveh stavkov, ki sta povezana s skupno spremenljivko.

Če imamo pred uporabo sheme v bazi znanja na voljo naslednji stavek:

$$\begin{array}{c}
 \text{ded}(d, v) \leftarrow \text{moški}(d), \text{moški}(v), \\
 \text{roditelj}(d, m), \text{roditelj}(m, v).
 \end{array}$$

potem dobimo:

$$P(d, m) \leftarrow \text{moški}(d), \text{roditelj}(d, m).$$

Prerok (uporabnik) nam takšno shemo potrdi in jo poimenuje *oce*. V bazo znanja vstavimo:

$$\text{oče}(d, m) \leftarrow \text{moški}(d), \text{roditelj}(d, m).$$

Kasneje v procesu učenja, ko opravi sistem generalizacijo predikata $\text{ded}(X, Y)$ se ponovi tudi procesiranje shem in generalizira se tudi shema.

4.2 Konstrukcija predikatov na zahtevo

Sistemi te vrste poskušajo odkriti stanja, v katerih trenutni opisni jezik ne bi zadostoval in v tem primeru konstruirajo nov predikat. Odločitev o tem je hevristična. Tudi tu se pojavljajo sistemi, ki delujejo na principu inverzne resolucije in shem, opazimo pa tudi druge zanimive pristope, npr. specializacijo pod predpostavko zaprtega sveta (*Bain in Muggleton, 1992*). Često se kot hevristika pojavlja kompresija inducirane teorije oz. MDL (Minimum Description Length).

5. Drugi pristopi

Tudi v drugih pristopih k strojnemu učenju posredno ali neposredno najdemo konstruktivno indukcijo.

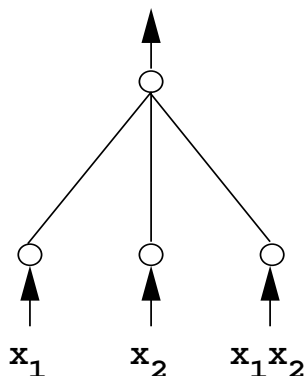
5.1 Bayesov klasifikator

Bayesov klasifikator temelji na predpostavki neodvisnosti atributov; če je ta predpostavka v glavnem izpolnjena, se izkaže za zelo uspešen način učenja (*Kononenko, 1993*). Delno naivni Bayesov klasifikator (*Kononenko, 1991*) išče odvisne attribute s primerjavo pogojnih verjetnosti združenih in nezdruženih atributov. Odkrite odvisnosti upošteva v izračunu klasifikacije kot samostojne attribute.

5.2 Nevronske mreže

Nevroni na skritih nivojih mreže pomenijo konstrukte oziroma koncepte, ki so med vhodno predstavitvijo in izhodnim konceptom. To lastnost izkoriščajo tudi različne metode za izločanje simboličnega znanja iz nevronske mreže.

Malo drugačen pristop je uporabljen pri mreži s funkcijsko povezavo (*Pao, 1989*). Tu imamo poleg običajnih vhodov tudi funkcijske, ki že vnaprej izražajo neko funkcijsko povezavo drugih vhodov. Če funkcijski vhod dejansko izraža neko funkcijsko odvisnost, ki se kaže tudi v izhodnem konceptu, je postopek učenja veliko krajši in uspešnejši. Primer za parnost je na sliki 4.



Slika 4: Mreža s funkcijsko povezavo za problem parnosti drugega reda.

5.3 Prepoznavanje vzorcev

V teoriji prepoznavanja vzorcev najdemo postopke, ki postopno gradijo hierarhijo prepoznavanega objekta, kar spominja na združevanje koncepov v skupine (clustering).

6. Sklep

Čeprav je konstruktivna indukcija že leta med raziskovalnimi usmeritvami na področju strojnega učenja (*Michalski, 1986*), je še vedno razmeroma privlačna razvojna smer. Najbolj obetavne smernice se mi zdijo:

- postopno in kontrolirano razširjanje jezika
- hevristično ugotavljanje potrebnih operatorjev oziroma potrebnega predznanja za dani učni problem
- novi operatorji in prostori, v katerih bodo lažje predstavljeni in izračunljivi
- izmenjava idej z drugimi pristopi
- uporaba idej iz propozicijskega učenja v ILP in narobe
- avtomatizacija potrjevanja koristnih konstruktov (odprava prerokov).

Metodi, ki najbolj obetata pri izpolnitvi zgornjih nalog, se mi zdita:
MDL, ki bi nam lahko pomagal ugotoviti kdaj ustaviti iskanje, in
RELIEFF (*Kononenko, 1994*), ki se zdi učinkovit usmerjevalec iskanja.

Zahvala

Zahvaljujem se Igorju Kononenku za napotila, vspodbudo in komentarje k članku ter Matjažu Kukarju in Urošu Pompetu za komentarje k zgodnejšim različicam tega besedila.

Kratka strokovna biografija:

Avtor je decembra 1993 diplomiral na Fakulteti za elektrotehniko in računalništvo v Ljubljani z diplomsko nalogo "Konstruktivna indukcija z odločitvenimi drevesi". Konec leta 1994 je začel raziskovalno delovati v Laboratoriju za umetno inteligenco pod vodstvom mentorja doc.dr. Igorja Kononenka. Ukvarja se s področjem avtomatskega učenja.

Literatura

- [1] M. Bain, S. Muggleton: Non-Monotonic Learning. In S.Muggleton (ed):*Inductive Logic Programming*, pp. 145-161, Academic Press, 1992
- [2] R.B. Banerji: Learning Theoretical Terms. In S. Muggleton(ed): *Inductive Logic Programming*, pp. 93-112, Academic Press, 1992
- [3] L. DeRaedt, M. Bruynooghe: Interactive concept-learning and constructive induction by analogy. *Machine Learning* 8(2):107-150, 1992
- [4] Thomas G. Dietterich, Ryszard S. Michalski: A Comparative Review of Selected Methods for Learning from Examples. In R. Michalski, J. Carbonnel, T. Mitchell (editors): *Machine Learning: An Artificial Intelligence Approach*, pp. 41-82, Kaufman, Tioga, Palo Alto, CA, 1983
- [5] R. Kerber: ChiMerge: Discretization of Numeric Attributes. *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 123-128, AAAI Press/MIT Press, Menlo Park, 1992
- [6] Dennis Kibler, Pat Langley: Machine Learning as an Experimental Science. In T.G. Dietterich, J.W. Shavlik: *Readings in Machine Learning*. pp. 45-59, Morgan Kaufmann, Palo Alto, CA, 1990
- [7] Kononenko Igor: Estimating attributes: analysis and extensions of RELIEF. In Bergadano F., L. De Raedt (Eds.): *Machine Learning: ECML-94*. 171-182, Springer-Verlag, 1994
- [8] Igor Kononenko: Inductive and Bayesian Learning in Medical Diagnosis. *Applied Artificial Intelligence* 7:317-337, 1993
- [9] Igor Kononenko: Semi-Naive Bayesian classifier. In Kodratoff Y.(ed.): *Proceedings of European Working Session on Learning*, pp. 206-219, Springer-Verlag, New York, 1991
- [10] R.S. Michalski, R. Stepp: Learning from observation: conceptual clustering. In R. Michalski, J. Carbonnel, T. Mitchell (editors): *Machine Learning: An Artificial Intelligence Approach*, pp. 331-364, Kaufman, Tioga, Palo Alto, CA, 1983

- [11] R.S. Michalski: A Theory and Methodology of Inductive Learning. In R. Michalski, J. Carbonnel, T. Mitchell (editors): *Machine Learning: An Artificial Intelligence Approach*, pp. 83-134, Kaufman, Tioga, Palo Alto, CA, 1983
- [12] R.S. Michalski: Understanding the nature of learning: issues and research directions. In R. Michalski, J. Carbonnel, T. Mitchell (editors): *Machine Learning: An Artificial Intelligence Approach*, pp. 3-25, Kaufmann, Los Altos, CA, 1986
- [13] S. Muggleton, W. Buntine: Machine Invention of First-Order Predicates by Inverting Resolution. In S. Muggleton(ed.): *Logic Programming*, pp. 261-280, Academic Press 1992
- [14] S. Muggleton: *Inductive Acquisition of Expert Knowledge*. Addison-Wesley, Workingham, England, 1990
- [15] S. Muggleton (ed.): *Inductive Logic Programming*, Academic Press, 1992
- [16] Yon-Han Pao: *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley, Reading, Massachusetts, 1989
- [17] Bernhard Pfahringer: Robust Constructive Induction. *Proceedings der 18. Deutsche Jahrestagung für Künstliche Intelligenz*, Springer, 1994
- [18] Bernhard Pfahringer: *Controlling Constructive Induction in CiPF: An MDL Approach*. In F. Bergadano, L. De Raedt (Eds.): *Machine Learning: ECML-94*, pp. 242-256, Springer-Verlag, 1994
- [19] H. Ragavan , L. Rendell: Lookahead Feature Construction for Learning Hard Concepts. In *Proceedings of the Tenth International Machine Learning Conference*, pp. 252-259, Morgan Kaufman, 1993,
- [20] H. Ragavan, L. Rendell, M. Shaw, A. Tessmer: Learning Complex Real-World Concepts through Feature Construction. Technical Report. The Beckman Institute, University of Illinois, Urbana, 1993
- [21] J. Rissanen: Modelling by Shortest Data Description. In *Automatica* 14:465-471, 1978

- [22] Marko Robnik: Konstruktivna indukcija z odločitvenimi drevesi. Diplomsko delo, Univerza v Ljubljani, Fakulteta za elektrotehniko in računalništvo, Ljubljana 1993
- [23] C. Rouveriol: ITOU: Induction of first-order theories. In *First International Workshop on Inductive Logic Programming*, 1991; also in S. Muggleton (ed.): *Logic Programming*, pp. 63-92, Academic Press, 1992
- [24] G. Silverstein, M.J. Pazzani: Relational Cliches: Constraining Constructive Induction During Relational Learning. In *Proceedings of Machine Learning Workshop*, 1991
- [25] Irene Stahl: An Overview of Predicate Invention Techniques in ILP. ESPRIT BRA 6020: Inductive Logic Programming, 1993
- [26] D.S. Yang, L. Rendell, G. Blix: A Scheme for Feature Construction and a Comparison of Empirical Methods. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pp. 699-704, 1991
- [27] Paul E. Utgoff: Shift of Bias for Inductive Concept Learning. In R. Michalski, J. Carbonnel, T. Mitchell (editors): *Machine Learning: An Artificial Intelligence Approach, Volume II*, pp. 107-148, Kaufman, Los Altos, 1986
- [28] R. Wirth: Completing Logic Programs by Inverting Resolution. In K. Morik (ed.): *Proceedings of the 4th European Working Session on Learning*, pp. 239-250, Pitman, 1989