

# IZBOLJŠEVANJE PREPOZNAVANJA AKTIVNOSTI IZ POLOŽAJEV ZNAČK

Mitja Luštrek

Odsek za inteligentne sisteme

Institut Jožef Stefan

Jamova cesta 39, 1000 Ljubljana, Slovenija

Telefon: +386 1 4773380; telefaks: +386 1 4773131

E-pošta: mitja.lustrek@ijs.si

## POVZETEK

Iz koordinat štirih značk, pritrjenih na telo, smo s strojnim učenjem prepoznavali aktivnosti. Koordinate smo pridobili s sistemom za določanje položaja Ubisense. Ugotovili smo, da je najprimernejši algoritem za strojno učenje naključni gozd in da je atributni vektor najbolje sestaviti iz niza atributov, ki pripadajo desetim zaporednim zajemom koordinat značk. Glavni prispevek so štirje načini za izboljševanje prepoznavanja aktivnosti, ki upoštevajo časovno zaporedje prepoznav. Z njimi se je klasifikacijska točnost povečala za 1,66 odstotne točke.

## 1 UVOD

Prebivalstvo v Evropi in razvitem svetu nasploh se naglo stara. To pomeni, da je starejših, ki so potrebni skrbi, vedno več, mladih, ki bi zanje lahko skrbeli, pa vedno manj. Zato je veliko raziskav usmerjenih v iskanje tehnoloških pripomočkov za skrb za starejše. Eden izmed takih pripomočkov se razvija v projektu Confidence [1].

Cilj projekta Confidence je starejše osebe opremiti z radijskimi značkami, s pomočjo katerih bo moč zaznavati položaj telesa in na podlagi tega ugotavljati zdravstvene težave, posebej padce. V prispevku se ukvarjamo s prepoznavanjem aktivnosti, ki je predpogoj za nadaljnjo analizo obnašanja uporabnika.

Ker so značke in senzorji, ki bodo uporabljeni v projektu, še v razvoju, smo namesto njih uporabili sistem za določanje položaja Ubisense [7]. Ta sistem uporablja radijsko tehnologijo in je najboljši komercialni približek strojni opremi, ki bo razvita v projektu. V preteklih poizkusih [3][4][5] smo uporabljali infrardeči sistem Smart [2], katerega izmerkom smo zaradi bistveno večje natančnosti od pričakovane v Confidence dodajali Gaussov šum. Takšni podatki so bili manj realistični od Ubisensovih.

Aktivnosti smo prepoznavali s strojnim učenjem. V prispevku najprej primerjamo klasifikacijsko točnost različnih algoritmov za strojno učenje na Ubisensovih podatkih in načine sestavljanja atributnega vektorja. Nato se posvetimo glavni temi prispevka, to je načinom za izboljševanje prepoznavanja aktivnosti, ki prepoznane aktivnosti popravlja na podlagi njihovega zaporedja v času.

## 2 UČNI IN TESTNI PODATKI

Za učne in testne podatke smo uporabili 25 posnetkov obnašanja petih oseb, po pet za osebo. Posnetki so sestavljeni iz 32 aktivnosti v osmih sklopih. Prepoznavali smo osem različnih aktivnosti: hojo, normalno sedenje, sedenje na tleh, ležanje, na vseh štirih, normalno spuščanje (sedanje in leganje), dviganje (iz sedenja in ležanje) ter padanje. Sestava posnetkov je prikazana v Tabeli 1.

Sklop	Aktivnosti
Hoja	hoja
Normalno sedanje	spuščanje, sedenje, dviganje
	hoja
Spotikanje	padanje, ležanje
	dviganje, hoja
Normalno leganje	spuščanje, ležanje, dviganje
	hoja
Omedlavanje stoje	padanje, ležanje
	dviganje, hoja
Hitro sedanje	spuščanje, sedenje
Zdrs s stola	padanje, sedenje na tleh
	dviganje, hoja
Hitro leganje	spuščanje, ležanje, dviganje
	hoja
Iskanje predmeta pod mizo	spuščanje, na vseh štirih, ležanje
	dviganje, hoja

Tabela 1. Sestava učnih/testnih posnetkov.

Osebe na posnetkih so bile opremljene z štirimi značkami: na obeh stopalih, na pasu in za vratom. Koordinate smo zajemali s frekvenco 9,25 Hz. Predprocesirane so bile z medianinim in Kalmanovim filtrom [6].

## 3 OSNOVNO PREPOZNAVANJE AKTIVNOSTI

Za prepoznavanje aktivnosti s strojnim učenjem moramo iz koordinat značk izračunati atributni vektor. Ob vsakem zajemu koordinat iz njih izračunamo nabor atributov, nakar več zaporednih naborov združimo v atributni vektor. To storimo z metodo drsečega okna, kar pomeni, da se vsak atributni vektor od prejšnjega razlikuje za en nabor.

V preteklosti [5] smo uporabljali več različnih vrst naborov atributov: attribute v referenčnem koordinatnem sistemu, attribute v štirih različnih telesnih koordinatnih sistemih in kote med deli telesa. A glede na to, da so se referenčni atributi obnesli najboljše in da je ostale z le štirimi značkami težko izračunati (v preteklosti smo uporabljali do 12 značk), smo se tokrat omejili na referenčne. Ti so bili:

- koordinate z vseh značk,
- hitrosti vseh značk,
- hitrosti vseh značk v smeri z,
- razdalje med vsemi pari značk,
- razdalje med vsemi pari značk v smeri z,
- razdalje med vsemi pari značk v ravnini xy.

Koordinat x in y nismo uporabili, ker so odvisne od mesta v prostoru, kjer se aktivnost dogaja, to pa nas ne zanima.

### 3.1 Algoritmi za strojno učenje

Najprej smo za Ubisensove podatke ugotovili, kateri algoritem za strojno učenje je za prepoznavanju aktivnosti najprimernejši. Pri tem smo se oprli na dognanja iz poizkusov s Smartom [5]. Attribute smo v atributne vektorje združevali tako, da smo deset zaporednih naborov atributov nanizali enega za drugega. Od algoritmov za strojno učenje smo primerjali tistih pet, ki so se najboljše obnesli na Smartovih podatkih. Poizkuse smo pognali v paketu za strojno učenje Weka [8] s privzetimi nastavitvami. Klasifikacijsko točnost smo dobili z desetkratnim prečnim preverjanjem na podatkih, opisanih v razdelku 2. Rezultati so prikazani v Tabeli 2. Vidimo lahko, da največjo točnost da naključni gozd, ostale metode z izjemo podpornih vektorjev pa so blizu. Zanimivo je, da so podporni vektorji, ki so se s Smartovimi podatki obnesli najboljše, tu najslabši.

Algoritem	Klasifikacijska točnost
3-najbližji sosedi	95,5
podporni vektorji	87,4
naključni gozd	<b>96,7</b>
“bagging”	94,2
“boosting” Adaboost M1	94,1

Tabela 2. Klasifikacijska točnost [%] z različnimi algoritmi za strojno učenje.

Ker je cilj projekta Confidence izdelati sistem, ki bo deloval na novem uporabniku brez prehodnega učenja, smo izračunali še klasifikacijsko točnost na način »izpusti eno osebo«. To pomeni, da smo za učne podatke uporabili posnetke štirih oseb, za testne pa posnetke pete, kar smo ponovili za vseh pet oseb. Klasifikacijska točnost z algoritmom naključni gozd je bila v tem primeru 81,5 %. Razlog za tolikšno zmanjšanje v primerjavi s prečnim preverjanjem je, da pri slednjem odseki iste aktivnosti (ki so si zelo podobni) pridejo med učne in testne podatke.

### 3.2 Združevanje atributov v atributne vektorje

Pri združevanju naborov atributov znotraj drsečega okna v atributne vektorje je mogoče spreminjati število naborov in

način združevanja. Uporabili smo od enega do deset naborov. Več ni smiselno, saj deset naborov obsega že 1,08 s, nekatere aktivnosti pa so krajše od ene sekunde.

Uporabili smo tri načine združevanja naborov atributov v atributne vektorje. Prvi način je že omenjeno nizanje. Drugi je povprečenje vsakega atributa čez vse nabore znotraj drsečega okna. Tretji pa je izračun naklona linearne interpolacije vsakega atributa čez vse nabore znotraj okna; tega smo uporabili le skupaj s povprečenjem.

Primerjava različnih števil naborov atributov in načinov združevanja je prikazana v tabeli 3. Kot algoritem za strojno učenje je bil uporabljen naključni gozd, rezultati pa so dobljeni z desetkratnim prečnim preverjanjem. Vidimo, da se več naborov vedno obnese bolje kot manj in da je najboljši način združevanja najpreprostejši: nizanje.

Združevanje \ št. naborov	1	3	5	10
Nizanje	96,0	96,1	96,3	<b>96,7</b>
Povprečenje		96,1	96,2	96,5
Povprečenje + naklon	/	95,3	96,0	96,5

Tabela 3. Klasifikacijska točnost [%] z različnimi načini združevanja atributov v atributne vektorje.

Ker so si klasifikacijske točnosti vseh treh načinov združevanja podobne, smo jih pri desetih naborih atributov primerjali še na način »izpusti eno osebo«. Pri nizanju smo dobili točnost 81,5 %, povprečenje in povprečenje + naklon pa sta bila z 81,0 % in 80,9 % tudi tu slabša.

## 4 IZBOLJŠANO PREPOZNAVANJE AKTIVNOSTI

Zamislimo si, da klasifikator za prepoznavanje aktivnosti kot aktivnost desetkrat zapored prepozna hojo. Če temu enkrat sledi spuščanje, nismo zelo gotovi, da je do spuščanja res prišlo. Če mu sledi več spuščanj, smo pa dokaj prepričani, da se uporabnik res spušča. A če po drugi strani nemu spuščanju sledi več hoj, se zdi bolj verjetno, da je bilo spuščanje prepoznano napačno. Takisto moramo posumiti, da je prišlo do napake, če hoji sledi ležanje, saj bi vmes moralo biti spuščanje ali padanje. Zato si želimo informacije o zaporednih prepoznanih aktivnosti izkoristiti za izboljšanje prepoznavanja aktivnosti.

### 4.1 Opis izboljšav

Prepoznane aktivnosti opazujemo v oknu dolžine  $W$ . Naj bodo  $[a_1^P, \dots, a_W^P]$  prepoznane aktivnosti in  $[a_1, \dots, a_W]$  pripadajoče prave aktivnosti. Privzemimo, da v vsakem oknu lahko nastopata največ dve različni pravi aktivnosti: leva  $a_L$  in desna  $a_R$ . To pomeni, da za vsak par pravih aktivnosti obstaja  $W$  različnih oken (indeks pomeni število pojavitev desne aktivnosti):

$$w_0(a_L, a_R) = [a_L, a_L, \dots, a_L, a_L, a_L]$$

$$w_1(a_L, a_R) = [a_L, a_L, \dots, a_L, a_L, a_R]$$

$$w_2(a_L, a_R) = [a_L, a_L, \dots, a_L, a_R, a_R]$$

...

$$w_{W-1}(a_L, a_R) = [a_L, a_R, \dots, a_R, a_R, a_R]$$

Če obstaja  $A$  različnih aktivnosti, imamo tako  $A(A - 1)W$  različnih oken pravih aktivnosti. Za vsako tako okno lahko določimo verjetnostno porazdelitev prepoznanih aktivnosti. To s pomočjo učnih podatkov storimo na štiri načine.

**Prvi način** za vsako okno pravih aktivnosti prešteje vsa okna prepoznanih aktivnosti, ki se pri tistem oknu pravih aktivnosti pojavijo v učnih podatkih. To pomeni, da če npr. pri oknu pravih aktivnostih [hoja, spuščanje, spuščanje] prepoznamo [hoja, hoja, padanje], za ena povečamo števec takih primerov. Če je  $W$  velik, je možnih oken prepoznanih aktivnosti ogromno, zato večine v učnih podatkih ni. Da ublažimo razliko med nikoli in redko videnimi okni prepoznanih aktivnosti, uporabimo Laplaceovo glajenje: na začetku števec vseh oken prepoznanih aktivnosti nastavimo na 1. Takšno glajenje uporabimo tudi pri ostalih načinih.

Prvi način tako določi verjetnost vsakega okna prepoznanih aktivnosti pri vsakem oknu pravih aktivnosti:

$$P([a_1^P, \dots, a_W^P] | [a_1, \dots, a_W])$$

Ko želimo določiti okno pravih aktivnosti za neko okno prepoznanih aktivnosti  $[a_1^P, \dots, a_W^P]$ , za vsa okna pravih aktivnosti  $w_i(a_L, a_R)$  preverimo, kolikšna je verjetnost  $P([a_1^P, \dots, a_W^P] | w_i(a_L, a_R))$ . Za tisto okno pravih aktivnosti, za katero je ta verjetnost največja, odločimo, da ustreza dejanskim aktivnostim.

**Drugi način** je milejši od prvega in za vsako okno pravih aktivnosti prešteje le pojavitve prepoznanih aktivnostih na posamičnih mestih okna. To pomeni, da če pri oknu pravih aktivnostih [hoja, spuščanje, spuščanje] prepoznamo [hoja, hoja, padanje], za ena povečamo števec hoj na prvem in na drugem mestu tega okna pravih aktivnosti ter padanj na tretjem mestu. To pomeni, da določimo verjetnosti:

$$P(\alpha_i^P = \hat{a}_j | [a_1, \dots, a_W]); i = 1 \dots W, j = 1 \dots A$$

Z  $\alpha_i$  označimo aktivnosti na  $i$ -tem mestu v oknu, z  $\hat{a}_j$  pa  $j$ -to vrednost, ki jo ta aktivnost lahko zavzame.

Ko želimo določiti okno pravih aktivnosti za neko okno prepoznanih aktivnosti  $[a_1^P, \dots, a_W^P]$ , za vsa okna pravih aktivnosti  $w_i(a_L, a_R)$  preverimo, kolikšna je verjetnost:

$$P(\alpha_1^P = a_1^P | w_i(a_L, a_R)) \cdot \dots \cdot P(\alpha_W^P = a_W^P | w_i(a_L, a_R))$$

Za tisto okno pravih aktivnosti, za katero je ta verjetnost največja, odločimo, da ustreza dejanskim aktivnostim.

**Tretji način** je še milejši in za vsako okno pravih aktivnosti prešteje le pojavitve prepoznanih aktivnostih na mestih, ki pripadajo levi in desni pravi aktivnosti. To pomeni, da če pri oknu pravih aktivnostih [hoja, spuščanje, spuščanje] prepoznamo [hoja, hoja, padanje], za ena povečamo števec hoj pri levi pravi aktivnosti (hoji) ter števec hoj in padanj pri desni pravi aktivnosti (spuščanju). To pomeni, da določimo verjetnosti:

$$P(\alpha_L^P = \hat{a}_j | [a_1, \dots, a_W]); j = 1 \dots A$$

$$P(\alpha_R^P = \hat{a}_j | [a_1, \dots, a_W]); j = 1 \dots A$$

Ko želimo določiti okno pravih aktivnosti za neko okno prepoznanih aktivnosti  $[a_1^P, \dots, a_W^P]$ , za vsa okna pravih aktivnosti  $w_i(a_L, a_R)$  preverimo, kolikšna je verjetnost

$$P(\alpha_L^P = a_1^P | w_i(a_L, a_R)) \cdot \dots \cdot P(\alpha_L^P = a_{W-i}^P | w_i(a_L, a_R)) \cdot P(\alpha_R^P = a_{W-i+1}^P | w_i(a_L, a_R)) \cdot \dots \cdot P(\alpha_W^P = a_W^P | w_i(a_L, a_R))$$

Za tisto okno pravih aktivnosti, za katero je ta verjetnost največja, odločimo, da ustreza dejanskim aktivnostim.

**Četrty način** je enak tretjemu, le da je še milejši. Za neko okno pravih aktivnosti pojavitve prepoznanih aktivnosti ne štejemo le, če nastopijo pri tistem oknu pravih aktivnostih, ampak če nastopijo kjerkoli pri pravih aktivnostih iz okna.

**Združevanje** več načinov izboljševanja prepoznavanja aktivnosti je enostavno. Naj bo  $P_k([a_1^P, \dots, a_W^P] | w_i(a_L, a_R))$  verjetnost, ki jo za neko okno prepoznanih aktivnosti in okno pravih aktivnosti  $w_i(a_L, a_R)$  določi  $k$ -ti način. Najprej za vsak način te verjetnosti za vsa okna pravih aktivnosti  $i = 0 \dots W - 1, a_L, a_R = \hat{a}_1 \dots \hat{a}_A$  normaliziramo na interval  $[0, 1]$ , da načine izenačimo. Nato pa za vsako okno pravih aktivnosti verjetnosti za vse načine  $k = 1 \dots 4$  seštejemo. Za okno pravih aktivnosti z največjim seštevkom odločimo, da ustreza dejanskim aktivnostim.

**Na koncu** je treba biti pozoren še na nekaj podrobnosti. Prepoznavanje aktivnosti izboljšujemo z drsečim oknom. To pomeni, da vsako aktivnost popravljamo  $W$ -krat: od takrat, ko se pojavi na desnem robu okna, do takrat, ko pride do levega roba. Popravki se lahko razlikujejo. A največ vemo o aktivnosti, ko priptuje do levega roba okna, zato jo dokončno popravimo šele takrat.

Če privzamemo, da je dokončno popravljena aktivnost pravilna, se mora okno, ki ji sledi, začeti s to aktivnostjo ali pa more vsebovati same enake aktivnosti. V nasprotnem primeru namreč kršimo predpostavko, da znotraj okna dolžine  $W$  nastopata največ dve različni aktivnosti. Brez te omejitve bi se lahko npr. zgodilo, da bi za prave aktivnosti določili [sedenje, sedenje, sedenje], okno pomaknili za eno aktivnost naprej (podčrtana aktivnost bi se zdaj štela za pravilno), za prave aktivnosti določili [spuščanje, sedenje, sedenje], okno spet pomaknili naprej in za prave aktivnosti določili [dviganje, sedenje, sedenje]. Rezultat bi bil zaporedje dokončno popavljenih aktivnosti sedenje, spuščanje, dviganje, v resnici pa bi uporabnik ves čas sedel. Z omejitvijo bi v drugem in tretjem koraku v poštev prišla samo okna pravih aktivnosti, ki se začenjajo s sedenjem ali vsebujejo same enake aktivnosti. Ker bi bila okna iz treh spuščanj in dviganj premalo verjetna, bi bilo zaporedje dokončno popavljenih aktivnosti sestavljeno iz treh sedenj.

## 4.2 Poizkusi

Za preizkus načinov za izboljševanje prepoznavanja aktivnosti sta potrebni dve testni množici: ena za učenje osnovnega klasifikatorja za prepoznavanje aktivnosti in druga za učenje mehanizma za izboljševanje. Če bi namreč tako osnovni klasifikator kot mehanizem za izboljševanje učili na istih podatkih, bi klasifikator na njih delal predobro,

da bi se mehanizem sploh lahko česa naučil. Uporabili smo podatke, opisane v razdelku 2. Kot učno množico za klasifikator smo uporabili prvi in četrti posnetek vsake osebe, kot učno množico za izboljševanje drugi in peti posnetek, kot testno množico pa tretji posnetek.

Najprej smo preverili, kolikšno izboljšavo prinese vsak izmed štirih načinov za izboljševanje glede na klasifikacijsko točnost brez izboljševanja. Osnovni klasifikator za prepoznavanje aktivnosti smo zgradili z algoritmom naključni gozd in atributnimi vektorji iz desetih nanizanih naborov atributov. Za dolžino okna za izboljševanje smo izbrali  $W = 10$ . Rezultati so prikazani v tabeli 4. Vidimo, da se najbolje obnese način 2 in vsi načini skupaj. Način 2 očitno najbolje ujame ravnotežje med natančnostjo opisa napak, ki jih dela osnovni klasifikator, in zahtevo po dovolj učnih podatkih. Najslabši je način 1, ki ga pesti pomanjkanje učnih podatkov, slabša pa sta tudi načina 3 in 4, ki najbrž premalo natančno opisujeta napake, ki jih dela osnovni klasifikator. Odločili smo se, da bomo v prihodnje uporabljali vse načine skupaj. Kot prvo niso slabši od nobenega posamičnega načina, kot drugo pa je moč pričakovati, da se bodo obnesli pri količini podatkov, ki najbolj ustreza kateremukoli izmed načinov.

Način izboljševanja	Klasifikacijska točnost
Brez	83,4
Način 1	77,5
Način 2	<b>84,7</b>
Način 3	84,3
Način 4	84,6
Vsi načini skupaj	<b>84,7</b>

Tabela 4. *Klasifikacijska točnost [%] z različnimi načini izboljševanja.*

V naslednjem koraku smo preverili, kakšen je vpliv velikosti okna  $W$  na izboljševanje. V levi polovici tabele 4 vidimo, da je najprimernejše okno velikosti 20, s katerim se klasifikacijska točnost s 83,4 % dvigne na 85,0 %. Bržkone gre tudi v tem primeru za najboljše ravnotežje med natančnostjo opisa napak, ki jih dela osnovni klasifikator za prepoznavanje aktivnosti, in zahtevo po učnih podatkih.

Okno	Izboljšanje	Št. naborov	Izboljšanje
5	0,62	1	1,74
10	1,30	3	2,08
15	1,47	5	2,01
20	<b>1,66</b>	10	1,66
25	1,41		

Tabela 5. *Izboljšanje klasifikacijske točnosti [v odstotnih točkah] z različnimi velikostmi okna za izboljševanje in različnimi števili naborov atributov v atributnem vektorju.*

Na koncu smo pri oknu za izboljševanje velikosti  $W = 20$  preverili še, kakšen je vpliv števila naborov atributov v atributnem vektorju. Domnevali smo, da bo izboljšanje pri manj naborih večje, saj krajši atributni vektorji sami po sebi zajamejo manj informacij o dogajanju v času. V desni

polovici tabele 4 vidimo, da je bila domneva pravilna. A ker klasifikacijska točnost osnovnega klasifikatorja pri manj naborih pade hitreje, kot jo izboljševanje poveča, je še vedno najbolje uporabiti deset naborov.

## 5 ZAKLJUČEK

V prispevku smo določili najprimernejši algoritem za strojno učenje in attribute za prepoznavanje aktivnosti iz Ubisensovih podatkov. Ugotovili smo, da naključni gozd, ki kot atributni vektor uporablja niz desetih zaporednih naborov atributov posamičnih položajev telesa, doseže klasifikacijsko točnost 96,7 % pri desetkratnem prečnem preverjanju in 81,5 % na način »izpusti eno osebo«.

Razvili smo štiri načine izboljševanja prepoznavanja aktivnosti, ki upoštevajo informacije o časovnem zaporedju prepoznav. Z vsemi štirimi skupaj se je klasifikacijska točnost dvignila za 1,66 odstotne točke. Ta dvig se zdi majhen, a glede na to, da večina posegov v prepoznavanje prinaša podobno velike spremembe, je možno, da brez radikalno drugačnega pristopa več ni mogoče doseči.

V prihodnje nameravamo mehanizem za izboljševanje temeljiteje preizkusiti, denimo na način »izpusti eno osebo«. Poleg tega razmišljamo o mehanizmu za izboljševanje, ki bi poleg časovnega zaporedja upošteval tudi podobnosti med aktivnostmi.

## Zahvala

Raziskave, opisane v prispevku, je financirala Evropska skupnost s Sedmim okvirnim programom pod številko projekta 214986. Znatno del programja, uporabljenega pri delu, opisanem v prispevku, je napisal Boštjan Kaluža.

## Literatura

- [1] Confidence. <http://www.confidence-eu.org/> [2008-09-15]
- [2] eMotion. Smart motion capture system. <http://www.emotion3d.com/smart/smart.html> [2008-09-15]
- [3] Luštrek, M., Gams, M., in Veléz, I. (2009). Posture and movement monitoring for ambient assisted living. IST-Africa 2009.
- [4] Luštrek, M., in Gams, M. (2008). Prepoznavna položaja telesa s strojnimi učenjem. Informacijska družba 2009, str. 30-33.
- [5] Luštrek, M., in Kaluža, B. (2009). Fall Detection and Activity Recognition with Machine Learning. Informatica 33 (2), 205–212.
- [6] Maybeck, P. S. (1979). Stochastic models, estimation, and control. Mathematics in Science and Engineering 141.
- [7] Ubisense. <http://www.ubisense.net/> [2008-09-15]
- [8] Witten, I. H., in Frank, E. (2005). Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann, San Francisco, ZDA.