

1. domača naloga

A Moving Average based Filtering System with its Application to Real-time QRS Detection

Živa Škof

Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

V poročilu je opisan algoritem QRS detektorja, ki temelji na filtru premikajočega se povprečja in zaznava srčne utripe v realnem času. Algoritem je sestavljen iz treh pomembnih korakov, na podlagi katerih zazna v povprečju več kot 99 odstotkov srčnih utripov iz podatkovne baze: MIT-BIH Arrhythmia Database [1].

1 Uvod

QRS kompleks predstavlja vrhove ECG signalov in depolarizacijo prekatov. Uporablja se za merjenje in detekcijo srčnih utripov. Na podlagi ECG signalov lahko določimo stanje srca in ugotovimo morebitne nepravilnosti [2]. V sklopu prve domače naloge sem implementirala algoritem, ki temelji na obstoječem algoritmu, ki je opisan v članku z naslovom A Moving Average based Filtering System with its Application to Real-time QRS Detection [3]. V članku je na podlagi treh glavnih korakov implementiran algoritem, ki je kljub svoji preprostosti učinkovit in hiter. Pravilno zazna več kot 99.5 procentov QRS kompleksa iz standardne ECG baze. Algoritem, kot omenjeno, temelji na treh korakih, ki si sledijo v naslednjem zaporedju. Najprej čez ECG signal izvedemo High-Pass filter, nad katerim izvedemo Low-Pass filter in nato sledi še zadnji korak odločevanja.

2 Metode in pristopi

Za implementacijo algoritma sem uporabila MATLAB [4]. Podatki, na katerih sem svoj algoritem testirala, so del MIT-BIH Arrhythmia Database [1] baze, ki sem jih pridobila z zagonom skripte mitdb.sh in jih sproti prevedla v .mat format, ki je potreben za rokovanje z MATLABOM.

Ker sem svoj algoritem analizirala na celotni bazi, sem za iteracijo skozi podatke uporabila program readFile-Run.m, iz katerega sem klicala funkcijo Detector.

V funkciji Detector sem določila vrednosti parametrov M, ki predstavlja velikost filtra, MW, ki predstavlja velikost premikajočega se okna pri Low-Pass filtru in MWD, ki predstavlja velikost okna v fazi odločitve. Iz te funkcije sem klicala funkcijo QRSDetect, kjer sem implementirala QRS detector. Funkcija je vrnila zaznane srčne utripe, ki sem jih v funkciji Detect prevedla v .asc format.

Najpomembnejša funkcija je že prej omenjena funkcija QRSDetect. Nad ECG posnetekoma srčnih utripov sem izvedla naslednje tri korake.

- High-Pass filter, ki zaduši nezaželeno nizkonivjske ECG signale in poudari QRS kompleks. Pri izračunu High-Pass filtra sem se sklicevala na enačbo podano v referenčnem članku[3].
- Low-Pass filter izvede usmerjanje in nelinearno ojačitev signala. Z oknom, katerega velikost določimo mi, se premikamo čez signal in seštejemo in kvadriramo točke trenutnega okna. S kvadriranjem ohranjamo lastnosti QRS signala, zgladimo nizke amplitude visokih frekvenc in se znebimo negativnih vrednosti.
- V koraku odločevanja sem zopet skozi signal iterirala z oknom, ki sem mu predhodno določila vrednost. V prvi iteraciji sem poiskala največjo vrednost in jo shranila v threshold. V vsakem naslednjem koraku sem poiskala največjo vrednost in jo nato primerjala z vrednostjo thresholda. Če je bila nova vrednost večja, sem threshold posodobila z enačbo, ki je zapisana v članku [3]. Vsakič, ko sem našla tako vrednost, sem s številko 1 označila vrh in si to shranila v y na mesto kjer se utrip/vrh nahaja. Položaj maksimalne vrednosti je eden od parametrov, ki jih vrne max().

Ko so bili vsi koraki algoritma zaključeni, sem poiskala vse 1 v y, ki predstavljajo najden utrip/vrh in jih shranila v utrip.

Ko so se v funkciji Detector zgenerirale .asc datoteke, sem jih s skripto convertToQRS.sh pretvorila v .qrs format. Zadnji korak je bila evalvacija rezultatov, ki sem jo izvedla s skripto evaluate.sh, ki je z bxb in sumstats ustvarila končno datoteko finalResults.txt, kjer se nahaja celotna analiza utripov in za nas najbolj pomembna podatka občutljivost (Se) in pozitivna napoved (+P). Ter povprečni vrednosti obeh, za kar je poskrbel ukaz sumstats.

3 Rezultati

Za vrednosti parametrov, ki so sledeče, so rezultati prikazani na sliki 1 pri M=5, MW=10, MWD=140, alpha=0.05

in $\gamma=0.15$.

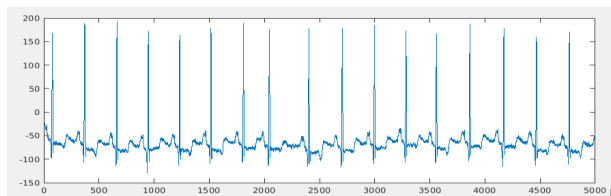
Record	Nn'	Vn'	Fn'	On'	Nv	Vv	Fv'	Ov'	No'	Vo'	Fo'	Q Se	Q +P	V Se	V +P	V FPR
100	2269	1	0	289	0	0	0	0	3	0	0	99.87	88.71	0.00	-	0.000
101	1856	0	2	238	0	0	0	0	7	0	0	99.62	88.65	-	-	0.000
102	97	4	2083	255	0	0	0	0	2	0	1	99.86	89.54	0.00	-	0.000
103	2084	0	0	311	0	0	0	0	0	0	0	100.00	87.01	-	-	0.000
104	103	2	2057	358	0	0	0	0	0	0	7	99.69	86.12	0.00	-	0.000
105	2515	41	5	596	0	0	0	0	11	0	0	99.57	81.12	0.00	-	0.000
106	1506	493	0	248	0	0	0	0	1	27	0	98.62	88.96	0.00	-	0.000
107	0	57	2077	770	0	0	0	0	0	2	1	99.86	73.48	0.00	-	0.000
108	1728	17	2	1099	0	0	0	0	16	0	0	99.99	61.38	0.00	-	0.000
109	2490	37	2	678	0	0	0	0	2	1	0	99.88	78.86	0.00	-	0.000
111	2122	1	0	575	0	0	0	0	1	0	0	99.95	78.69	0.00	-	0.000
112	2508	0	0	428	0	0	0	0	31	0	0	98.78	85.42	-	-	0.000
113	1794	0	0	240	0	0	0	0	1	0	0	99.94	88.20	-	-	0.000
114	1831	43	4	292	0	0	0	0	1	0	0	99.95	86.54	0.00	-	0.000
115	1952	0	0	286	0	0	0	0	1	0	0	99.95	87.22	-	-	0.000
116	2283	108	0	412	0	0	0	0	20	1	0	99.13	85.30	0.00	-	0.000
117	1534	0	0	238	0	0	0	0	1	0	0	99.93	86.57	-	-	0.000
118	2241	15	0	352	0	0	0	0	21	0	0	99.03	86.50	0.00	-	0.000
119	1543	444	0	378	0	0	0	0	0	0	0	100.00	84.02	0.00	-	0.000
121	1860	1	0	477	0	0	0	0	2	0	0	99.89	79.60	0.00	-	0.000
122	2467	0	0	494	0	0	0	0	9	0	0	99.64	83.32	-	-	0.000
123	1513	0	0	213	0	0	0	0	2	3	0	99.67	87.66	0.00	-	0.000
124	1567	47	5	486	0	0	0	0	0	0	0	100.00	79.95	0.00	-	0.000
200	1770	824	1	603	0	0	0	0	3	2	1	99.77	81.14	0.00	-	0.000
201	1724	198	2	325	0	0	0	0	39	0	0	98.01	85.55	0.00	-	0.000
202	2113	18	1	337	0	0	0	0	3	1	0	99.81	86.35	0.00	-	0.000
203	2477	417	2	698	0	0	0	0	54	27	3	97.18	80.58	0.00	-	0.000
205	2572	62	10	347	0	0	0	0	2	9	1	99.55	88.40	0.00	-	0.000
207	1644	192	0	387	0	0	0	0	6	18	0	98.71	77.77	0.00	-	0.000
208	1556	950	367	385	0	0	0	0	32	42	8	97.23	88.18	0.00	-	0.000
209	2992	1	0	406	0	0	0	0	12	0	0	99.60	88.06	0.00	-	0.000
210	2432	177	10	498	0	0	0	0	13	18	0	98.83	84.02	0.00	-	0.000
212	2747	0	0	387	0	0	0	0	1	0	0	99.96	87.65	-	-	0.000
213	2666	219	362	564	0	0	0	0	3	1	0	99.88	85.20	0.00	-	0.000
214	1906	251	3	348	0	0	0	0	97	5	0	95.49	86.12	0.00	-	0.000
215	3191	159	1	547	0	0	0	0	7	5	0	99.64	85.97	0.00	-	0.000
217	244	158	1802	675	0	0	0	0	0	4	0	99.82	76.55	0.00	-	0.000
219	2087	64	1	360	0	0	0	0	2	0	0	99.91	85.67	0.00	-	0.000
220	2040	0	0	304	0	0	0	0	6	0	0	99.61	87.03	-	-	0.000
221	2031	386	0	416	0	0	0	0	10	0	0	99.59	85.32	0.00	-	0.000
222	2403	0	0	281	0	0	0	0	80	0	0	96.78	89.53	-	-	0.000
223	2118	467	14	452	0	0	0	0	0	0	0	99.77	85.19	0.00	-	0.000
228	1683	361	0	513	0	0	0	0	8	1	0	99.56	79.94	0.00	-	0.000
230	2254	1	0	398	0	0	0	0	1	0	0	99.96	85.00	0.00	-	0.000
231	1569	2	0	215	0	0	0	0	0	0	0	100.00	87.96	0.00	-	0.000
232	1778	0	0	253	0	0	0	0	2	0	0	99.89	87.54	-	-	0.000
233	2188	822	11	510	0	0	0	0	49	9	0	98.12	85.56	0.00	-	0.000
234	2741	0	0	384	0	0	0	0	9	3	0	99.56	87.71	0.00	-	0.000
Sum	1578920	119832	150008	441532	0	0	0	0	0	0	9084	3180	374			
Gross												99.32	80.72	0.00	-	0.000
Average												99.37	81.13	0.00	-	0.000
Total QRS complexes:	1861398	Total VEBs:	123012													

Summary of results from 816 records

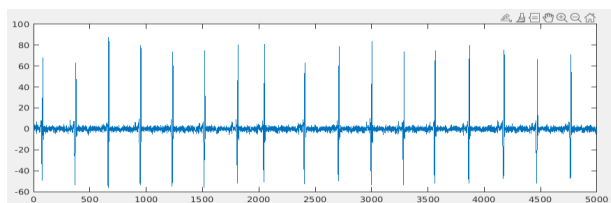
Slika 1: Slika prikazuje tabelo, ki vsebuje analizo posameznega ECG signala.

Občutljivost algoritma je v vseh primerih nad 94 procentov in v povprečju nad 99 procentov. Nekoliko nižja, a še vedno zadovoljiva je pozitivna predikcija, ki je v povprečju nad 80 procentov. Se pa s spreminjanjem parametrov rezultati lahko izboljšajo. Med številkami, ki sem jih preizkusila, sem z zgornjimi dobila najboljši rezultat.

Grafi 2, 3, 4 prikazujejo obdelavo vhodnih signalov skozi korake algoritma.



Slika 2: Graf prikazuje ECG signal.

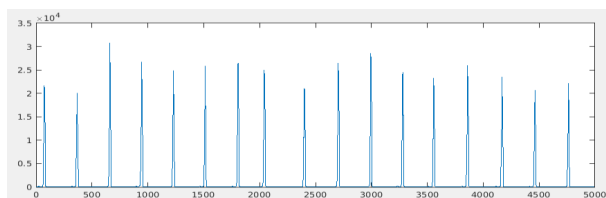


Slika 3: Graf prikazuje ECG signal po prvem koraku: High-Pass filter.

4 Možne izboljšave

Algoritem se seveda lahko izboljša s preizkušanjem različnih kombinacij vrednosti parametrov (dolžine filtra, velikosti premikajočih se oken).

Algoritem sem poskusila izboljšati na dva načina.



Slika 4: Graf prikazuje ECG signal po drugem koraku: Low-Pass filter.

- Z uporabo `detrend()`, ki zmanjša odstopanje iz osnovne linije (ang. Baseline Drift).
- Da bi se v primeru šuma le temu izognila, sem uporabila dva ECG posnetka srčnega utripa, ki se navezujeta na isti primer. V tretji fazi sem v primeru, da je bila maksimalna vrednost pri iskanju vrhov ECG posnetka manjša od praga (ang. threshold), tega primerjala z maksimalno vrednostjo drugega ECG posnetka. Na ta način sem preverila, ali je ECG posnetek vseboval šum, kar bi posledično lahko vodilo do izpusta srčnega utripa.

5 Zaključek

Na podlagi rezultatov lahko sklepamo, da je bila implementacija QRS algoritma uspešna. Pravilno zazna večino srčnih utripov, se pa s spreminjanjem parametrov rezultati in natančnost spreminjajo. Iz tega lahko sklepamo, da ima algoritem prostor za izboljšavo. V nadaljevanju bi bilo zanimivo izboljšati mojo predlagano izboljšavo. Torej, da bi algoritem vključeval analizo vseh ECG signalov, ki jih vsebuje posamičen primer.

Literatura

- [1] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.
- [2] "Pan-tompkins algorithm." [Online]. Available: https://en.wikipedia.org/wiki/Pan-Tompkins_algorithm
- [3] H. C. Chen and S. W. Chen, "A moving average based filtering system with its application to real-time qrs detection," in *Computers in Cardiology*, 2003, 2003, pp. 585–588.
- [4] "Matlab - mathworks - matlab simulink." [Online]. Available: <https://www.mathworks.com/products/matlab.html>