

Gručenje v hadronske curke

Matic Debeljak

29. avgust 2021

1 Naloga

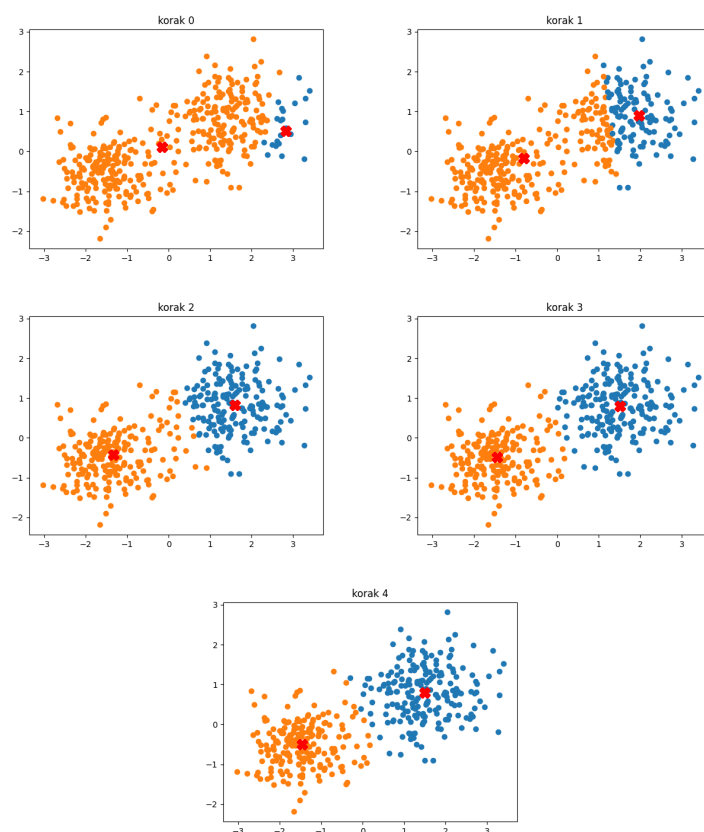
1. Implementiraj algoritem k-means, tako da slediš postopku, opisanem v sekciji 2 in ga preizkusi na priloženih Gaussovskih podatkih.
2. Poženi implementiran algoritem k-means na prvem dogodku (element 0) iz podatkov za $k = 2$ in $k = 10$. V obeh primerih predvidi, da curka z največjim p_T prihajata iz Higgsovega bozona in poskusi rekonstruirati maso Higgsovega bozona z uporabo invariante $m_h^2 = (p_1 + p_2)^2$, kjer sta p_1 in p_2 četverca izbranih curkov.
3. Preizkusi infrardečo varnost implementiranega algoritma k-means, tako da iz prvega dogodka postopoma odstranjuješ delce z najmanjšim p_T . Nariši odvisnost rekonstruirane mase Higgsovega bozona od števila delcev v dogodku.
4. Implementiraj algoritem k_t , tako da slediš postopku, opisanem v sekciji 2. Z uporabo parametra $R = 0.4$ ponovi test infrardeče varnosti.
5. Reproduciraj test infrardeče varnosti še z uporabo algoritma k_t iz knjižnice pyjet in primerjaj rezultate s svojim algoritmom.
6. Primerjaj časovno zahtevnost svojega algoritma z algoritmom iz pyjet, tako da ponovno postopoma odstranjuješ delce z najmanjšim p_T , in v vsakem koraku meriš čas izračuna.
7. Poženi algoritem iz knjižnice pyjet na vseh dogodkih, podanih v datoteki in nariši spekter invariantnih mas vsote curkov z največjim p_T .

2 Rezultati

2.1 K-means

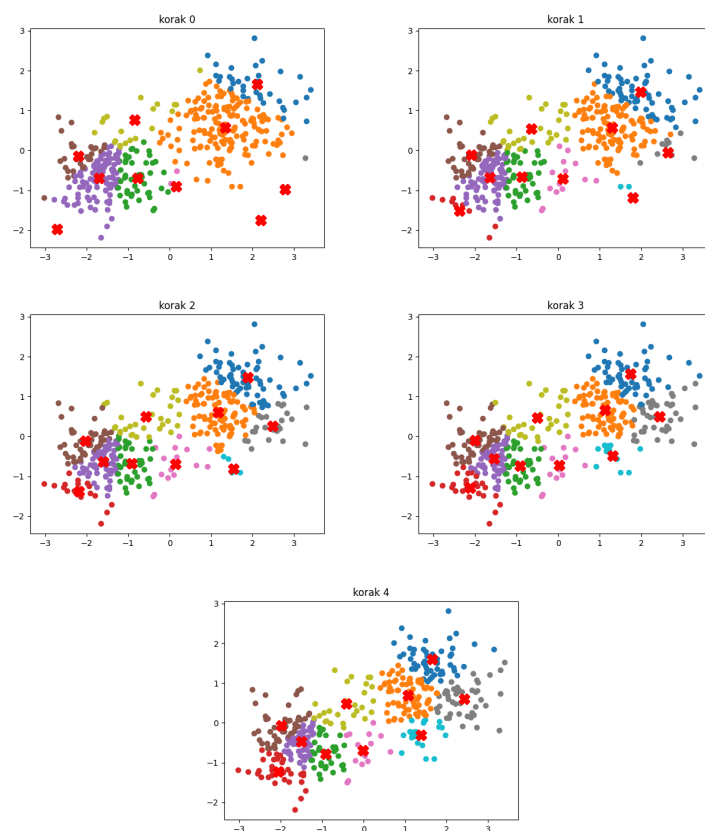
Opmoba: Pri nalogi sem uporabljal napačno enačbo za izračun mase Higgovega bozona iz gibalne količine dveh curkov. To sem ugotovil šele proti koncu in to tudi popravil (histogram na sliki 7 je pravilen) a vseh izračunov in grafov (predvsem zaradi časovne zahtevnosti) nisem popravljajl.

Pogljmo si najprej kako deluje ročno implementiran algoritem k-means. Za začetek sem ga pognal za gručenje na dve skupini (slika 1).



Slika 1: Prikaza delovanja mojega k-means algoritma za primer dveh gruč

Na sliki 1 lahko vidimo, da algoritem deluje pravilno in da konvergira precej hitro (kljub dokaj slabi izbiri začetnih centrov). Opazimo, da je delitev na dve gruči logična. Za primerjavo si lahko pogledamo še kako bi izgledalo če bi enake podatke poskisl razdeliti v 10 gruč (slika 2).

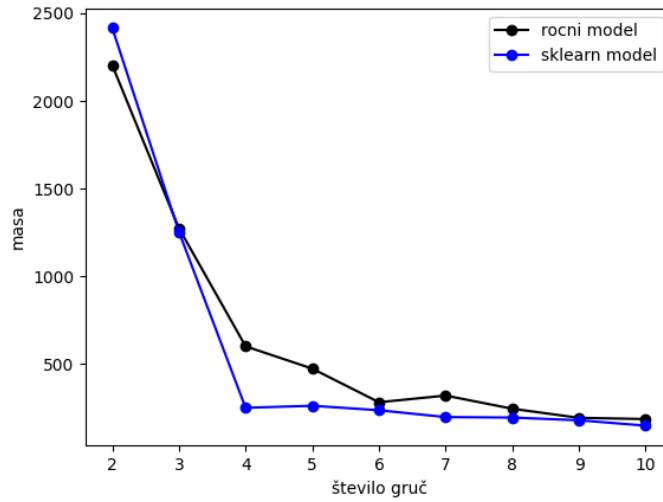


Slika 2: Prikaza delovanja mojega k-means algoritma za primer desetih gruĉ

Vidimo, da delitev na 10 gruĉ ni preveĉ logiĉna, zato program potrebuje tudi precej veĉ ĉasa da konvergira k rešitvi, ki pa je tudi odvisna od izbire zaĉetnih centrov, medtem ko tega pri deljenju na dve gruĉi nisem opazil.

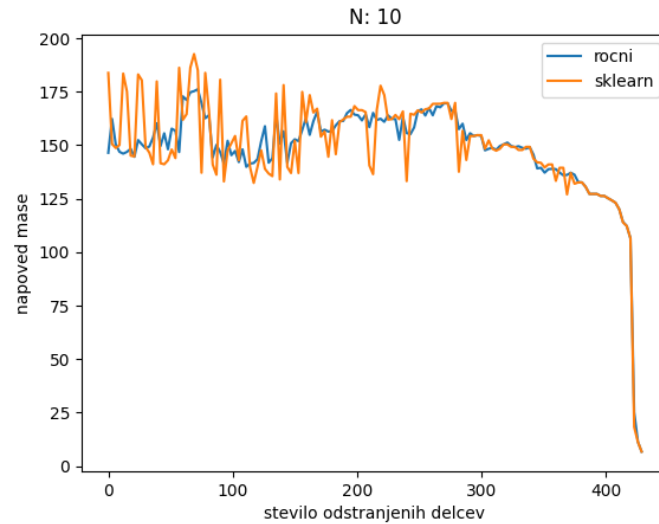
2.2 Hadronski curki

Poglejmo ali si lahko s pomočjo algoritma k-means pomagamo pri gručenju v hadronske curke in s tem pri napovedi mase Higgsovega bozona.



Slika 3: Napoved masa Higgsovega bozona v odvisnosti od števila gruč

Na sliki 3 opazimo, da se ob večjem ($N > 4$) številu gruč začnemo približevati dejanski masai Higgsovega bozona ($m_h = 125\text{Gev}$). A težava nastane ob preverjanju infrardeče varnost algoritma (slika 4). Da bi algoritem sprejeli kot dobro delujočega, bi se morala napovedana masa zelo malo spreminjati z odstranjevanjem delcev (do neke kritične točke).



Slika 4: Preverjanje infrardeče varnosti k-means algoritma

2.3 Algoritem k_t

V upanju, da bi rešili problem infrardeče varnosti sprogramiramo algoritem k_t . Poglejmo ali je ta uspešnejši.

Kot vidimo na sliki 5 se napoved k_t algoritma spreminaja zelo počasi z odvzemanjem delcov z najmanjšo gibalno količino, kar ustreza naši zahtevi po infrardeči varnosti.

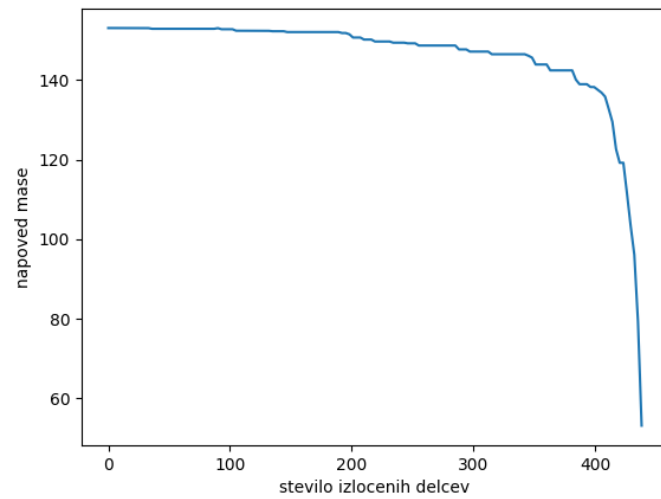
Ker pa je moj algoritem dokaj počasen, raje uporabljamo algoritem iz knjižnjice pyjet, ki deluje precej hitreje, primerjajmo časovni zahtevnosti obeh algoritmov (slika 6).

S pomočjo pyjetovega algoritma lahko zanaliziramo vse dobljene podatke in narišemo histogram napovedanih mas Higgsovega bozona (slika 7).

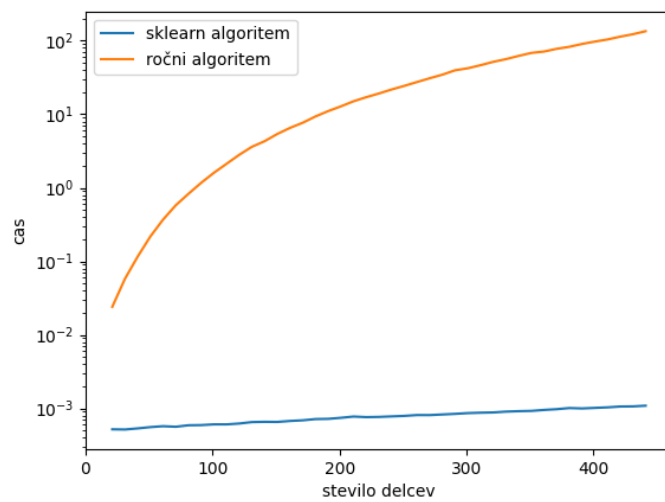
Opazimo vrh okol 125GeV, kar je tudi prava masa Higgsovega bozona.

3 Zaključek

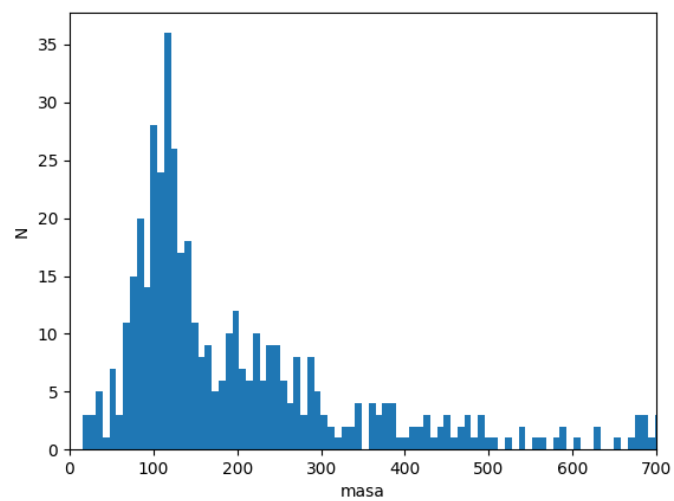
Pri tej domači nalogi sem se spoznal z dvema algoritmoma gručenja (k-means in k_t) in ju tudi sprogramiral na roke. S pomočjo teh sem nato določil maso Higgsovega bozona. Moja allgoritma sta delovala pravilno, a veliko počasneje kot enaki algoritmi, ki jih lahko najdemo v pythonovih knjižnjicah (sklearn in pyjet).



Slika 5: Preverjanje infrardeče varnosti k-means algoritma



Slika 6: Primerjava časovne zahtevnosti za ročno napisani algoritem in algoritem iz knjižnice pyjet



Slika 7: Prikaz napovedi mase Higgsovega bozona za 500 simluranih eksperimentov