UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
ODDELEK ZA FIZIKO

PRAKTIKUM STROJNEGA UČENJA V FIZIKI

6. naloga: Gručenje v hadronske curke

Žiga Šinigoj, 28222025

Ljubljana, januar 2024

1 Uvod

Pri trku delcev na trkalniku se delci za kratek čas razcepijo na kvarke, ki pa se hitro nazaj vežejo v hadrone. Hadrone zaznamo na detektorju. Če združimo curke hadronov lahko dobimo informacijo o nastalih kvarkih pred hadronizacijo. S pomočjo metod gručenja lahko učinkovito gručimo hadronske curke in na ta način preučujemo kvarke. V nalogi se bomo srečali z dvema algoritmoma za gručenje K-means in k_t. S pomočjo algoritmov bomo poskušali določiti maso Higgsovega bozona. Najprej testiramo implementacijo algoritma na znanih porazdelitvah iz datoteke *gauss.npy* in nato ocenimo iz podatkov v datoteki *h_bb_sorted.py*. Implementacija obeh algoritmov je opisana v navodilih naloge.

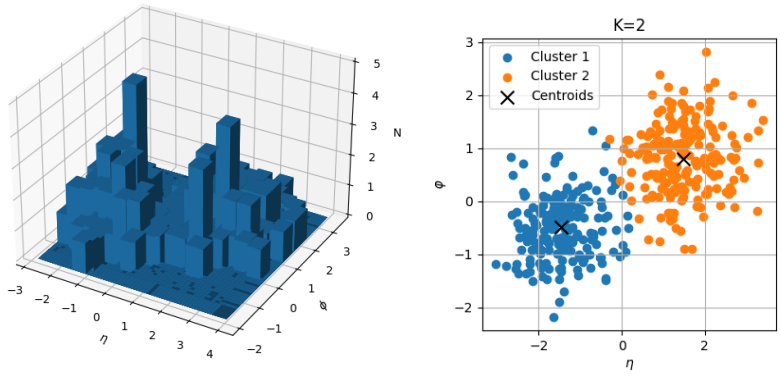
Curke opisujemo v prostoru koordinat (p_T, η, ϕ), kjer je p_T transverzalna komponenta gibalne količine, pravokotna na smer osi trka. Vektor gibalne količine v kartezičnem koordinatnem sistemu dobimo kot $\mathbf{p} = (p_x, p_y, p_z) = (p_T\cos(\phi), p_T\sin(\phi), p_T\sinh(\eta))$. Maso Higgsovega bozona izračunamo kot

$$m_H = \sqrt{(E_1 + E_2)^2 - (|\mathbf{p}_1 + \mathbf{p}_2|)^2}\ , \tag{1}$$

kjer uporabimo relativistični približek $E_{1,2} \approx |\mathbf{p}_{1,2}|$.

2 Gručenje s K-means

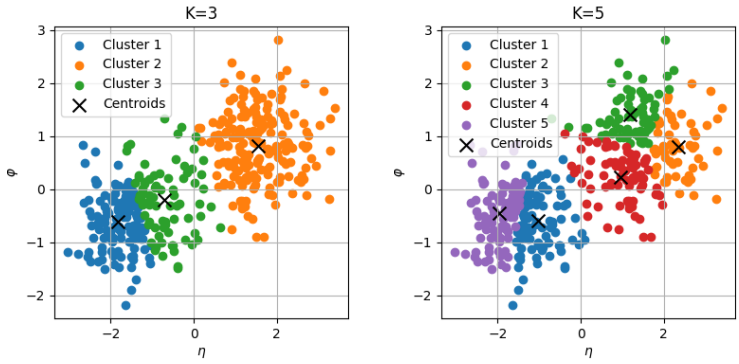
Pri algoritmu K-means izberemo začetne centroide in na podlagi razdalj do točk ločimo točke glede na pripadnost danemu centroidu. V naslednjem koraku povprečimo točke centroida in dobimo novo pozicijo centroida. Ta postopek ponavljamo do konvergence. Največja težava K-means algoritma je konvergenca, ki je zelo odvisna od začetne izbire centroidov. Testni set iz datoteke *gauss.npy* (slika 1a) je sestavljen iz dveh porazdelitev s središčem v (-1.5,-0.5) in (1.5,0.75) in odklonoma 0.3 in 0.5. Če sem za začetne točke izbral naključne točke iz množice $\phi \in [-3, 3]$ in $\eta \in [-3, 3]$ je bila v več kot polovici primerov konvergenca algoritma napačna. Rezultat lahko izboljšamo tako, da večkrat poženemo algoritem in povprečimo. Še boljše je, če za začetne točke centroidov vzamem kar meritve oz. točke iz dane množice. Primer gručenja prikazujejo slike 1b, 2. V danem primeru je smiselno iskati 2 gruči (K=2), vendar lahko z algoritmom iščemo K gruč. Kot dokaz da algoritem deluje lahko vidimo, da sta položaja centroidov za K=2 približno na mestih pričakovanih vrednosti porazdelitev. Konvergenco algoritma K-means sem definiral tako, da se algoritem



(a) (b)

Slika 1: a) Porazdelitvi iz datoteke gauss.npy . b) Gručenje s K-means v dve gruči.

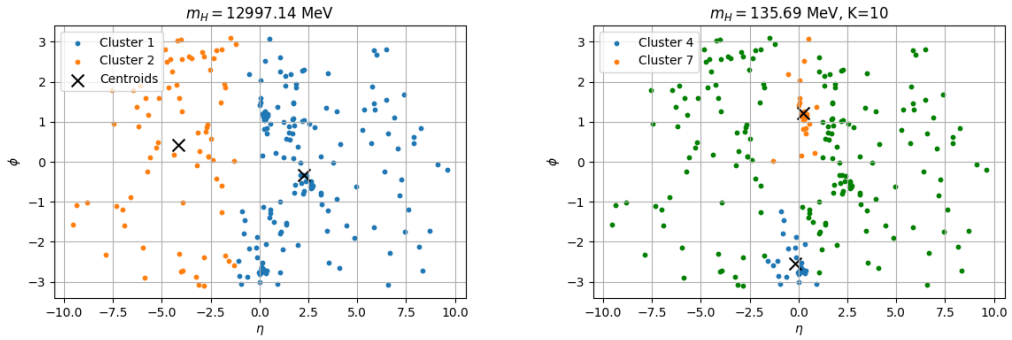
zaustavi, ko je druga norma razlik centroidov med dvema korakoma algoritma manjša kot vrednost ϵ ali pa ko je preseženo maksimalno število korakov. V drugem primeru ne nujno pride do konvergence, vendar pa je takih primerov bilo relativno malo. Če dane podatke iz *gauss.npy* želimo gručiti v več kot dve gruči, kar ni fizikalno smiselno, postajajo položaji centroidov vedno bolj odvisni od začetnih pogojev. Z večanjem K narašča število iteracij do konvergence algoritma, saj je več možnih konfiguracij za K grup.



(a) (b)

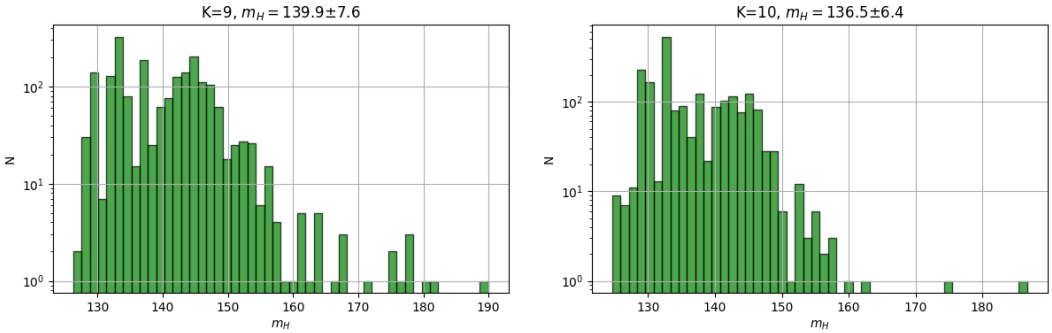
Slika 2: Gručenje s Kmeans. a) Tri gruče ($K = 3$). b) Pet gruč ($K = 5$).

Ob preverbi delovanja K-means, ga lahko sedaj uporabimo za določitev mase Higgsovega bozona. Gručenje algoritma za $K = 2$ in $K = 10$ prikazuje slika 3. Približek dveh gruč je zelo slab. Za $K > 2$ pa določimo maso Higgsa tako, da poiščemo gruči, ki imata največjo transverzalno giblano količino in izračunamo maso po enačbi 1. Pri $K = 10$ smo že precej bližje pravi masi. Da bi bolj natančno določil maso Higgsovega bozona sem naredil 2000 ponovitev gručenja in rezultate pri različnih K prikazal na sliki 4 in 5.



(a) (b)

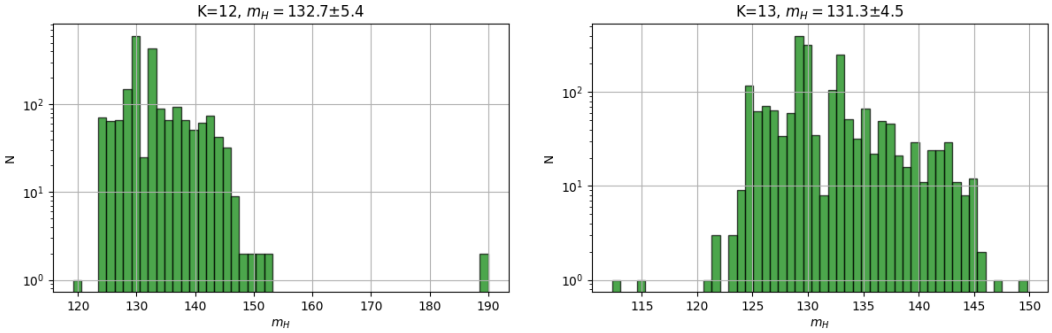
Slika 3: Gručenje podatkov za določitev mase Higgsovega bozona. Z modro in oranžno sta označeni gruči z največjo transverzalno gibalno količino. a) Gručenje na dve gruči ($K = 2$). b) Gručenje na 10 gruč ($K = 10$), irelevantne gruče so obarvane v zeleno.



Slika 4: Porazdelitev mas Higgsovega bozona po 2000 ponovitvah gručenja pri različnih K. Masa m_H je v enotah $[GeV]$. Masa Higgsovega bozona je določena kot povprečje, napaka pa kot standardni odklon porazdelitve.

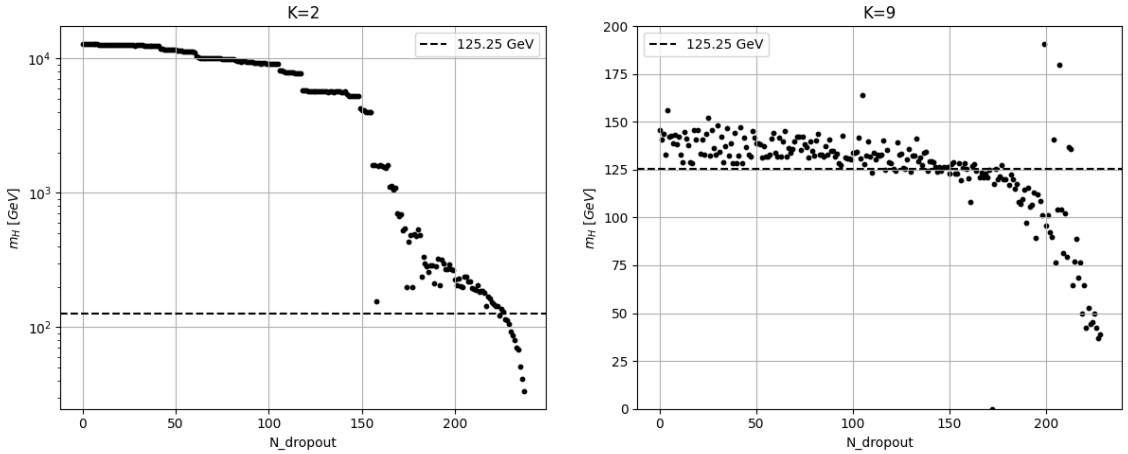
Ker nam da večji K boljšo oceni, se zdi smiselno povečati število gruč. Z večanjem števila gruč pa je

pogostejša divergenca K-means in so rezultati precej slabi. Maso Higgsovega bozona in odstopanje sem izračunal iz povprečja in standardnega odklona histogramov (slika 4 in 5). Najbližje pravi vrednosti je histogram s $K = 13$, ki pa se še vedno v okviru napake ne ujema s pravo vrednostjo $m_H = 125.25\ GeV$.



Slika 5: Porazdelitev mas Higgsovega bozona po 2000 ponovitvah gručenja pri različnih K. Masa m_H je v enotah $[GeV]$. Masa Higgsovega bozona je določena kot povprečje, napaka pa kot standardni odklon porazdelitve.

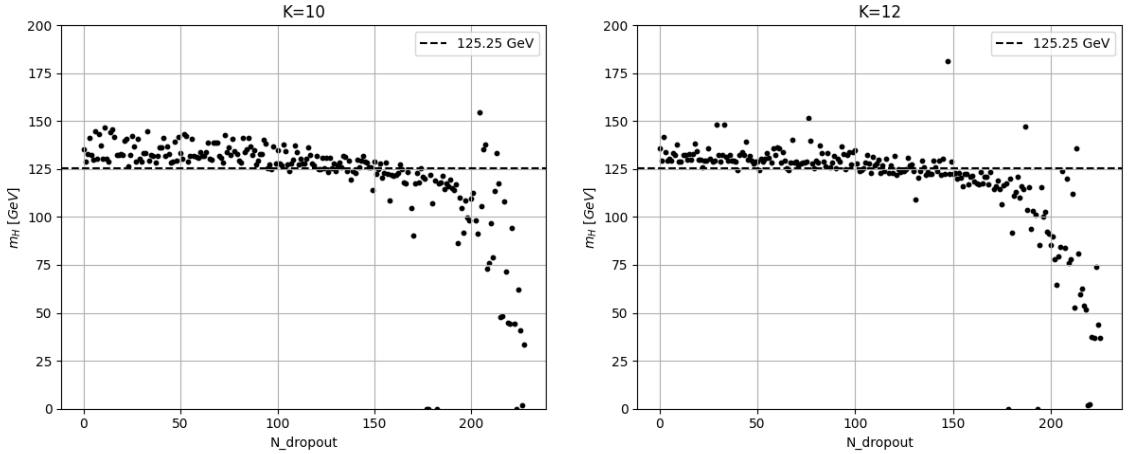
Željeni lastnosti pri algoritmu gručenja sta infrardeča varnosti in kolinearna varnost. Infrardečo varnost algoritma preverimo tako, da odstranjujemo delce z najmanjšo transverzalno gibalno količino p_T in opazujemo izračunano maso Higgsa. Dano odvisnost pri različnem številu gruč prikazujeta sliki 6 in 7. Pri $K = 2$ se odvisnost najhitreje spreminja, saj vsak odstranjeni delec prihaja iz gruče, ki ima največjo transverzalno gibalno količino. Pri večjih K je odvisnost manj strma na začetku. Zdi se, da se z večanjem K manjša raztros okrog vrednosti m_H. Nekaj raztrosa bo vedno, saj ima K-means po konstrukciji težave s konvergenco. Mogoče bi lahko rezultate izboljšal, če bi povprečil vrednosti pri danem $N_{dropout}$, za več zagonov algoritma.



(a) (b)

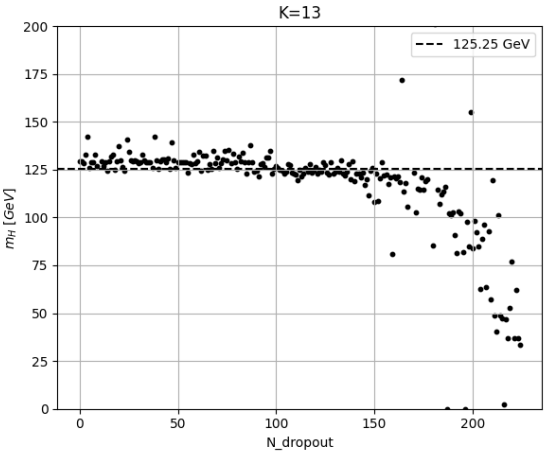
Slika 6: Test infrardeče varnosti algoritma K-means pri različnih K. $N_{dropout}$ označuje število zavrženih delcev z najmanjšo p_T.

Z večanjem K se tudi manjšajo velikosti grup. Posledično se zmanjšuje možnost, da je dana zavržena točka iz obeh relevantnih gruč in raztros točk okrog prave vrednosti je manjši.



(a)

(b)



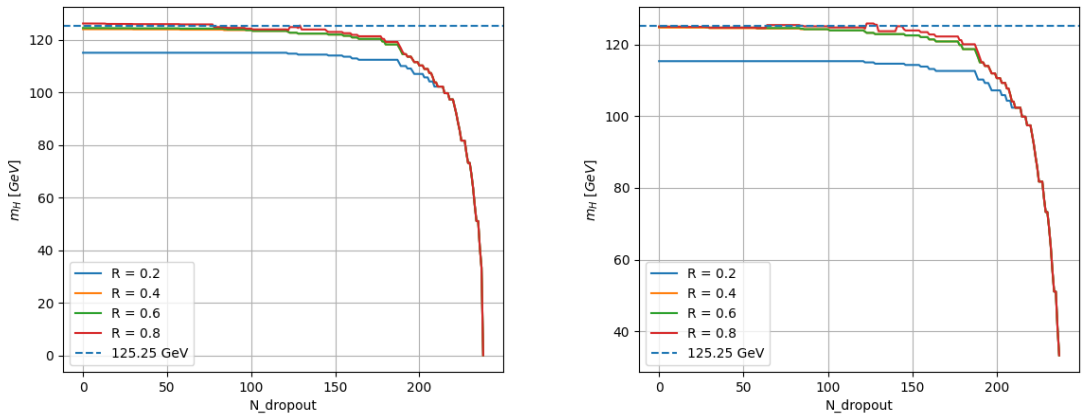
(c)

Slika 7: Test infrardeče varnosti algoritma K-means pri različnih K. $N_{dropout}$ označuje število zavrženih delcev z najmanjšo p_T.

3 Hierarhično gručenje - algoritem k_t

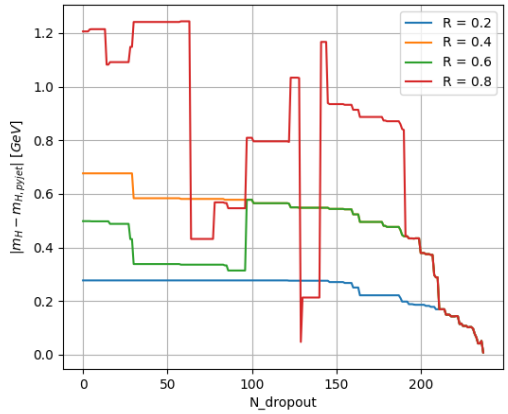
V namen navedenih željenih lastnosti se je razvil algoritem hierarhičnega gručenja. Bolj specifično bomo uporabili algoritem k_t, ki iz pseudocurkov naredi prave curke. Maso Higgsovega bozona, po delovanju z algoritmom na podatkih, dobimo iz delcev, ki imata največjo transverzalno gibalno količino. Delovanje implementiranega algoritma prikažem na test infrardeče varnosti (slika 8a za različne vrednosti parametra R. V primerjavi s K-means je veliko manj fluktuacij okrog Higgsove mase. Mogoče je malo presenetljivo, da algoritem ni stabilen za večje $N_{dropout}$, ampak ima koleno približno na enaki vrednosti kot K-means. Moja implementacija je precej počasnejša in neoptimizirana v primerjavi s Pyjet implementacijo algoritma k_t. Infrardečo varnost za Pyjet implementacijo prikazuje slika 8b. Odvisnost od parametra R ni velika za vrednosti okrog optimalne $R = 0.6$. Algoritem je bolj občutljiv navzdol od vrednosti $R = 0.6$. Hitrejša Pyjet implementacija da podobne rezultate, odstopanja po absolutni vrednosti prikazuje slika 8c. Odstopanja so reda 1 MeV, kar predstavlja okrog 1% odstopanje v najslabšem primeru. Zdi se tudi, da odstopanja algoritmov rastejo z večanjem parametra R.

Časovno odvisnost moje in Pyjet implemetacije v odvisnosti od števila zavrženih delcev prikazuje slika 8d. Razlika v hitrostih je zelo velika, na začetku celo več kot 4 velikostne rede. Vsaj dva velikostna reda sta zaradi implementacije C++ in Python. Približno dva velikostna reda pa potem morata biti skrita v optimizaciji algoritma. Pričakovano se časovna odvisnost manjša z manjšanjem števila delcev oz. večanjem $N_{dropout}$. Pyjet implementacija ima odvisnost $Nlog(N)$, moja implementacija pa približno N^3, kjer je N število protocurkov.

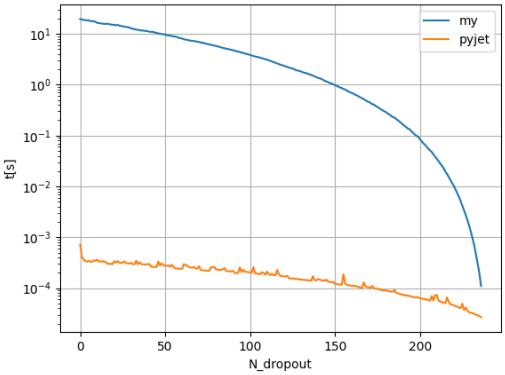


(a)

(b)



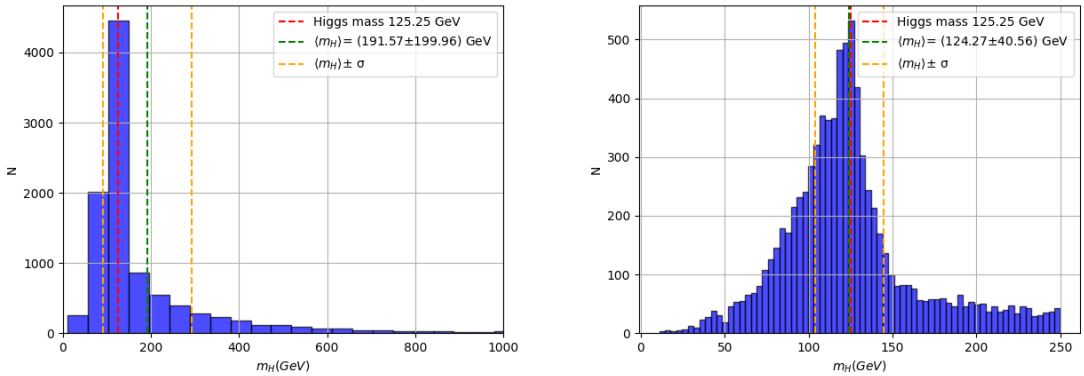
(c)



(d)

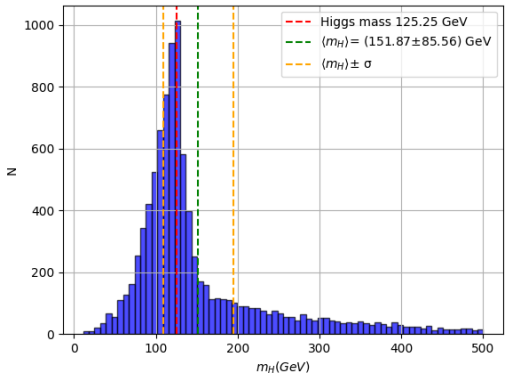
Slika 8: a) Test infrardeče varnosti primitivnega algoritma k_t pri različnih vrednostih parametra R. b) Test infrardeče varnosti algoritma k_t iz knjižnice *Pyjet* pri različnih vrednostih parametra R. c) Odstopanje Pyjet algoritma k_t od primitivnega algoritma k_t. d) Časovna zahtevnost primitivnega (my) in Pyjet k_t algoritma v odvisnost od števila zavrženih najnižjih transverzalnih količin.

Za bolj natančno določitev mase Higgsovega bozona poženemo algoritem na vseh dogodkih 10000 dogodkih. Za vsak dogodek poiščem delca z največjo p_T in izračunamo maso Higgsovega bozona. Porazdelitev po masah prikazuje slika 9. Če naivno izračunam povprečje in standardni odklon iz histograma so rezultati zelo slabi (slika 9a). Majhen delež mas je povsem zgrešenih (več tisoč MeV). Najverjetneje so to dogodki, ki ne opisujejo nastanka Higgsovega bozona. Za boljšo oceno sem izločil vse take dogodke in vzel simetrični interval glede na maso m_H [0 GeV, 250 GeV]. V tem primeru dobimo dobro oceno za maso Higgsa (slika 9b), a je nedoločenost še vedno zelo velika. V tem primeru je na danem intervalu 76% vseh mas. Ker se mi je zdelo, da vseeno izpustim preveč meritev, sem razširil interval na [0 GeV, 500 GeV] (slika 9c). Rezultati so slabši, ampak je delež podatkov v tem intervalu 94%. Spomnil sem se prve naloge iz PSUF, kjer smo imeli prav tako opravka z Higgsovim bozonom. Da bi dobil boljšo oceno sem prilagajal Crystal Ball funkcijo z optimalnimi parametri (PSUF 1.naloga) na dano porazdelitev in dobil najboljšo oceno za maso Higgsovega bozona $m_H = (116 \pm 25)\ GeV$.

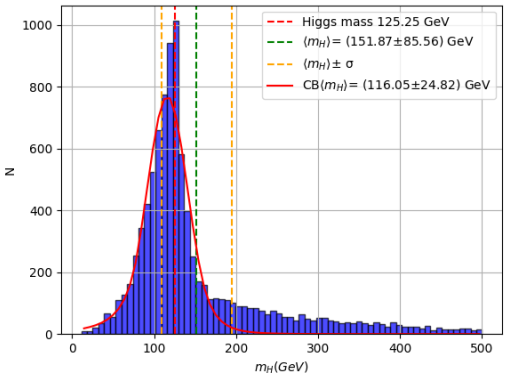


(a)

(b)



(c)



(d)

Slika 9: Porazdelitev invariantnih mas Higgsovega bozona. a) Povprečje in σ izračunana na celotni porazdelitvi. b) Povprečje in σ izračunana na simetričnem intervalu okrog mase Higgsovega bozona $m_H \in [0\ \text{GeV}, 250\ \text{GeV}]$. Delež vseh podatkov na tem intervalu je 76 %. c) Povprečje in σ izračunana na intervalu okrog mase Higgsovega bozona $m_H \in [0\ \text{GeV}, 500\ \text{GeV}]$. Delež vseh podatkov na tem intervalu je 94 %. d) Enako kot c) z dodanim fitom Crystal Ball funkcije ($CB\langle m_H \rangle$).

4 Zaključek

V nalogi smo preučevali algoritme gručenja, K-means in k_t, ter njihovo uporabo pri določanju mase Higgsovega bozona. K-means je, kljub svoji učinkovitosti, občutljiv na začetne pogoje in ima lahko težave s konvergenco. Nasprotno, algoritem k_t ponuja bolj stabilno rešitev, a je računsko bolj zahteven.