Универзитет у Београду – Електротехнички факултет

Катедра за сигнали и системе

Дипломски рад

Компаративна анализа метода за кластеризацију података

Кандидат
Алекса Јовановић, бр. индекса 2019/0146

Ментор
др Жељко Ђуровић, редовни професор

Београд, *септембар* 2023. године

САДРЖАЈ

1 Увод

У данашње време подаци имају огроман значај јер представљају кључан ресурс за развој и унапређење инжењерских решења која ће проширити домен проблема који рачунари могу да реше у областима готово свих људских делатности и на тај начин подигну продуктивност читавог друштва. Први кључан тренутак у историји који је имао сличан револуционаран ефекат био је проналазак парне машине, а касније и осталих мотора који су успели да разне форме енергије претворе у механички рад који је заменио рад људских мишића. Људи су морали да управљају "механичким мишићима" све до 20. века када се догодила нова револуција појавом рачунара. Рачунари су, попут човека, тада били способни да извршавају алгоритме засноване на логичком закључивању и да кроз те алгоритме управљају машинама. Такав развој догађаја довео је до аутоматизације многих послова и омогућио људима да се баве другим делатностима. Међутим, у то време рачунари нису имали кључну могућност људског мозга, а то је да уче на основу претходних искустава. Људи на основу својих претходних акција и опсервација могу да извуку закључке који их у будућности чине много способнијим. Људски експерти у било којој области се у многоме вреднују по томе колико искуства имају, односно кроз које су све практичне ситуације прошли зато што учећи из претходних искустава људи постају опремљенији за решавање нових проблема из своје струке као и ефикасније решавање већ постојећих проблема. Нагли развој рачунара и њихових капацитета за чување и обраду података омоућио им је да имитирају процес учења из претходних искустава који се до скоро одвијајо искључиво код биолошких система.

Податак представља опсервацију која може бити физичка величина, нека чињеница или чак било каква вредност од интереса која ја нумерички изражена и сачувана најчешће у дигиталној форми у меморији рачунара, што се може упоредити са сећањима односно личним искуством које човек држи у својој глави. Технике обраде података дају рачунару могућност да из сирових података извлачи информације, изводи закључке и на неки начин учи, баш као и ми. Подаци дакле представљају предуслов за развој интелигнетнијих и далеко способнијих рачунара који ће довести до олакшање људског живота и бољег животног стандарда. Имајући све ово у виду јасно је због чега су подаци огроман ресурс којим се већ данас тругује. Неке од највећих компанија на свету из области рачунарства и информационих технологија зарађују велике своте новца продавајући податке које прикупљају од свих нас.

У том светлу овај рад се бави техникама за кластеризацију и класификацију података као начинима за екстракцију информација из доступних података у сврху груписања и категорисања. Ове технике на рачунару имитрају комплексне когнитивне процесе које људи свакодневно обављају, а то су препознавања облика (патерна) и доношења одлука.

Главна тема рада јесте кластеризација податка која служи да међу сировим подацима пронађе везе и групише их на паметан начин у скупове које називамо кластери. У улазном скупу података ми не знамо који подаци припадају којој групи, а често не мора бити познато ни колико заправо група података треба издвојити. Постоји велики број метода за кластеризацију, а циљ овог рада јесте поређење перформанси најчешће коришћених метода у зависности од карактеристика улазног скупа података. Параметри који карактеришу скуп улазних података, а биће од интереса јесу да ли је познат број кластера, ако јесте колики је он, тип расподеле података итд.

Поред кластеризације детаљније ће бити описане и методе за класификацију података. Класификација података је сличан проблем проблему кластеризације осим што нам је на располагању и предзнање који улазни подаци припадају којој класи. Циљ класификације је да на основу улазног скупа података пронађе дискриминациону функцију која ће представљати границу између различитих класа података да би се у будућности на основу те функције нови подаци (за које не знамо којој класи припадају) могли класификовати. Многе методе кластеризације су засноване управо на методама класификације података. Из тог разлога је добро прво разумети како се обавља класификација, а затим се бавити проблемом кластеризације.

1.1 Обележје (енг. feature)

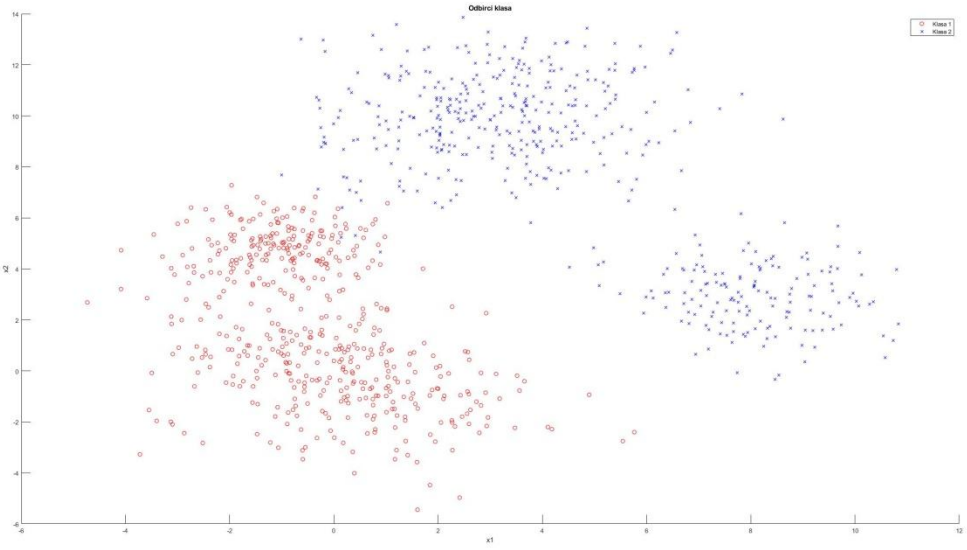
Обележје је кључан појам који се користи при препознавању облика тј. класификацији и кластеризацији. Обелжје представља податак који карактерише премете или појаве из реалног света. Кроз обележја заправо апстрахујемо реалне појаве од интереса тако што њихове главне карактеристике нумрички изарзимо и сместимо у вектор обележја. Вектор обележја се још назива и облик. На тај начин преводимо оно што нас интересује у матеамтички домен те га можемо обрађивати унутар рачунара користећи математички апарат. Технике кластеризације и класификације заправо прихватају податке у форми великог броја вектора обележја где сваки вектор представља један одбирак. Поступак екстракције обележја је веома важан јер класификациони алгоритми оптимизују класификацију на основу датог скупа обележја, па уколико обележја нису довољно имформативна чак и оптимлани

класификатор неће моћи да постигне високу тачност. Одабир обележја је на инжењеру који пројектује класификатор, међутим треба нагласити да постоје методе у области машинског учења попут deep learning-а где се неуралне мрежа тренира да сама препознаје битна обележја из изворних података као што су слике, видео/аудио записи и томе слично. Очекивано за коришћење deep learning-а је потребно много више процесорске снаге него када човек селектује обележја.

Посматрање и филтрирање битних карактеристика субјекта нпр. код препознавања лица јесте нешто што људски мозак подсвесно такође ради. Тако када рецимо покушавамо да проценимо да ли нам је нека особа позната посматрамо њену/његову боју косе, боју очију, облик лица и томе слично.

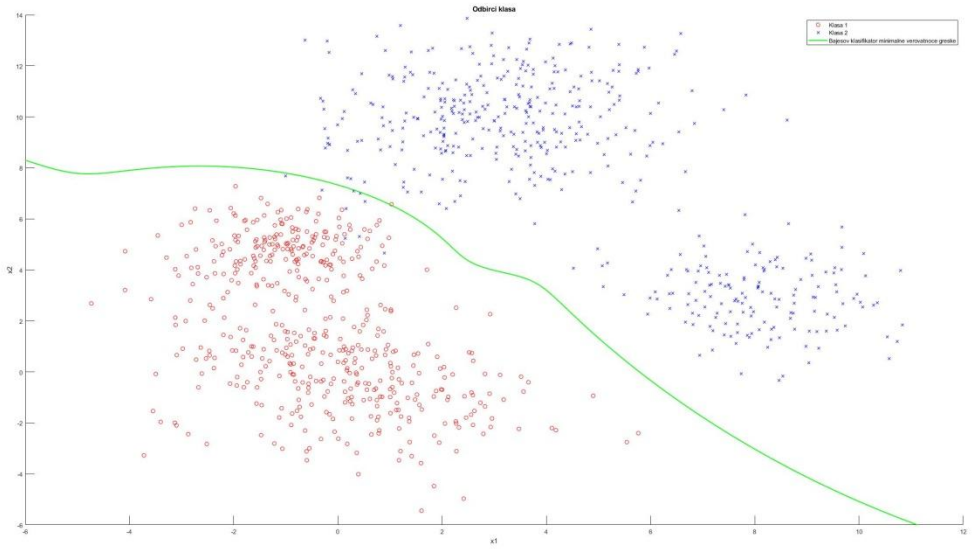
Природа нумеричких података које добијамо екстракцијом обележја је стохастичка. Било које обележје представља случајну променљиву која има своју функцију расподеле, а облик случајни вектор такође са својом мултидимензионланом функцијом расподеле. Јасно је да ће облици који припадају једној класи имати другачије статистичке карактеристике у односу на облике који потичу из неке друге класе. Управо на овој основи различитих статистичких карактеристика облика из различитх класа пројектујемо систем који ће за непознати облик проценити којој класи припада тј. за коју класу је највећа вероватноћа да њој припада.

Након што се издвоје битна обележја и добијемо улазни сет података можемо те податке исцртати у простору обелеђжа као тачке као што је то приказано на слици 1.



Слика 1. Приказ синтетички генерисаних облика у дводимензионом простору обележја подељених на 2 класе. Подаци су генерисани уз помоћ Matlab (TheMathworks, Natick, USA) функције "mvrnd.m".

Проблем пројектовања класификатора и кластеризације своди се на поделу читавог простора обележја у неколико региона. Функција или више њих које ограничавају ове регионе називају се дискримнационе функције. Подела простора обележја илустрована је на слици 2.



Слика 2. Приказ испројектоване класификационе линије која дели простор обележја у део који додељујемо првој класи и део који додељујемо другој класи.

Напомена: У раду су коришћене многе математичке ознаке, па су оне пописане и објашњене у ПРИЛОГУ А. Такође, већина извођења заснива се на теоремама и формулама из вероватноће и статистике чији се преглед може наћи у литератури [1].

2 Технике за статистичку класификацију облика

Као што је беħ наглашено, најпре ћемо се бавити техникама за класификацију облика заснованим на статистичкој анализи где су нам доступни одбирци за које је познато којој класи припадају. На основу тих информација треба да пронађемо оптимални начин за сврставање нових одбирака за које не знамо из које класе потичу. Најједноставнији случај класификације је када облици потичу из само две класе и доста метода је засновано на тој претпоставци. Међутим, често је могуће уопштити приниципе који се корсите за класификацију у 2 класе на више класа што ће и бити показано.

Методе класификације могу се поделити у 3 групе:

1. Тестирање хипотеза
2. Методе параметарксе класификације
3. Методе непараметарске класификације

Биће обрађено по неколико метода из сваке наведене групе.

2.1 Тестирање хипотеза

Технике под називом тестирање хипотеза претпостављају да нам је статистичка расподела облика из појединх класа у потпуности позната за све могуће вредности обележја. Другим речима неопходно је да знамо условну фукнцију густине вероватноће облика за сваку класу $f_i(X)$. Такође, морамо знати колико често се јављају облици из сваке класе тј. априорну верватноћу појаве облика P_i. Познавање ових параметара у пракси готово никада није случај, али се најчешће уз неке претпоставке они могу естимирати на основу улазних података. Најједносатвнији начин јесте претпоставити да је функција густине вероватноће Гаусовка (што је у значајном броју случајева оправдано јер се Гаусовска нормална расподела најчешће јавља у природи). Нормална расподела има само два параметра средњу вредност и коваријациону матрицу који се лако могу естимирати, као што се лако може естимирати и априорна вероватноћа појаве облика и то на следећи начин:

$$\widehat{M}_i = \frac{1}{N_i}\sum_{k=0}^{N_i} X_k \ , X_k \ \in \omega_i$$

$$\hat{\Sigma}_i = \frac{1}{N_i - 1}\sum_{k=0}^{N_i} (X_k - \ \widehat{M}_i)(X_k - \ \widehat{M}_i)^T \ , X_k \ \in \omega_i$$

$$\hat{P}_i = \frac{N_i}{N}$$

$$\hat{f}_i(X) \sim N\left(\widehat{M}_i, \hat{\Sigma}_i\right) = (2\pi)^{-n/2}|\hat{\Sigma}_i|^{-1/2}\exp\left[-\frac{1}{2}\left(\boldsymbol{X} - \widehat{M}_i\right)^{\boldsymbol{T}}\hat{\Sigma}_i^{\ -\mathbf{1}}\left(\boldsymbol{X} - \widehat{M}_i\right)\right]$$

Наравно ово није једини начин за естимацију функције густине вероватноће, али служи да илуструје да технике тестирања хипотеза имају употребну вредност и нису чисто теоријске. Заправо читава трећа група метода, метода непараметарске класификације, је развијена баш са циљем да на основу улазних одбирака што боље процени функцију гуситне вероватноће а онда користи правила закључивања изведена при тестирању хипотеза.

2.1.1 Бајесов класификатор минималне грешке

Бајесов класификатор је оптимални класификатор у смислу да минимизује укупан број грешака класификације, тачније вероватноћа погрешне класификације нових одбирака је минимална. Бајесов класификатор се пројектује за случај када имамо две класе одбирака, али се уопштава и на случај више класа.

Идеја овог класификатора је веома једноставна и каже да на основу расподеле одбирака и њихових априорних вероватноћа појављивања треба проценити вероватноћу да непознати одбирак X припада првој класи и вероватноћу да припада другој класи. Ако је вероватноћа да X припада првој класи већа X треба сместити у прву класу, а у супротном ако је процењена вероватноћа да X припада другој класи већа, X треба сместити у другу класу.

$$q_1(X) > q_2(X) \Rightarrow X \ \in \ \omega_1$$

$$q_1(X) < q_2(X) \Rightarrow X \ \in \ \omega_2$$

Вероватноће $q_1(X)$ и $q_2(X)$ могу се срачунати преко Бајесове формуле као:

$$q_i(X) = \frac{P_i f_i(X)}{f(X)}$$

Па се израз класификације своди на:

$$\frac{f_1(X)}{f_2(X)} > \frac{P_2}{P_1} \Rightarrow X \in \omega_1 \text{ ,у супротном } X \in \omega_2$$

Да је Бајесов класификатор оптималан интуитивно има смисла, јер поредимо саме вероватноће да неки одбирак припада једној односно другој класи. Оптималност се може и формално доказати. За детаље доказа погледати литературу [2].

2.1.2 Бајесово правило одлучивања минималне цене

Бајесов класификатор који је прво разматран јесте оптималан по питању укупне грешке, међутим у пракси је често битније да одбирци једне класе буду добро класификовани чак и по цену да се одбирци друге класе класификују мање прецизно. Вероватноћа да објекат из прве класе буде погрешно класификован назива се врватноћа грешке првог типа и обележава се са ε_1, док је аналагно вероватноћа грешке другог типа ε_2 вероватноћа да погрешно буде класификован одбирак из друге класе. Да би се постигла боља тачност класификације за једну од класа треба модификовати претходни класификатор увођењем параметар који прдстављају цене. Са c_{ij} обележава се цена одлуке да се одбирак X придружи класи ω_i када он заправо припада класи ω_j. За сваки одбирак сада нећемо доносити одлуку тако да је најмања шанса да погрешимо већ ћемо гледати да платимо најмању укупну цену одлуке. Цена одлуке смештања одбирка у класу ω_i се формулише на следећи начин:

$$r_i(X) = c_{i1}q_1(X) + c_{i2}q_2(X)$$

Одлуку доносимо поређењем цене смештања одбирка у прву и другу класу.

$$r_1(X) < r_2(X) \Rightarrow X \in \omega_1$$

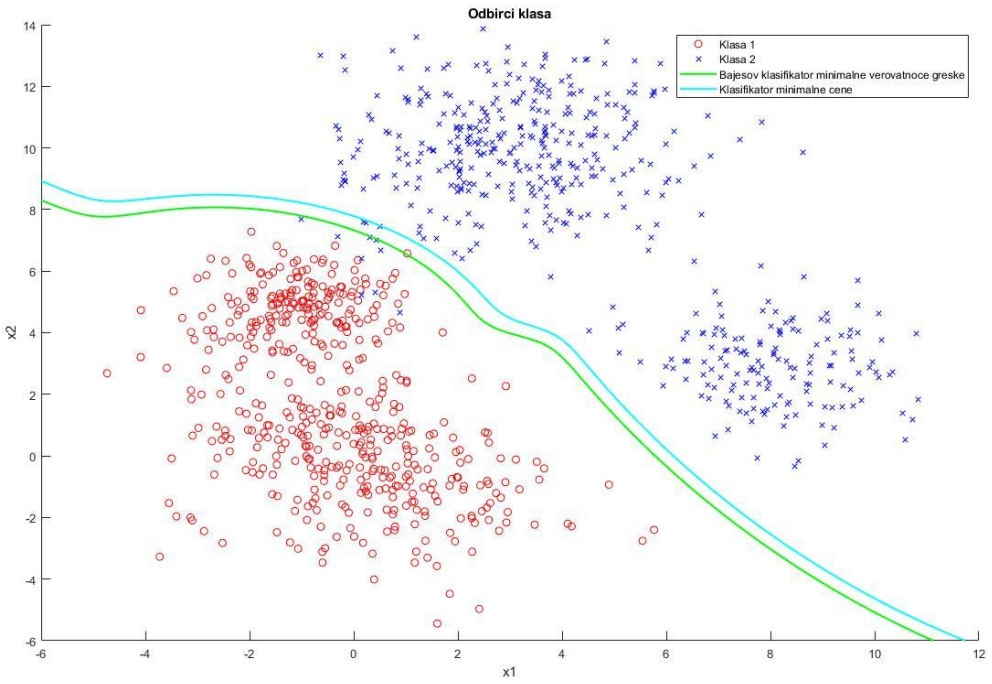
$$r_1(X) > r_2(X) \Rightarrow X \in \omega_2$$

Увртавањем израза за $q_i(X)$ и аритметичким сређивањем (детаљи су доступни у литератур [2]) правило одлучивање се може изразити као:

$$\frac{f_1(X)}{f_2(X)} > \frac{(c_{12} - c_{22})P_2}{(c_{21} - c_{11})P_1} \Rightarrow X \in \omega_1$$

$$\frac{f_1(X)}{f_2(X)} < \frac{(c_{12} - c_{22})P_2}{(c_{21} - c_{11})P_1} \Rightarrow X \in \omega_2$$

Због општости изрази се изводе са ценама c_{11} и c_{22}, али се у пракси ове цене углавном постављају на нулу јер нема разлога пенализовати исправно донету одлуку.



Слика 3. Поређење Бајесовог класификатора минималне грешке (зелена линија) и класификатора минималне цене (светло плава линија).

На слици 3 видимо поређење класификатора минималне грешке и класификатора минималне цене. Узето је да је c_{21} пет пута већа од цене c_{12} и на тај начин је од класификатора тражено да прецизније класификује одбирке прве класе. Самим тим се класификациона линија померила ка одбирцима друге класе.

2.1.3 Neyman-Pearson-ов тест

Neyman-Pearson-ов тест се користи у случају да је од интереса једну од грешака класификације нпр. ε_2 поставити на тачно одређену вредност. ε_2 ћемо дакле фиксирати на жељену вредност ε_0, а онда је од интереса да се под тим условом ε_1 минимизује. Да би се то урадило формира се критеријумска функција r која се условно минимизује уз помоћ Лагранжевог мултипликатора:

$$r = \varepsilon_1 + \mu(\varepsilon_2 - \varepsilon_0)$$

Резултат који се добије овим поступком је:

$$\frac{f_1(X)}{f_2(X)} > \mu \Rightarrow X \in \omega_1$$

$$\frac{f_1(X)}{f_2(X)} < \mu \Rightarrow X \in \omega_2$$

Испоставља се да је праг одлучивања μ веома незгодно срачунати јер је интегралном везом спрегнут са функцијом густине вероватноће облика и жељном вероватноћом грешке ε_0 што ограничава употребљивост овог алгоритма. Једноставнији начин за конторлисање ε_2 био би да се користи Бајесово правило одлучивања минималне цене, али тако што се кроз више итерација модификује однос цена c_{12} и c_{21} све док не добијамемо задовољавајуће ε_2.

2.1.3 Минимакс тест

Уколико нису познате априорне вероватноће појаве облика из различитих класа или их не можемо довољно прецизно одредити опасно је користити Бајесова правила одлучивања. Уколико се класификатор пројектује за неке вредности P_1 и P_2, али се оне у реалности знатно разликују перформансе класификатора могу нагло да деградирају. Идеја минимакс теста је да се ова деградација перформаси избегне тако што се за P_1 и P_2 узму најнеповољније вредности. Тачније, за P_1 и P_2 се претпостави да имају вредност за коју је грешка класификације оптималног Бајесовог класификатора максимална. Испоставља се да ће на тај начин, чак и са потпуним одступањем правих вредности P_1 и P_2, веровтноћа грешке класификатора остати константна. Ова идеја има интуитивног смисла јер, као у животу ако се припремимо за најгори могући случај, шта год да се деси не можемо проћи горе од онога на шта смо се већ припремили. Тест се зове минимакс, јер тражимо максимимум минималне (оптималне) грешке класификације у зависности од парамеатара P_1 и P_2 (прецизиније ове вероватноће су спрегнуте релацијом $P_2 + P_1 = 1$ па се само једна од њих бира као параметар). Детаљно извођење је превише опширно за овај рад, али је доступно у литератури [2].

2.1.4 Уопштавање Бајесових тестова на проблем више од две класе

Када је потребно препознавати одбирке из више од две класе, Бајесов тест мининалне грешке се може уопштити тако што пронађемо вероватноћу да одбирак припада свакој класи $q_i(X)$, $i = 1, \ldots, L$. Одлуку да $X \in \omega_i$ доносимо за $i = \max_i q_i(X)$.

Ако желимо да у правило одлучивања уфакторишемо и цене класификације дефинасаћемо очекивану цену одлуке за све класе:

$$r_i(X) = \sum_{j=0}^{L} c_{ij}\, q_j(X)\,, \qquad i = 1, \ldots, L$$

А доносимо одлуку да одбирак X припада класи i за коју је $r_i(X)$ минимално.

2.1.5 Тест једне хипотезе

Код свих метода класификације које смо до сад разматрали имали смо L класа које су тачно дефинисане. Међутим, нису нам све класе подједнако битне, а врло често имамо само једну од класа коју је битно идентификовати док су нам остале у потпуности неважне. Пример овог случају ја када уз помоћ радарског система покушавамо да детектујемо рецимо путнички авион. Радар може да детектује разне сигнале који потичу од птица, облака, метеоролошких балона и сличних летелица али нам је све то неважно. Једино је битно видети да ли је непознати објекат авион, а ако он то није шта је то заправо нас не интересује. То ћемо урадити тако што непознати облик покушавамо да доведемо у везу са облицима за које знамо да припадају класи од интереса. Прва идеја која нам пада на памет јесте да за непознати облик X гледамо његово Еуклидско растојање или квадрат Еуклидског растојања (ради лакшег и бржег израчунавања) од средње вредности облика из класе од интереса.

$$\|X - M\|^2 = (X - M)^T (X - M)$$

Онда то растојање можемо поредити са неким прагом који бирамо тако да имамо задовољавајућу вредност грешке. Посматрање Еуклидског растојања јесте најједноставније, али боље перформансе класификације можемо добити ако посматрамо статистичко растојање облика X од M. Статистичко растојање узима у обзир и коваријациону матрицу познатих облика из класе од интереса. На тај начин се обраћа пажња и на облик расподеле, који неће увек бити симетричан и што је случај када постоје корелације између компоненти вектора обележја. Статистичко растојање се дефинише овако:

$$d^2(X) = (X - M)^T \Sigma^{-1} (X - M)$$

Као и код Еуклидског статистичко растојање се онда пореди са прагом који се може одредити експериментално тако да производи жељену вероватноћу грешке.

2.1.6 Опција одбацивања

Опција одбацивања је метод који дорађује основи Бајесов класификатор минималне грешке. Код класификатора минималне грешке када су непознати облици близу класификационе линије, матеамтички изражено $q_1(X) \cong q_2(X)$, постоји велика вероватноћа (око 50%) да ћемо погрешити коју год одлуку да донесемо. За неке практичне случајеве ово уопште није прихватљиво и исплативије је уопште не доносити одлуку него погрешити. Разлог за то је што облике које је тешко класификовати можемо посебно посматрати као један подскуп проблема и за њих пројектовати неки други начин класификације који ће много мање грешити и на тај начин свеукупно добити бољи класификатор. Ово ћемо имплементирати тако што у Бајесово правило одлучивање додајемо додатни праг t. Избор параметра t је на пројектанту и представља маскималан ризик (вероватноћу погрешне класификације) који је могуће толерисати. Узимајући ову идеју у обзир правило одлучивања добија облик:

$$r_i(X) = \min\big(q_1(X), q_2(X)\big) > t \Rightarrow \text{ одбирак } X \text{ се одбацује}$$

$$q_1(X) > q_2(X) \ \wedge \ q_2(X) < t \ \Rightarrow X \ \in \ \omega_1$$

$$q_1(X) < q_2(X) \ \wedge \ q_1(X) < t \ \Rightarrow X \ \in \ \omega_2$$

Наравно у рачунање ризика одлуке се могу уврстити и цене, па онда t постаје праг очекиване цене одлуке коју смо спремни да платимо.

2.1.7 Секвенцијално тестирање хипотеза и Wald-ове секвенцијални тест

До сада смо кроз претходне методе подразумевали да сваки одбирак представљаља независан примерак одређене класе што не мора увек карактерисати класификационе проблеме. Ако рецимо у војном окружењу имамо радарски систем за препознавање претећих ракета, тај систем не мора да само једном изврши скенирање непознатог објекта и на основу мерења процени да ли је он опасан или не. Могуће је у више еквидистантних временских тренутака скенирати исти објекат и на основу свих колективних мерења класификовати објекат као претећи или непретећи. У ту сврху развијени су секвенцијални тестови који сумирају информације из свих мерења у реалном времену како оне пристижу. Што више одбирака имамо на располагању моћићемо да извршимо класификацију са мањом грешком, међутим код овог типа тестова је углавном од интереса донети одлуку што пре. Инжењери који се баве овим

типом проблема морају да праве компромис између тачности и брзине доношења одлуке.

Пристигле одбирке који представљају предмет класификације обележићемо са $X_1, X_2, \dots, X_m$ и претпоставићемо да су одбирци међусобно некорелисани. Информације пристигле новим мерењима анализирају се уз помоћ здружене функције густине вероватоће одбирака из прве и друге класе и то посматрајући њихов количник (количник веродостојности, означн са l_m) или негативни логаритам количника (дискримнациона функција, означена са s_m):

$$l_m = \frac{f_1(X_1, \dots, X_m)}{f_2(X_1, \dots, X_m)}$$

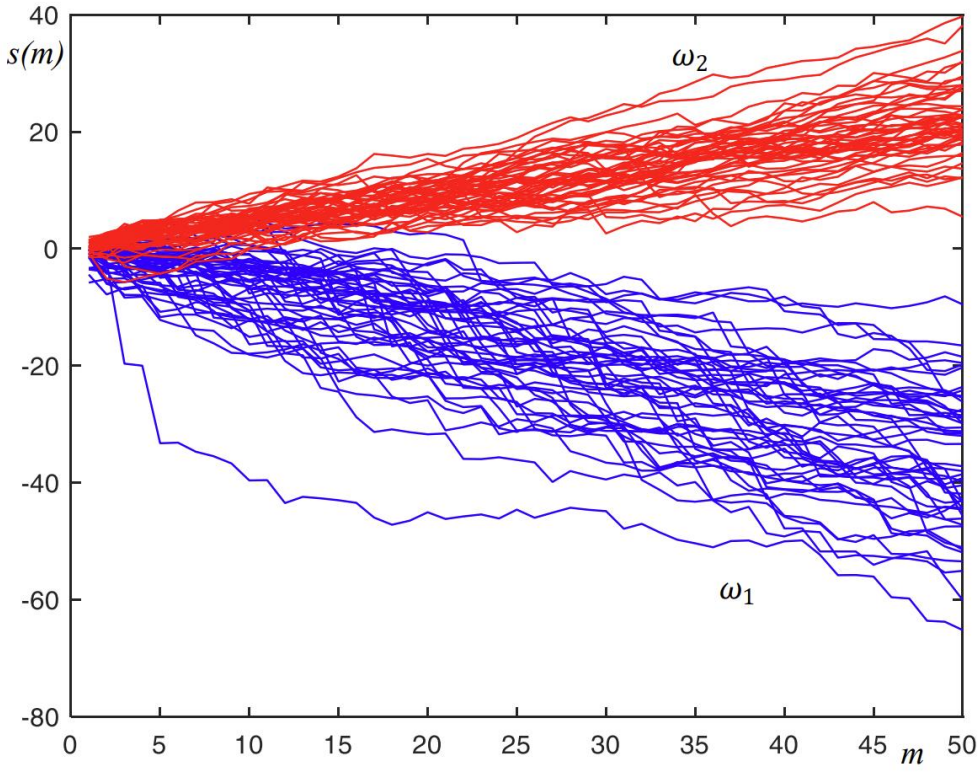
$$s_m = -ln\frac{f_1(X_1, \dots, X_m)}{f_2(X_1, \dots, X_m)}$$

Ако се одбрици $X_1, X_2, \dots, X_m$ некорелисани здружена функција густине вероватноће се своди на производ простих те се изрази изнад своде на:

$$l_m = \prod_{i=1}^{m} \frac{f_1(X_i)}{f_2(X_i)}$$

$$s_m = -\sum_{i=1}^{m} ln\, \frac{f_1(X_i)}{f_2(X_i)}$$

Ако пристигли одбирци потичу из прве класе за очекивати је да је $ln\, \frac{f_1(X_i)}{f_2(X_i)} > 0$, а ако пристигли одбирци долазе из друге класе за очекивати је да је $ln\, \frac{f_1(X_i)}{f_2(X_i)} < 0$. Одатле следи да ће s_m временом постајати све позитивнији или негативнији у зависности од тога из које класе потичу одбирци што је и приказано на слици 4. Пројектовање секвенцијалног теста се самим тим своди на одређивање прагова одлучивања за функцију s_m или l_m.



Слика 4. Приказ како се функција $s(m)$ мења за различите вредности дужине секвенце m за обе класе понаособ. Слика је преузета из [4].

Један од најпозантијих секвенцијалних тестова јесте Wald-ов секвенцијални тест који прагове одлучивања, које можемо означити са A и B, доводи у везу са вероватноћама грешке првог и другог типа. Детаљно извођење доступно је у литератури [3], а овде ће бити изложено само коначно изведено правило одлучивања:

$$l_m = \frac{f_1(X_1, \ \dots \ , X_m)}{f_2(X_1, \ \dots \ , X_m)} \geq A \Rightarrow X \ \in \ \omega_1$$

$$l_m = \frac{f_1(X_1, \ \dots \ , X_m)}{f_2(X_1, \ \dots \ , X_m)} \leq B \Rightarrow X \ \in \ \omega_2$$

$$B < l_m < A \Rightarrow \text{треба сачекати још један одбирак}$$

Веза вероватноћа грешке првог и другог типа са праговима одлучивања A и B изгледа овако:

$$\varepsilon_1 \ \cong \frac{B(A-1)}{A-B}$$

$$\varepsilon_2 \ \cong \frac{1-B}{A-B}$$

Користећи овај резулат, класификатор можемо пројектовати тако што одаберемо параметре A и B, па срачунамо ε_1 и ε_2 и видимо да ли нам оне одговарају. Такође можемо и прво одабарати ε_1 и ε_2, и на основу њих срачунамо прагове A и B. Сличан резултат се може изверсти и ако се користи функција s_m уместо l_m:

$$s_m = -ln\frac{f_1(X_1, \dots, X_m)}{f_2(X_1, \dots, X_m)} \geq a \Rightarrow X \in \omega_1$$

$$s_m = -ln\frac{f_1(X_1, \dots, X_m)}{f_2(X_1, \dots, X_m)} \leq b \Rightarrow X \in \omega_2$$

$$B < l_m < A \Rightarrow \text{треба сачекати још један одбирак}$$

$$a = -ln\, A \cong -ln\frac{1-\varepsilon_1}{\varepsilon_2}$$

$$b = -ln\, B \cong -ln\frac{\varepsilon_1}{1-\varepsilon_2}$$

Овако изведен Wald-ов секвенцијални тест поседује 3 важне особине:

1. Може се показати да везе између A и B и ε_1 и ε_2 важе чак и уколико одбирци $X_1, \dots, X_m$ нису независни, па чак не морају бити ни исто расподељени.
2. Тест се завршава са вероватноћом 1 тј. шанса да дискриминациони критеријум не пређе неки од прагова тежи 0 како број одбирака тежи бесконачности.
3. Тест минимизира средњу вредност потребног броја одбирака да се донесе одлука за задате вредности ε_1 и ε_2.

Све ово чини Wald-ов секвенцијални тест одличним за примену у пракси. Треба још додати да уколико постоји временско ограничење за доношење одлуке тј. ако је број одбирака $X_1, \dots, X_m$ ограничен након последњег одбирка можемо рећи да непознати предмет припада онај класи чијем прагу је ближа тренутна вредност l_m.

2.2 Параметарске методе класификације

У претходном делу рада приказане су методе класификације које захтевају тачно познавање уловних функција густине вероватноће сваке класе, а за циљ су имале минимизацију вероватноће грешке или у општем случају ризика или цене коју дефинишемо. Како тачна функција густине вероватноће никада није позната, а њена прецизна естимација је јако нумерички захтена, развијене су методе које користе само основне статистичке карактеристике које се могу брзо и лако проценити на основу тест

скупа података. То су најчешће матеамтичко очекивање и коваријациона матрица. Најчешће коришћени параметарски методи су линеарни, квадратни и део-по-део линеарни класификатор. Поред ових метода постоје и параметарски методи са нешто једноставнијим идејама које такође вредни споменути, а то су корелациони класификатор, 'matched' филтар и класификатор дистанце. Најпре ћемо се позабавити овим једноставнијим методама, а након тога биће обрађени линеарни, квадратни и део-по-део линеарни класификатор. Већина овог дела рада базирана је на литератури [4], па се ту могу пронаћи детаљи везани за ове класификаторе док ће овде бити предочене основне идеје и резултати.

2.2.1 Корелациони класификатор

Производ два вектора $M_i^T X$ се често назива корелацијом између вектора M_i и X и може се протумачити као груба процена сличности ових вектора. Ово својство можемо искористити када имамо два стохастичка процеса и реализацију $x(t)$ једног од тих процеса, а од интереса нам је да одредимо од ког процеса "потиче" $x(t)$. Ако $x(t)$ одабирамо у еквидистантним временским тренутцима $t_1, t_2, \dots, t_m$ можемо формитари вектор X као:

$$X = [x(t_1)\ \ x(t_2)\ \ \dots\ \ x(t_m)]^T$$

Даље потребно нам је да знамо параметре $m_i(t_1), \dots, m_i(t_m),\ i = 1{,}2$ који представљају математичко очекивање првог и другог стохастичког процеса од интереса у одговарајућим временским тренутцима. На основу ових параметара формирамо векторе M_1 и M_2:

$$M_i = [m_i(t_1)\ \ \dots\ \ m_i(t_m)]^T, i = 1{,}2$$

Када то све имамо можемо формулисати правило одлучивања на основу ког процењујемо да ли је $x(t)$ реализација првог или другог стохастичког процеса тј. да ли $x(t)$ припада првој или другој класи. То радимо тако што поредимо корелацију вектора X са векторима M_1 и M_2, па ако је X сличнији M_1 доносимо одлуку да $X \in \omega_1$ а у супротном процењујемо да $X \in \omega_2$. По потреби се може увести и неки праг одлучивања c помоћу ког класификатор постаје пристрасан једној од класа (мада се најчешће узима $c = 0$). Формално математички то можемо записати овако:

$$M_1^T X - M_2^T X > c \Rightarrow X \in \omega_1$$

$$M_1^T X - M_2^T X < c \Rightarrow X \in \omega_2$$

Ако пак имамо случај да $x(t)$ представља континуално мерење и ако нам је познато математичко очекивање стохастичких процеса $m_i(t)$ у сваком тренутку времена, можемо користити израз за корелацију континуалних случајних променљива:

$$\int_0^t m_i(\tau)x(\tau)d\tau\,, i = 1{,}2$$

Ова континуална варијанта корелације се рачуна за обе класе и пореди (уз евентуално додавање прага одлучивања c) исто као у дискретном случају:

$$\int_0^t m_1(\tau)x(\tau)d\tau - \int_0^t m_2(\tau)x(\tau)d\tau > c \Rightarrow X \in \omega_1$$

$$\int_0^t m_1(\tau)x(\tau)d\tau - \int_0^t m_2(\tau)x(\tau)d\tau < c \Rightarrow X \in \omega_2$$

2.2.2 ‘Matched’ филтар

Ако на основу функције $m_i(t)$ дефинишемо нову функцију $g_i(t)$ такву да је $g_i(T-t) = m_i(t)$ израз за корелацију $m_i(t)$ и $x(t)$ се своди на:

$$\int_0^t m_i(\tau)x(\tau)d\tau = \int_0^t g_i(T-\tau)x(\tau)d\tau\,, i = 1{,}2$$

Тиме смо добили да се корелација $m_i(t)$ и $x(t)$ своди на излаз филтра импулсног одзива $g_i(t)$ када се кроз њега провуче $x(t)$. Суштински користимо потпуно исти критеријум као код корелационог класификатора само на другачији начин реализујемо рачунање корелације. Дакле, потребно је испројектовати по један филтар за сваку класу тако да на излазу филтра добијамо корелацију улазног сигнала са математичким очекивањем стохастичког процеса. Даље само поредимо излазе филтара и на основу тога који излаз филтра је већи доносимо одлуку о припадности $x(t)$ једној од класа. Ова иста идеја се може применити и за дискретни случај.

2.2.3 Класификатор дистанце

Корелациони класификатор и ‘matched’ филтар имају велику сличност са веома једноставним класификатором, класификатором дистанце. Уколико су нам познати параметри M_1 и M_2 који представљају математичко очекивање облика из прве и друге

класе, можемо срачунати Еуклидско растојање (или квадрат Еуклидског растојања због једноставности рачунања) непознатог облика X од вектора M_1 и M_2. Поређењем да ли је облик X геометријски билижи очекивању M_1 или M_2 доносимо одлуку где да сврстамо X:

$$\|X - M_1\|^2 - \|X - M_2\|^2 < c \Rightarrow X \in \omega_1$$

$$\|X - M_1\|^2 - \|X - M_2\|^2 > c \Rightarrow X \in \omega_2$$

Као и код претходна два класификатора одабиром прага одлучивања c можемо фаворизовати смештање X у неку од класа. Ако су нам поред M_1 и M_2 познате и коваријационе матрице Σ_1 и Σ_2 уместо Еуклидског можемо користити статистичко растојање што даје боље перформансе:

$$d_1^2(X) - d_2^2(X) < c \Rightarrow X \in \omega_1$$

$$d_1^2(X) - d_2^2(X) > c \Rightarrow X \in \omega_2$$

$$d_i^2(X) = (X - M_i)^T \Sigma_i^{-1} (X - M_i), i = 1,2$$

2.2.4 Линеарни класификатор

Линеарни класификатор је, иако за нијансу компликованији од њих, близак техникама као што су кореалциони класификатор и класификатор дистанце. Релативна једноставност и интуитивност овог класификатора чине га једним од широко употребљиваних класификатора у пракси. Идеја овог класификатора је да се непознати одбирак X уз помоћ прецизно одређеног вектора V трасформише у скалар, а онда да се добијени скалар пореди са прагом v_0:

$$V^T X + v_0 < 0 \Rightarrow X \in \omega_1$$

$$V^T X + v_0 > 0 \Rightarrow X \in \omega_2$$

Израз $V^T X + v_0$ се назива линерана дискриминациона функција и означава се са $h(X)$. Оваквим поступком се сва обележја облика X агрегирају у један број. Геометријски се овај поступак може протумачити као пресликавање или пројекција n-димензионог вектра X у једну тачку на правој која представља продужетак вектора V. Суштина пројектовања линеарног класификатора је проналажења трансформационог вектора V и прага одлучивања v_0 тако да тачност класификације буде максимална. Притом треба нагласити да интензитет вектора V не игра никакву улогу већ је битно

одредити само његов правац у простору. За проналазак ових параметара на располагању нам је тест скуп одбирака.

Линеарни класификатор за дводимензионе облике даје граничну функцију која је права, за тродимензионе облике раван, а вишедимензионе облике хипер-раван.

Када одбирке из једне и друге класе пресликамо на вектор V добијамо две једнодимензионе расподеле за $h(X/\omega_1) = h_1(X)$ и $h(X/\omega_2) = h_2(X)$. Случајне променљиве $h_1(X)$ и $h_2(X)$ добијају се линеарном комбинацијом n случајних променљива (компоненете вектора X). То значи да за њих можемо претпоставити да имају Гаусовску нормалну расподелу захваљујући тврђењу централне граничне теореме. Са η_1 и η_2 означићемо средње вредности случајних променљива $h_1(X)$ и $h_2(X)$, а са σ_1 и σ_2 њихове стандарне девијације. Зарад што боље класификације пожељно је да се η_1 и η_2 што више разликују, а да σ_1 и σ_2 буду што мање. Ови захтеви се могу уврстити у критеријумску функцију коју онда треба оптимизовати и на основу те оптимизације пронаћи V и v_0.

Дефинисаћемо критеријсмску функцију у општем облику $J = J(\eta_1, \eta_2, {\sigma_1}^2, {\sigma_2}^2)$ и усвојити да је прва класа расподељена као $N(M_1, \Sigma_1)$, а друга као $N(M_2, \Sigma_2)$. $J(\eta_1, \eta_2, {\sigma_1}^2, {\sigma_2}^2)$ експлицитно зависис од параметара расподела $h_1(X)$ и $h_2(X)$, а имплицитно од V и v_0. Оптимизациони поступак подразумева прноалажење парцијалних извода и њихово изједначавње са нулом:

$$\frac{\delta J}{\delta V} = 0$$

$$\frac{\delta J}{\delta v_0} = 0$$

Даље детаљно извођење може се пронаћи у литератури [4], а резулат тог извођења гласи да се вектор V може срачунати по формули:

$$V = [s\Sigma_1 + (1-s)\Sigma_2]^{-1}(M_2 - M_1)$$

Где је s праметар који зависи од парцијалних извода критеријумске функције по ${\sigma_1}^2$ и ${\sigma_2}^2$. Испоставља се да аналитичко израчунавање вектора V није могуће, па се из тог разлога он одређује нумерички баш као и v_0. За нумеричко извођење се користи чињеница да параметар s може да поприма вредности само из опсега од нула до један. Две методе (засноване на сличном агоритму) које су најпрактичније за проналажење V и v_0 јесу метод ресупституције и ‘holdout’ метод.

Метод ресуституције каже следеће:

1) На основу обележених одбирака процени M_1, M_2, Σ_1 и Σ_2

2) Параметар s треба прошетати са малим кораком кроз све вредности на интервалу [0, 1]
3) За свако s срачунати вектор V и трансформисати све одбирке улазног скупа

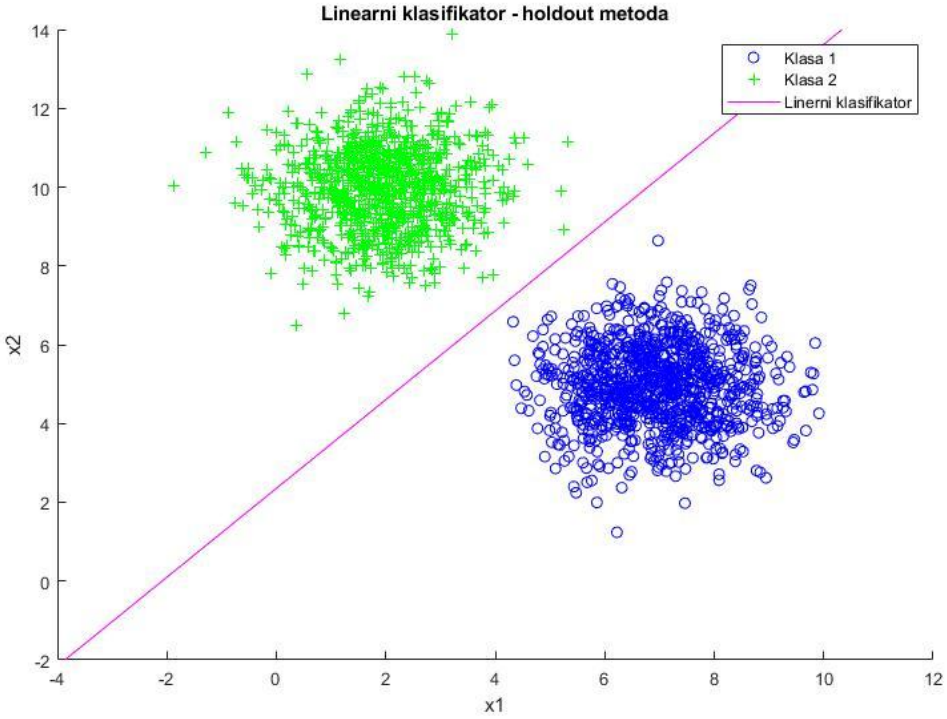
$$Y_i = V^T X_i, i = 1, \dots, N$$

4) Сада добијамо скуп једнодимензионих одбирака Y за које знамо којој класи припадају па на основу њих треба пронаћи v_0. Вредност v_0 треба "погађати" тако што све одбирке Y сортирамо и за могуће вредности v_0 узимамо по једну тачку која се налази између свака два суседна трансформисана одбирка Y_i и Y_j.
5) За свако v_0 извршимо класификацију одбирака Y_i:

$$Y_i > -v_0 \Rightarrow X_i, Y_i \in \omega_1$$
$$Y_i < -v_0 \Rightarrow X_i, Y_i \in \omega_2$$

6) Затим проценимо грешку класификације за свако v_0 и памтимо за које v_0 је постигната најбоља тачност за тренутно s.
7) Тако треба испробати свако v_0 за свако s и узети оно v_0 и оно s (на основу s добијамо V) за које је најбоља тачност класификације

'Holdout' метод је заснован на истом алгоритму осим што се код овог метода улазни скуп података подели на тренинг и на тест скуп. Одбрци из тренинг скупа се корсисте да се, на исти начин као код метода ресупституције, одреде могуће вредности параметара V и v_0, али се сада квалитет потенцијалног решења V и v_0 процењује преко тест скупа одбирака. Овај метод уважава чињеницу да није статистички оправдано тестирати класификатор на одбирцима на онсову којих је пројектован. Пример пројектованог класификатора 'holdout' методом приказан је на слици 5. Може се приметити да пошто су класе лако сепарабилне линеарни класификатор их јако добро разграничава и ниједан одбирак није погрешно класификован.



Слика 5. Пример линеарног класификатора пројектованог 'holdout' методом.

2.2.5 Квадратни класификатор

Главна мана линеарног класификатора је да се он може користити само код класа које су линерано сепарабилне. То су класе код којих сепарабилност потиче од међусобне удаљености математичких очекивања одбирака. За проблеме класификације где то није случај, за разграничење одбирака потребне су криве другог реда као што су кружница, парабола, хипербола (ако говоримо о дводимензионом простору обележја). У ту сврху треба пројектовати квадратни класификатор облика:

$$h(X) = X^T Q X + V^T X + v_0 < 0 \Rightarrow X \in \omega_1$$

$$h(X) = X^T Q X + V^T X + v_0 > 0 \Rightarrow X \in \omega_2$$

Сада $h(X)$ неће зависити само од компоненти X већ и од њихових квадрата као и међусобних производа. Проблем проналажења квадратног класификатора се може свести на проблем налажења линеарног класификатора, да би то илустровали расписаћемо израз за $h(X)$ у случају да радимо са дводимензионим одбирцима (поступак за вишедимензионе одбирке изгледа потпуно аналогно само са више чланова):

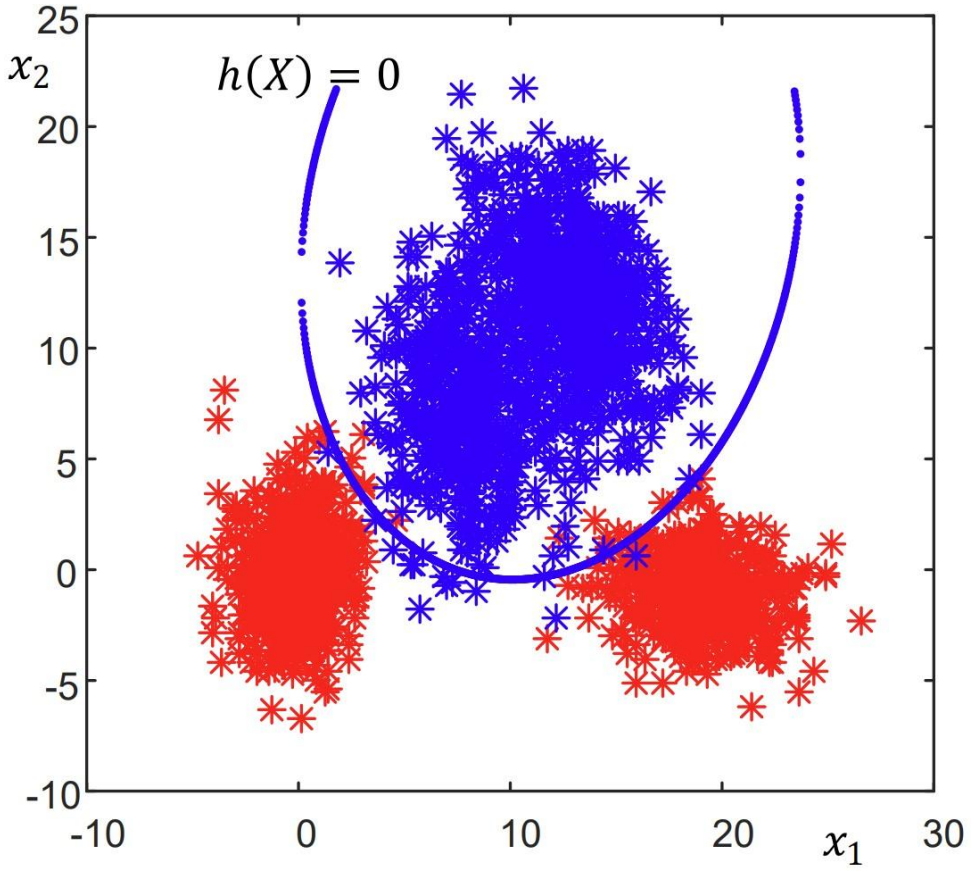
$$h(X) = [x_1\ x_2]\begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [v_1\ v_2]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + v_0$$

$$h(X) = q_{11}{x_1}^2 + (q_{12} + q_{21})x_1x_2 + q_{22}{x_2}^2 + v_1x_1 + v_2x_2 + v_0$$

$$h(X) = [q_{11}\ \ (q_{12} + q_{21})\ \ q_{22}\ \ v_1\ \ v_2][{x_1}^2\ \ x_1x_2\ {x_2}^2\ \ x_1\ \ x_2]^T + v_0$$

$$h(X) = {V'}^T X' + v_0$$

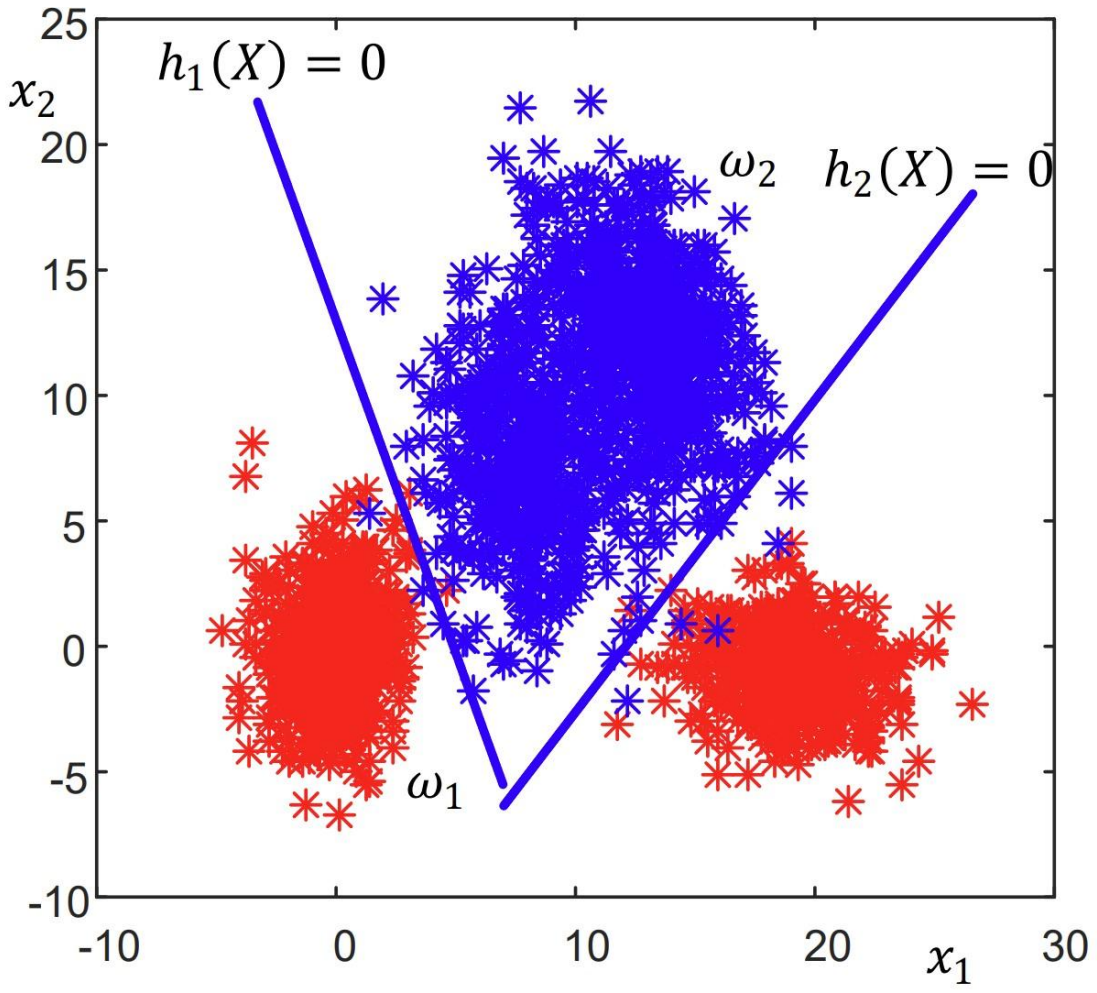
Из овога се може видети да се квадратни класификатор може пројектовати по угледу на линеарни тако што од улазних вектора X направимо векторе X' који ће укључивати и производе компоненти, а онда за такав нови вектор тражимо линеарни класификатор са трансформационим вектором V' из ког онда можем извући параметре вектора V и матрице Q. Треба нагласити да са повећањем димензионалности проблема n, потребно рачунарско време за проналажење квадратног класификатора расте сразмерно са n^2. Пример пројектованог квадратног класификатора приказан је на слици 6. Са слике се може приметитит да класе из примера не би могле добро да се разграниче линеарним класификатором.



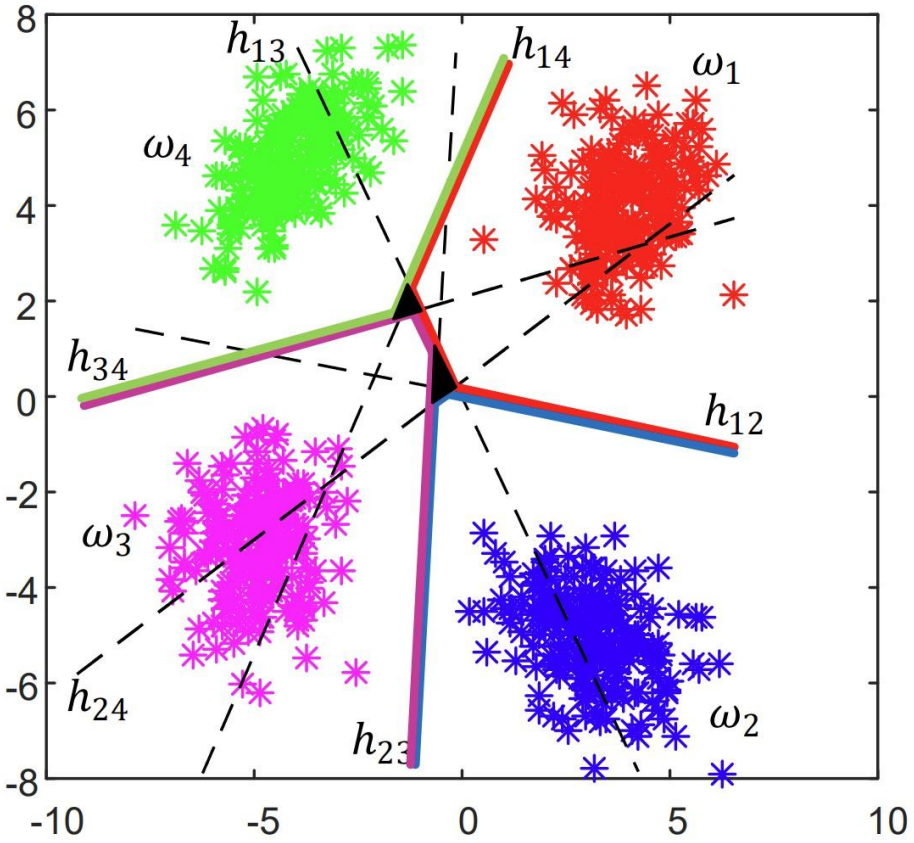
Слика 6.Пример испројектованог квадратног класификатора. Слика је преузета из [4].

2.2.6 Део-по-део линеарни класификатор

Део-по-део линеарни класификатори проширује идеју линеарног класификатора тако што ћемо за класе које нису линеарно сепарабилне користити већи број линија да их разграничимо (илустрација на слици 7). Такође, уколико имамо више од две класе то можемо решити део-по-део линеарним класификатором тако што пројектујемо линарени класификатор између сваке две класе па на онсову више линеарних класификатора доносимо одлуку у коју класу треба сврстати одбирак (илустрација на слици 8).



Слика 7. Пример коришћења део-по-део линаерног класификатора за препознавање класа које нису линеарно сепарабилне. Слика је преузета из [4].



Слика 8. Пример коришћења део-по-део линаерног класификатора за препознавање више од две класе. Слика је преузета из [4].

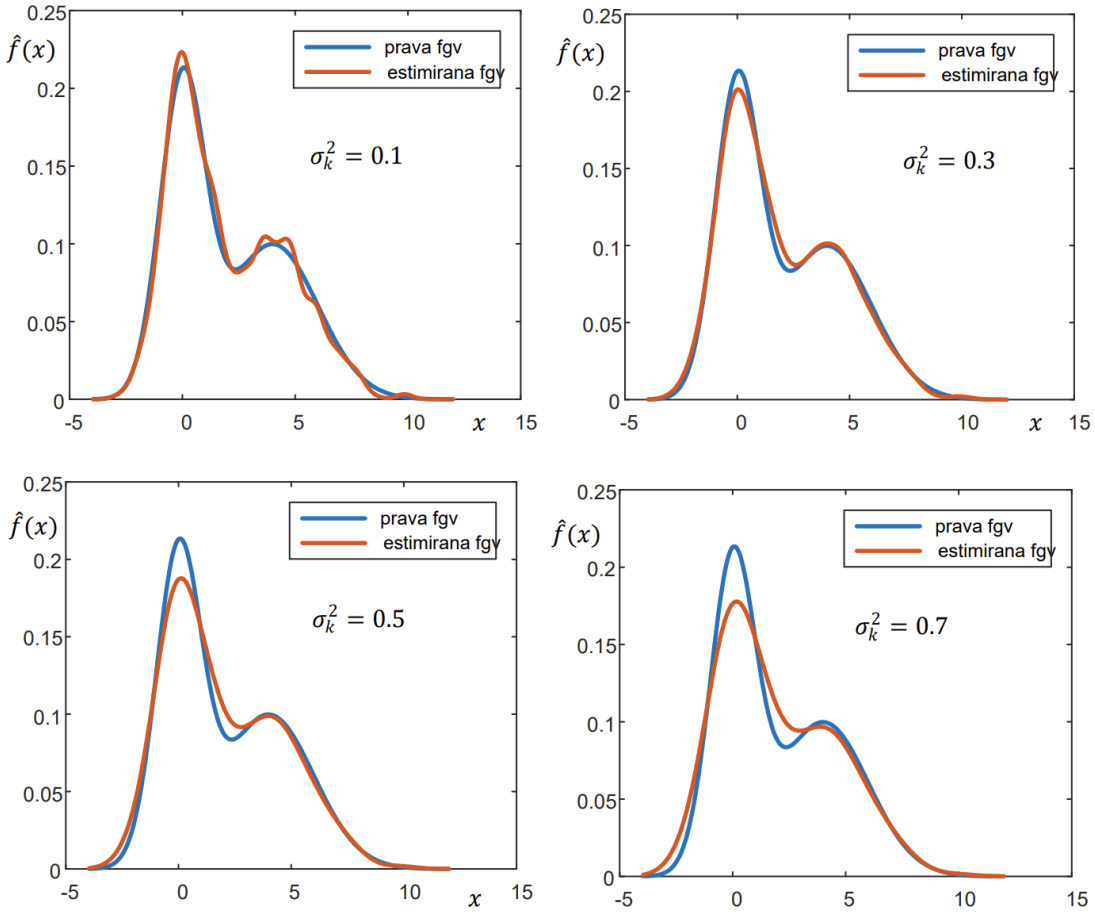
2.3 Непараметарске методе класификације

Непараметарске методе класификације су методе које имају за циљ да што боље процене функцију густину вероватноће улазних одбирака, а онда за процес доношења одлуке користе принципе тестирања хипотеза. Једани од најпознатијих представника непараметарских метода јесу кернел функције и метод k најближих суседа.

2.3.1 Кернел функције

Естимација функције густине вероватноће преко кернел функција се одвија на следећи начин. За сваки одбирак тест скупа дефинишемо око њега малу функицју густине вероватноће најчешће Гаусовског типа. Естимација здружене функције густине вероватноће се своди само на сумирање великог броја малих кернел функција. Кључан

избор при оваквој естимацији предстаља варијанса кернел функције. Сувише велика варијанас чини да естимирана здружена фукција густине вероватноће буде превише блага и слабо прати праву фгв. С друге стране превише мали избор варијансе чини естимацију превише оштром и почињу да се појављају пикови који не постоје у стварној функцији густине вероватноће. На слици 9 се може видети поређење естимиране фгв са правом фгв за различит одабир варијансе кернел функције.



Слика 9. Пример коришћења естимиране једнодимензионе функције густине вероватноће на основу кернел функција са различитом варијансом. Слика је преузета из [4].

2.3.2 Метод *k* најближих суседа

Метод к најближих суседа је једноставан за имплементацију, али у пракси резулати нису увек толико добри. Метод k најближих суседа се може и директно користити за класификацију непознатих одбирака, а може се преко њега и вршити естимација функције густине вероватноће. Идеја knn (енг. k nearest neighbours) класификатора је да за непознати одбирак треба пронаћи његових k најближих суседа у тест скупу

података. Доносим одлуку да непознати одбирак припада класи i ако међу k суседа из тест скупа има највише оних који су из класе i. Сличан принцип се користити и за естимацију функције густине вероватноће. Да бих естимирао функцију густине вероватноће у тачки у којој се налази одбирак посматрању геометријски колико далеко од посматраног одбирка морам да се померим да бих му нашао к суседа. Прецизније, посматрању колика је најмања запремина сфере око одбирка таква да се унутар ње налази бар k суседа, та запремина се обележава са $v(X)$. Јасно је да што је густина одбирака већа то ће ова запремина бити мања и обрнуто. На основу ове запремине, броја k и укупног броја одбирака N добија се естимација за функцију густине вероватноће у тачки где се налази одбирак од интереса:

$$\hat{f}(X) = \frac{k-1}{Nv(X)}$$

Од великог значаја је одабир броја суседа k. Оптимално k зависи од типа расподеле, а интуитивно је јасно и да k мора зависити од укупног броја одбирака N. Што је веће N треба бирати веће k. Неко генерално правило је да:

$$k = N^{\propto}, \qquad \propto \in (0.42, 0.61)$$

Уколико су класе лако сепарабилне метод k најближих суседа иако једноставан даје јако добре резулате. Уколико класе нису лако сепарабилне најчешће се исплати пројектовати класификатор неком другом методом.

3 Технике за кластеризацију података

Проблем кластеризације представља проблем класификација без расположивог тест скупа података. Наиме имамо скуп облика који треба да групишемо у кластере, али немамо никакаво априорно знање о одбирцима. Кластеризација се још често назива и класификација без супервизије (енг. unsupervised learning). Од интереса је пројектовати алгоритам који ће успети да увиди сличност између одбирака на основу којих их може поделити на класе. Резултат таквог алгоритма јесте издвојање делова у простору обележја у коме је група одбирака значајно сконцентрисана. Најпре ћемо математички формализовати проблем класификације.

Нека је N облика на улазу алгоритма класификације обележено са $X_1, \ \dots \ , X_N$. Сваки одбирак треба придружити једној од L класа $\omega_1, \dots, \omega_L$. Резултат класификације је вектор $\Omega = [\omega_{k1} \ \ \omega_{k2} \ \dots \ \ \omega_{kN}]$ где ω_{ki} представља класу којој је придружен. Вектор Ω се може срачунати минимизацијом некаквог критеријума (параметарска класификација) или уз помоћ алгоритма који тражи долине у фукцији густине вероватноће података (непараметарска класификација). Ако се идемо путем оптимизације критеријума, критеријум се обично бира тако да у себи садржи меру сепарабилности класа. Чест избор критеријума базиран је на матрицама унутаркласног и међукласног расејања. Алгоритми кластеризације који врше минимизацију критеријума се најчешће извршавају итеративно где се најпре одабере иницијална кластеризација $\Omega(0)$, затим се за њу срачуна критеријум и онда се посматра како се може рекласификовати сваки појединачни одбирак тако да та рекласификација доведе до смањења критеријумске функције. Након што се одреде рекласификације које изгледају обећавајуће, одговарајући облици се рекласификују, а поступак се опет понавља испочетка и тражи се још боље решење. Алгоритам се зауставља када би рекласификација било ког одбирка резултовла повећањем критеријумске функције.

Најпопуларнији параметарски метод k-mean clustering, а један од најпопуларнијих непараметарских метода јесте subtractive clustering. Ова два алгоритма ће бити детаљније описана, а затим ћемо их и упоредити на практичним примерима.

3.1 C-mean кластеризација

C-mean кластеризација је веома једноставан алгортиам рекласификације базиран на принципу најближе средње вредности. Оваквим алгоритмом максимизоваће се сепрабилност класа. Овај алгоритам спада у групу параметарских метода, а критеријумска функција J је дефинисана на следећи начин:

$$J = tr({S_m}^{-1} S_W)$$

S_m је матрица међукласног расејања а S_W матрица унутаркласног расејања. Овако дефинисан критеријум се аритметичким сређивањем (формални математички поступак досупан је у литератури [6]) може свести на:

$$J = \frac{1}{N} \sum_{r=1}^{L} \sum_{j=1}^{N_r} \left\| X_j^{(r)} - M_r \right\|^2$$

N_r представља број одбирака који се тренутно налази у класи ω_r, M_r представља средњу вредност одибрака који су тренутно у класи ω_r, а $X_j^{(r)}$ представља један од одбирака који је тренутно класивикован да припада ω_r. Овај критеријум сугерише да ће одбирци бити класификовани у L класа тако да се минимизује растојање одбирака унутар једне класе од њихове средње вредности.

3.2 Subtractive clustering

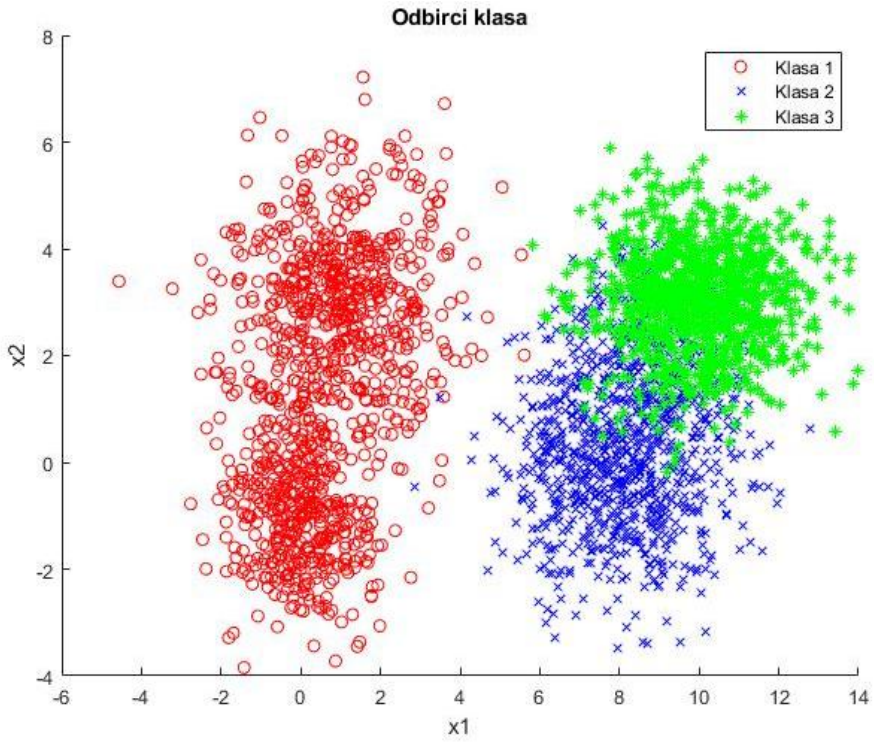
Subtractive clustering је веома интерсантан алгоритам који се бази на рачунање density measure функције у свим тачкама улазног скупа података. Density measure веома личи на функцију густине вероватноће и на основу ње процењују тачке које ће престављати центре оформљених кластера. Након што изаберемо центар једног кластера од density measure функције свих осталих одбирака морамо одузети утицај кластера који смо управо одвојили. У том поступку морамо да бирамо параметар радијус утицаја центра кластера. Алгоритам понавља ове кораке све док не извуче одогврајући број центара кластера или док му не понестане одбирака који могу постати нови центри кластера.

4 Поређење метода за кластеризацију података

Изложене методе кластеризације тестираћемо и поредити на три примера вештачки генерисаних података. Тестирање је рађено у програмском пакету Matlab. Тест подаци су генерисани кроз функцију “mvrnd” (литература [7]) у виду неколико класа. Наравно, алгоритмима кластеризације није прослеђена информација из које класе долазе генерисани одбирци. Свакак класа је имала по 1000 одбирака. За k-mean кластеризацију коришћена је функција “kmeans” (литература [8]). За subtractive clustering методу коришћена је функција “subclust” (литература [9]). Код коришћен у примерима који следе дат је у прилогу Б.

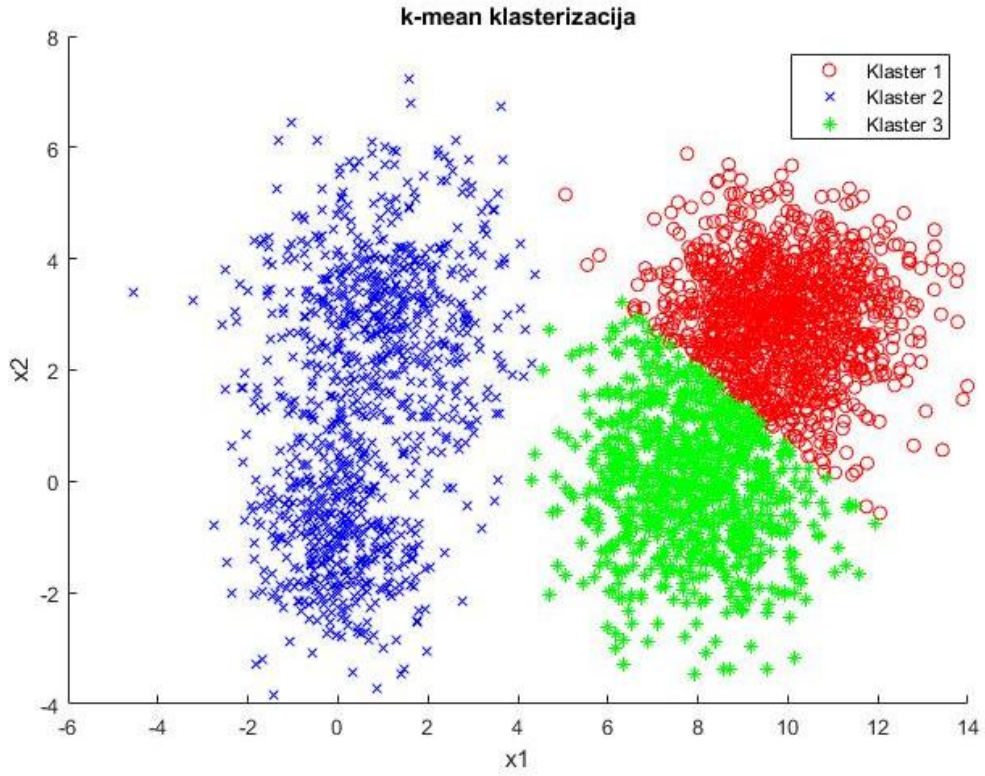
4.1 Пример 1

У првом примеру генерисани су подаци из три класе са дводимензионим вектором обележја. Једна класа је Гаусовски расподељена, а друге две имају бимодалну Гаусову расподелу. Подаци су приказани на слици 10.



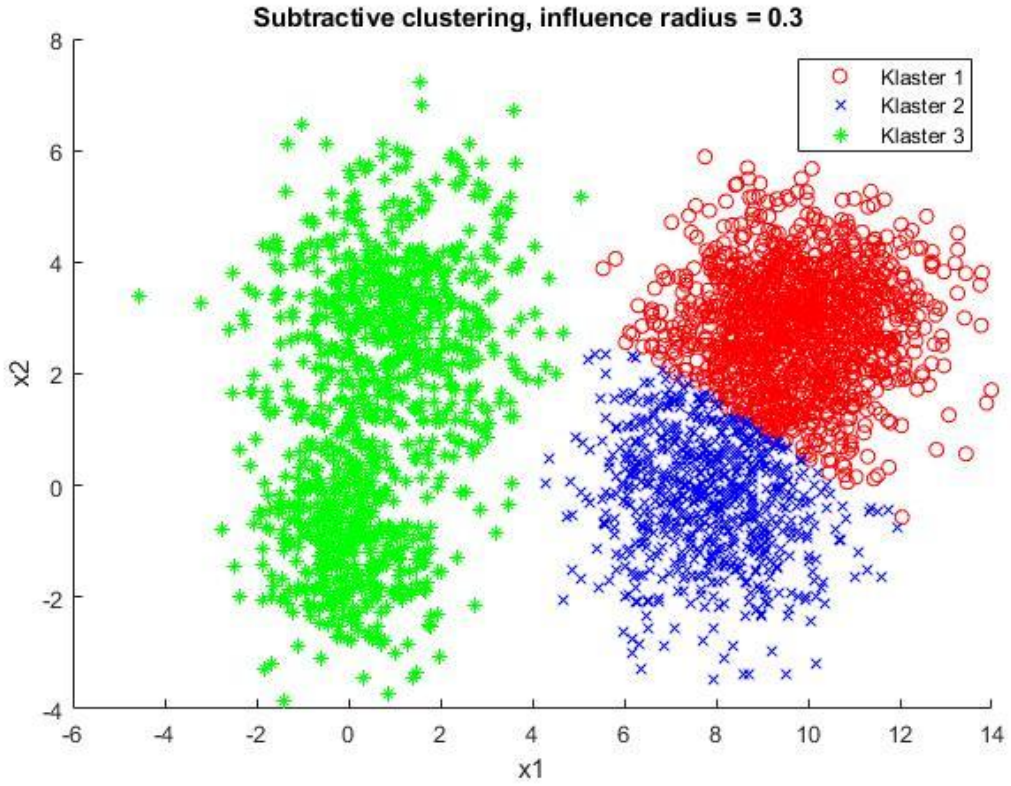
Слика 10. Приказ ситетички генерисанх података у дводименционом простору обележја.

Добијени кластери кроз k-mean кластеризацију дази су на слици испод.



Слика 11. Добијени кластери k-mean методом.

У први кластер је смештно 1197, у други кластер 998 а у трећи кластер 805 одбирака. Са слике 11 видимо да је било проблематично раздвојити одбирке оригиналних класа 2 и 3 (кластер 1 и кластер 3) о чему сведочи и број одбирака унутар кластера, међутим то је и било за очекивати с обзиром на њихово међусобно преклапање. На наредној слици можемо видети резултат subtractive clustering-a.

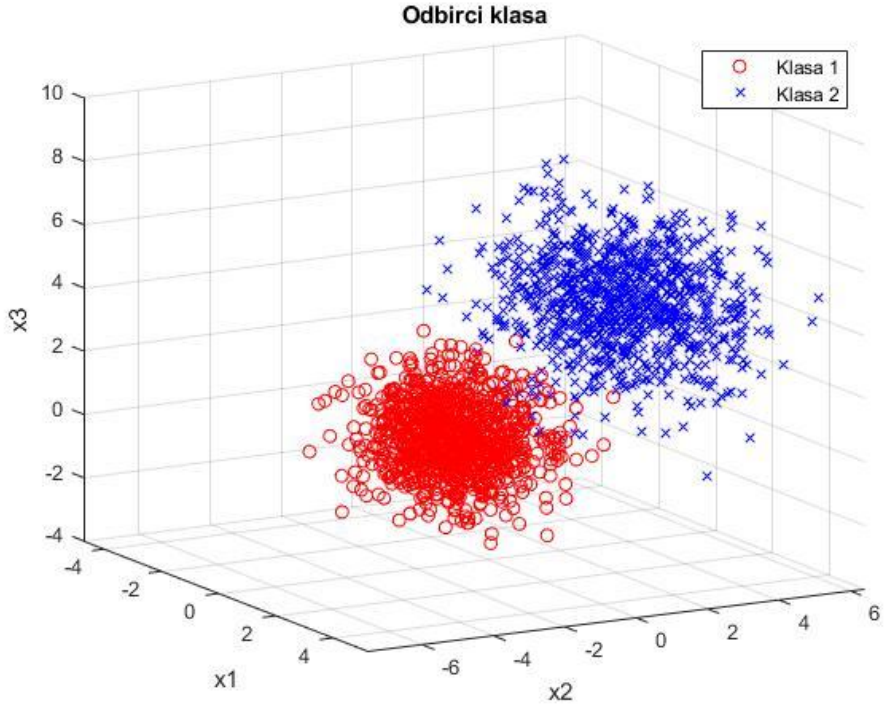


Слика 12. Добијени кластери subtractive clustering методом

У први кластер је смештно 1287, у други кластер 712 а у трећи кластер 1001 одбирак. Као и код k-mean метода проблем је био класификовати оригиналне класе 2 и 3. Све у свему може се закључити да се у овом случају k-mean метода показала нешто бољом.

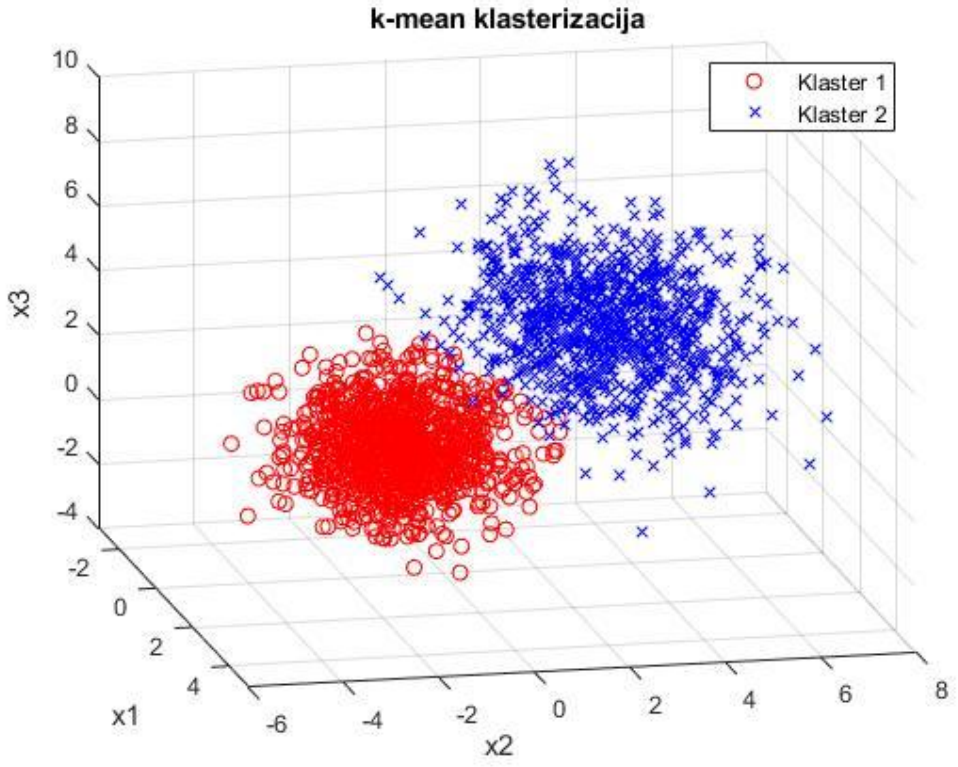
4.2 Пример 2

У примеру 2 генерисани су Гаусовски расподељени (тродимензиони) одбирци две класе које су релатицно лако сепарабилне. Па је за њих опет спроведен поступак кластеризације. Одбирци изгледају овако:

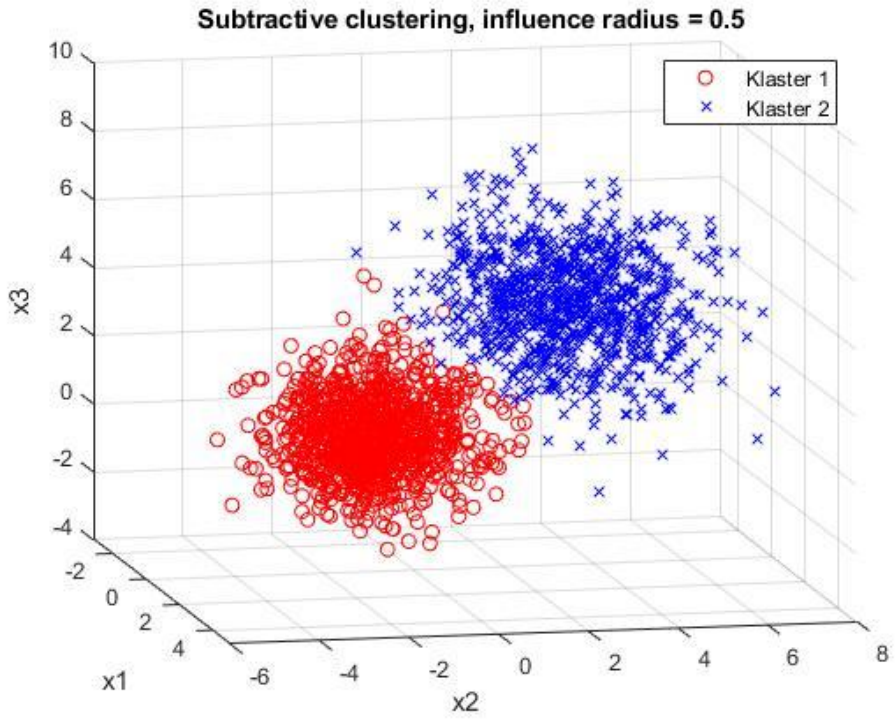


Слика 13. Приказ ситетички генерисанх података у тродименционом простору обележја.

Добијени су кластери:



Слика 14. Добијени кластери k-mean методом.



Слика 15. Добијени кластери subtractive clustering методом.

K-mean агоритам је у овом примеру у први кластеr ставио 1024 обирка, а у други 976. Subtractive clustering алгоритам је у овом примеру у први кластер ставио 1027 одбирака, а у други 973. На овом примеру где су класе лако сепарабилне оба алгоритма су показала готово исте перформансе.

5 Закључак

На основу приказаних екесперимената може се закључити да k-mean кластеризације ради боље или макар са једнаким перформансама као и subtractive clustering. Треба нагласити и да перформансе subtractive clustering-а у многоме зависе од радијуса утицаја центара кластера што у овом раду није детаљно испитано. Вреди нагласити да је k-mean кластеризација итеративни алгоритам чије извршавање може да траје јако дуго, док се subtractive clustering алгоритам извршава само у онолико итерација колики је и жељени број кластера. Због овога се може дати предност subtractive clustering алгоритму уколико су класе лако сепарабилне. У том случају ћемо добити исти учинак али уз мање потрошеног процесорског времена.

Ствари које би се још додатно могле испитати у будућности јесу:

- Шта се дешава ако се број димензија вишеструко увећа ($n \gg 3$)
- Шта се дешава ако један кластер има много више одбирака него други
- Како се ови класификатори показују када је жељени број кластера много већи од 2 и 3
- Како се мења понашање subtractive clustering са мењањем радијуса утицаја центара класт.

6 Литература

[1] Жељко Ђуровић, “Predavanje 1”, материјал са предмета Препознавање облика 13Е054ПО, Универзитет у Београду - Електротехнички факултет, 13E054PO (bg.ac.rs).
[2] Жељко Ђуровић, “Predavanje 2”, материјал са предмета Препознавање облика 13Е054ПО, Универзитет у Београду - Електротехнички факултет, 13E054PO (bg.ac.rs).
[3] Жељко Ђуровић, “Predavanje 3”, материјал са предмета Препознавање облика 13Е054ПО, Универзитет у Београду - Електротехнички факултет, 13E054PO (bg.ac.rs).
[4] Жељко Ђуровић, “Predavanje 4”, материјал са предмета Препознавање облика 13Е054ПО, Универзитет у Београду - Електротехнички факултет, 13E054PO (bg.ac.rs).
[5] Жељко Ђуровић, “Predavanje 5”, материјал са предмета Препознавање облика 13Е054ПО, Универзитет у Београду - Електротехнички факултет, 13E054PO (bg.ac.rs).
[6] Жељко Ђуровић, “Predavanje 7”, материјал са предмета Препознавање облика 13Е054ПО, Универзитет у Београду - Електротехнички факултет, 13E054PO (bg.ac.rs).
[7] https://www.mathworks.com/help/stats/mvnrnd.html , The Mathworks, Inc. , Natick, USA.
[8] https://www.mathworks.com/help/fuzzy/subclust.html , The Mathworks, Inc. , Natick, USA.
[9] https://www.mathworks.com/help/stats/kmeans.html , The Mathworks, Inc. , Natick, USA.

ПРИЛОГ А

Често коришћене ознаке и њихово објашњење:

N - укупан број одбирака

N_i - број одбирака из i-те класе

L - број класа у проблему

ω_i - конкретна i-та класа

$f_i(X)$ - функција густине вероватноће одбирака из i-те класе

$f(X)$ - здружена функција густине вероватноће свих одбирака

M_i - математичко очекивање облика из i-те класе

Σ_i - коваријациона матрица облика из i-те класе

P_i - априорна вероватноћа појаве облика из i-те класе, $\sum_{i=1}^{L} P_i = 1$

$q_i(X)$ - вероватноћа да одмирак X припада класи ω_i

ПРИЛОГ Б – код за пример 1

```
clear
close all
clc


%% Primer 1

% Generisanje podataka

rng(1)

N = 1000;

M1 = [0; -1]; S1 = [1 -0.1; -0.1 1];
M2 = [1; 3]; S2 = [2 0; 0 2];
M3 = [9; 2]; S3 = [1 -0.3; -0.3 1];
M4 = [8; 0]; S4 = [2 0; 0 1.8];
M5 = [10; 3]; S5 = [1.8 0; 0 1];

P11 = 0.4; P12 = 1 - P11;
P21 = 0.2; P22 = 1 - P21;

X1 = mvnrnd(M1, S1, N);
X2 = mvnrnd(M3, S3, N);
X3 = mvnrnd(M5, S5, N);

for i = 1:N
    if rand > P11
        X1(i, 1:2) = mvnrnd(M2, S2);
    end
    if rand > P21
        X2(i, 1:2) = mvnrnd(M4, S4);
    end
end

%% Prikaz odbiraka klasa

figure(1)
hold all
plot(X1(:, 1), X1(:, 2), 'ro')
plot(X2(:, 1), X2(:, 2), 'bx');
plot(X3(:, 1), X3(:, 2), 'g*');
xlabel('x1');
```

```
ylabel('x2');
legend('Klasa 1', 'Klasa 2', 'Klasa 3');
title('Odbirci klasa')


%% Klasterizacija
X = [X1; X2; X3];
k = 3;

idx_k = kmeans(X, k);

X1k = X(idx_k == 1, :);
X2k = X(idx_k == 2, :);
X3k = X(idx_k == 3, :);

disp(length(X1k))
disp(length(X2k))
disp(length(X3k))

figure(2)
hold all
plot(X1k(:, 1), X1k(:, 2), 'ro')
plot(X2k(:, 1), X2k(:, 2), 'bx');
plot(X3k(:, 1), X3k(:, 2), 'g*');
xlabel('x1');
ylabel('x2');
legend('Klaster 1', 'Klaster 2', 'Klaster 3');
title('k-mean klasterizacija')


idx_s = subtractive_clust(X, 3, 0.3);

X1s = X(idx_s == 1, :);
X2s = X(idx_s == 2, :);
X3s = X(idx_s == 3, :);

disp(length(X1s))
disp(length(X2s))
disp(length(X3s))

figure(3)
hold all
plot(X1s(:, 1), X1s(:, 2), 'ro')
plot(X2s(:, 1), X2s(:, 2), 'bx');
plot(X3s(:, 1), X3s(:, 2), 'g*');
xlabel('x1');
ylabel('x2');
legend('Klaster 1', 'Klaster 2', 'Klaster 3');
title('Subtractive clustering, influence radius = 0.3 ')
```

ПРИЛОГ В – код за пример 2

```
clear
close all
clc


%% Primer 2

% Generisanje podataka

rng(1)

N = 1000;

M1 = [0; -1; 0]; S1 = [1 -0.1 0; -0.1 1 0 ; 0 0 1];
M2 = [1; 3; 4]; S2 = [2 0 0; 0 2 0; 0 0 2];


X1 = mvnrnd(M1, S1, N);
X2 = mvnrnd(M2, S2, N);


%% Prikaz odbiraka klasa

figure(1)
hold all
plot3(X1(:, 1), X1(:, 2), X1(:, 3), 'ro')
plot3(X2(:, 1), X2(:, 2), X2(:, 3), 'bx');
xlabel('x1');
ylabel('x2');
zlabel('x3');
grid on;
legend('Klasa 1', 'Klasa 2');
title('Odbirci klasa')

%% Klasterizacija
X = [X1; X2];
k = 2;

idx_k = kmeans(X, k);

X1k = X(idx_k == 1, :);
X2k = X(idx_k == 2, :);
```

```
disp(length(X1k))
disp(length(X2k))

figure(2)
hold all
plot3(X1k(:, 1), X1k(:, 2), X1k(:, 3), 'ro')
plot3(X2k(:, 1), X2k(:, 2), X2k(:, 3), 'bx');
xlabel('x1');
ylabel('x2');
zlabel('x3');
grid on;
legend('Klaster 1', 'Klaster 2');
title('k-mean klasterizacija')


idx_s = subtractive_clust(X, 2, 0.5);

X1s = X(idx_s == 1, :);
X2s = X(idx_s == 2, :);

disp(length(X1s))
disp(length(X2s))

figure(3)
hold all
plot3(X1s(:, 1), X1s(:, 2), X1s(:, 3), 'ro')
plot3(X2s(:, 1), X2s(:, 2), X2s(:, 3), 'bx');
xlabel('x1');
ylabel('x2');
zlabel('x3');
grid on;
legend('Klaster 1', 'Klaster 2');
title('Subtractive clustering, influence radius = 0.5 ')
```

ПРИЛОГ Г – код за subtractive clustering

```
function idx = subtractive_clust(X, num_of_clusters,
influence_rng)
centers = subclust(X, influence_rng);
idx = zeros(length(X), 1);

for i = 1:length(X)
    dist = zeros(num_of_clusters, 1);

    for j = 1:num_of_clusters
        dist(j) = sum((centers(j, :)-(X(i, :))).^2);
    end
    [~ , ind] = min(dist);
    idx(i) = ind;
end

end
```