УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Александар А. Картељ

Примене метахеуристике засноване на електромагнетизму у решавању проблема класификације

Докторска дисертација

Београд, 2014

UNIVERSITY OF BELGRADE
FACULTY OF MATHEMATICS

Aleksandar A. Kartelj

Applications of electromagnetism-like metaheuristic in solving classification problems

Doctoral dissertation

Belgrade, 2014

Подаци о ментору и члановима комисије

Ментор

др Владимир Филиповић, ванредни професор, Математички факултет, Универзитет у Београду

Чланови комисије

др Душан Тошић, редовни професор, Математички факултет, Универзитет у Београду

др Вељко Милутиновић, редовни професор, Електротехнички факултет, Универзитет у Београду

др Ненад Митић, ванредни професор, Математички факултет, Универзитет у Београду

др Саша Малков, доцент, Математички факултет, Универзитет у Београду

Датум одбране:

Подаци о докторској дисертацији

Наслов докторске дисертације: Примене метахеуристике засноване на електромагнетизму у решавању проблема класификације

Резиме: У овом раду се испитују могућности побољшавања процеса класификације кроз разматрање три проблема која се појављују у класификацији: проблем одабира атрибута, проблем подешавања тежина атрибута и проблем подешавања параметара класификатора. Сва три проблема су изазовна за решавање и налазе се у фокусу начуних истраживања у области машинског учења. За њихово решавање, у раду су предложене популационе метахеуристичке методе засноване на електромагнетизму. Метахеуристика заснована на електромагнетизму је метода за комбинаторну и глобалну оптимизацију која је инспирисана законитостима привлачења и одбијања наелектрисаних честица. Свака честица је представљена као низ реалних вредности. Решење проблема се добија уз помоћ пресликавања низова реалних вредности честица у скуп допустивих решења. Честице које се пресликавају у боља решења, остварују виши ниво наелектрисања, што за последицу има да те честице имају јачи утицај на остале. Итеративним померањем популације честица индукованог силама између наелектрисаних честица, врши се претрага простора могућих решења. У имплементацијама метода се водило рачуна о два аспекта: 1) квалитету класификације који се добија након примене оптимизационе методе и 2) ефикасности предложених метода са аспекта временских и просторних ресурса. У свим методама су имплементиране специфичне процедуре локалне претраге зависне од природе проблема, у циљу побољшања квалитета решења.

Решавањем проблема одабира атрибута, врши се двојако побољшавање класификационог процеса. Елиминацијом непотребних атрибута може се смањити шум који нарушава класификациони модел, а истовремено се смањује димензија улазног проблема, па се и ефикасност процеса класификације повећава. Проблем одабира атрибута је врло ефикасно решен предложеном методом, при чему је квалитет класификације у великом броју случајева (тест проблема) унапређен у односу на методе из литературе. Код неких тест проблема, времена

извршавања предложене методе су и по неколико стотина пута мања од времена извршавања сродних метода из литературе.

Проблем подешавања тежина и подешавања параметара деле сличну репрезентацију решења, јер је у оба случаја реч о низовима реалних вредности. Пошто је и репрезентација наелектрисаних честица заснована на низовима реалних вредности, омогућен је *гладак* прелаз из простора честица у простор решења. Квалитет методе за решавање проблема подешавања тежина је демонстриран на методи најближих суседа. Извршена су тестирања над разнородним скуповима тест проблема и поређења са неколико метода из литературе. У већини случајева, предложена метода је надмашила остале упоредне методе.

Подешавање параметара класификатора има велики утицај на квалитет класификације. Предложена метода за подешавање параметара је примењена на методи подржавајућих вектора, која има сложену параметарску структуру када су у питању број параметара и домени њихових вредности. Хеуристичком иницијализацијом решења убрзано је проналажење региона квалитетних комбинација параметара. Извршена су темељна тестирања над тест проблемима различитих димензија и различите структуре атрибута: хомогене и хетерогене. У случају хомогене структуре, примењено је учење појединачних кернела, док се код хетерогених података користило вишекернелско учење. Упоредна анализа са методама из литературе је показала супериорност предложене методе када је у питању учење засновано на једном или више кернела са радијалном основом. Такође је показано да у осталим случајевима, предложена метода даје упоредиве резултате.

Све предложене методе су допринеле побољшању квалитета класификације. Због начина на који разматрају проблеме, све три методе се могу уопштити, и применити над широком класом класификационих модела и/или класификационих проблема.

Кључне речи: класификација, електромагнетизам, истраживање података, метахеуристике, машинско учење, оптимизација
Научна област: Рачунарство
Ужа научна област: Оптимизација
УДК број: [[004.832.2+004.85]:519.863]:537(043.3)

Dissertation Data

Doctoral dissertation title: Applications of electromagnetism-like metaheuristic in solving classification problems

Abstract: This work investigates the potential of improving the classification process through solving three classification-related problems: feature selection, feature weighting and parameter selection. All three problems are challenging and currently in the focus of scientific researches in the field of machine learning. Each problem is solved by using population-based metaheuristic method called electromagnetism-like method. This method is used for combinatorial and global optimization. It is inspired by laws of attraction and repulsion among charged particles. Each particle is represented by a vector of real values. The solution of the problem of interest is then obtained by mapping these real-valued vectors to the feasible solution domain. Particles representing better solutions achieve higher level of charge, which consequently produces greater impact on other particles. The search process is performed by iterating the particle movement, induced by charges. Through implementing the methods, two key aspects are managed: 1) the classification quality obtained after applying the optimization method and 2) the efficiency of the proposed methods from the perspective of time and space resources. All methods are equiped with problem-specific local search procedures which tend to increase the solution quality.

The benefit of applying feature selection for the classification process is twofold. Firstly, the elimination of unnecessary features decreases the data set noise, which degrades the quality of the classification model. Secondly, the problem dimension is decreased, thus the efficiency is increased. Feature selection problem is very efficiently solved by the proposed method. The classification quality is in the majority of cases (instances) improved relative to the methods from literature. For some of the instances, computational times are up to several hundred times smaller than those of the competing methods.

Feature weighting and parameter selection problem share similar underlying solution representation, based on the vectors of real values. Since the representation of charged particles is based on the same underlying domain, the transition from the particle to the solution domain behaves *smoothly*. The quality of the method for

feature weighting is demonstrated through nearest neighbors classification model. The testing of the method is conducted on different collection of instances, and after that, the comparison to several methods from literature is made. In the majority of cases, the proposed method outperformed the comparison methods.

The parameter selection, in classification, has a great impact on the classification quality. The proposed method for parameter selection is applied on the support vector machihe, which has a complex parametric structure when the number of parameters and the size of their domains is in question. By using heuristic initialization procedure, the detection of high quality regions for parameter combinations is accelerated. Exhaustive tests are performed on various instances in terms of their dimension and feature structure: homogenous and heterogeneous. Single kernel learning is adopted for homogenous, and multiple kernel learning for heterogeneous instances. The comparison with methods from literature showed superiority of the proposed method when single and multiple kernel learning based on radial basis function is considered. The method shows to be competitive in other cases.

All proposed methods improved the classification quality. Because of the way, the problem is being solved, all three methods can be generalized and applied to a wide class of classification models and/or classification problem.

Keywords: classification, electromagnetism, data mining, metaheuristics, machine learning, optimization
Scientific field: Computer science
Scientific discipline: Optimization
UDC number: [[004.832.2+004.85]:519.863]:537(043.3)

Предговор

У овом раду се разматрају начини за побољшавање класификације података кроз решавање три оптимизациона проблема: проблема подешавања параметара класификатора, проблема одабира атрибута и проблема подешавања тежина атрибута. За сваки од три поменута проблема је развијена метахеуристичка метода заснована на електромагнетизму. Рад се састоји од пет поглавља: уводног, у којем су уведени основни појмови из домена класификације и оптимизације метахеуристикама као и преглед релевантне литературе, затим три поглавља која су посвећена сваком од наведених проблема и поглавља у којем су изложени закључци и преглед научних доприноса.

Дугујем велику захвалност свом ментору, др Владимиру Филиповићу, на дугогодишњој подршци и разумевању. Захваљујем се члановима комисије др Душану Тошићу, др Вељку Милутиновићу, др Ненаду Митићу и др Саши Малкову на корисним коментарима и сугестијама за унапређење овог рада. Захвалност за подршку у писању рада дугујем и др Драгану Матићу са Природно математичког факултета у Бањалуци, као и Ђорђу Стакићу са Математичког факултета у Београду. Захваљујем се и другим колегама са Математичком факултета и Математичког института који су ми директно или индиректно помогли у процесу академског усавршавања.

Велику захвалност дугујем и Весни Поповић која ми је пружила подршку и стрпљење кад год је то било потребно. Захваљујем се својој породици, Обренки, Алексеју, Мирку и Љубомиру, на васпитању и усмеравању које су ми пружили и тиме омогућили да остварим своје академске, али и друге циљеве.

Садржај

Поглавље 1

Увод

Проблем класификације представља један од кључних проблема у области истраживања података и машинског учења. Методе за класификацију налазе широке примене као главни или помоћни механизми у системима за подршку у одлучивању, обради сигнала, медицинској дијагностици, обради мултимедијалних садржаја итд. С обзиром на практични, али и теоријски значај класификације, развијен је велики број метода (класификатора) које се баве овим проблемом. Неке од често примењиваних метода класификације су: метода подржавајућих вектора, метода најближих суседа, класификација коришћењем дрвета одлучивања, вештачке неуронске мреже и др.

Класификатори представљају надгледану технику учења, што значи да се у фази учења класификатор снабдева улазним вредностима и очекиваним излазним вредностима, односно очекиваним класама. Током процеса учења класификатора, наилази се на различите проблеме који су везани за квалитативне и/или квантитативне карактеристике улазних и излазних података, или стање параметара класификатора. Један од проблема везаних за квалитативне и квантитативне карактеристике улазних података је тзв. проблем одабира атрибута. Нека је дат скуп од N атрибута. С обзиром да сваки атрибут може да буде укључен или искључен из скупа разматраних атрибута, постоји $2^N - 1$ различитих начина да се одабере непразан подскуп скупа свих атрибута, односно подскуп атрибута који ће учествовати у процесу класификације. Одабир "адекватних" атрибута има кључни утицај, не само на квалитет, већ и на ефикасност класификације, јер димензија употребљеног подскупа атрибута утиче на дужину времена извршавања и

количину употребљеног меморијског простора. Сродан проблем, али на реалном домену, представља проблем одређивања тежина атрибута, где се тежина интерпретира као значај атрибута. За разлику од проблема одабира атрибута, код овог проблема атрибут не мора да буде само укључен или искључен, већ може да буде укључен са неким степеном значајности. У неким случајевима се дешава да, и поред адекватног одабира атрибута или њихових тежина, квалитет класификације није на задовољавајућем нивоу. Узрок овог проблема може бити лош одабир параметара методе за класификацију. С обзиром да се параметри обично претражују на домену реалних вредности, традиционалне технике за решавање проблема подешавања параметара, попут претраге мреже (енг. grid search), не успевају да произведу задовољавајуће резултате када је број ових параметара велики.

Предмет овог рада је примена метахеуристичке методе засноване на електромагнетизму у решавању горе поменутих проблема: одабира атрибута, подешавања тежина атрибута и подешавања параметара класификатора. Метахеуристика заснована на електромагнетизму (ЕМ) је популациона оптимизациона техника за комбинаторну и глобалну оптимизацију. Инспирисана је законитостима привлачења и одбијања наелектрисаних честица. Популација се састоји из тзв. ЕМ тачака (честица) где свака тачка представља једно потенцијално решење посматраног проблема. Тачке које дају боља решења добијају веће наелектрисање и на тај начин утичу већим интензитетом на остале ЕМ тачке. ЕМ тачка се интерно представља као вектор реалних вредности, што показује велики потенцијал у решавању проблема који се представљају у реалном домену. Оваква репрезентација се такође лако прилагођава и проблемима из дискретног домена. Метахеуристика заснована на електромагнетизму захтева мали број контролних параметара, што је чини једноставном за подешавање и погодном за примену на различитим класама проблема.

У раду се испитује у којој мери ЕМ метода, примењена на проблеме одабира атрибута, подешавања тежина атрибута и подешавања параметара класификатора, доприноси побољшању тачности класификације и/или умањењу броја атрибута који се при класификацији користе у односу на резултате из литературе. Предложене ЕМ методе се могу прилагодити за употребу у произвољном класификационом моделу, и применити у решавању било ког

класификационог проблема.

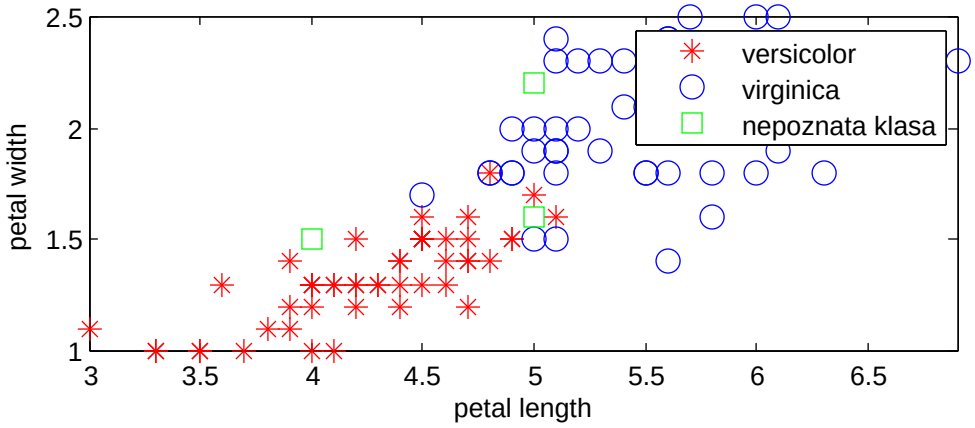
Рад се састоји из пет поглавља, а свако поглавље из већег броја секција. У првом поглављу се излажу основни појмови и концепти који ће се користити у даљем излагању: проблем класификације са свим релевантним аспектима, укључујући и проблеме подешавања параметара класификатора, одабира атрибута, и подешавања тежина атрибута; преглед метахеуристичких техника оптимизације; оптимизационе технике засноване на електромагнетизму; и преглед претходних резултата примене метахеуристичких оптимизационих техника у класификацији. У другом поглављу се детаљно разматра проблем подешавања параметара класификатора на примеру методе подржавајућих вектора. Потом, у трећем и четвртом поглављу су изложене EM методе за решавање проблема одабира атрибута и проблема подешавања њихових тежина. Последње, пето поглавље, садржи закључак и преглед научних доприноса. Засебну целину представља списак литературе коришћене у раду, и налази се на самом крају.

1.1 Класификација - основни концепти

Класификација се бави проблемом додељивања класе (категорије) неком објекту, при чему је број могућих класа коначан и унапред познат. Следећи пример илуструје проблем класификације у којој су могуће две класе.

Пример 1. *Пример је заснован на скупу података под називом Ирис. Реч је о структуираном скупу података о биљци (цвету) Ирис који се често користи као тест проблем за потребе класификације у литератури. Ирис скуп података се може преузети са Репозиторијума за машинско учење UCI [BL13]. Подаци су подељени у три категорије које представљају тип Ирис цвета:* iris setosa, iris versicolor *и* iris virginica. *За сваку од категорија постоји по 50 података, а сваки податак има следеће информације: дужина чашице (енг. sepal length), ширина чашице (енг. sepal width), дужина латице (енг. petal length) и ширина латице (енг. petal width). На Слици 1.1 је приказан подскуп скупа ових података. Хоризонтална оса одговара дужини латице, а вертикална њеној ширини. Због прегледности визуелне илустрације, преостале две информације, о дужини и ширини чашице, су*

изостављене. Додатно поједностављење је направљено и по питању броја класа тиме што су изостављени подаци за тип iris setosa.



Слика 1.1: Проблем бинарне класификације - Ирис

Звездицама су представљени подаци који одговарају типу iris versicolor*, док су подаци типа* iris virginica *означени круговима. Може се приметити да постоји одређена геометријска правилност по питању груписања два различита типа, наиме,* iris versicolor *је претежно распоређен у доњем левом углу где су ниже вредности оба посматрана својства, док је други тип Ириса претежно распоређен у горњем десном углу. Ова правилност упућује на закључак да се може поставити интуитивна граница између две класе цветова. Неформално гледано, одређивање правилности, по којој се подаци могу разврстати у класе, управо представља класификацију. Уколико би то правило било представљено правом линијом, оно не би било у могућности савршено да разграничи све податке, јер би се неки подаци налазили за погрешне стране праве. Са друге стране, употребом нпр. полиномске функције довољно великог степена, било би могуће разграничити податке у потпуности. Међутим, поставља се питање: да ли би таква функција успешно класификовала нове податке, који се не налазе у познатом скупу? Поред података који одговарају познатим класама цветова, на слици су приказани и квадрати који се односе на податке чија класа није позната, односно на нове податке који имају све информације осим класе. Интуитивно би било класификовати квадрат ближи доњем левом углу као звездицу, а други ближи горњем десном углу као круг. Поставља се дилема по питању класе квадрата који се налази ближе маргини између две области. Већ и код оваквог*

једноставног примера се могу уочити одређени изазови, што сугерише да је решавање класификационих проблема врло тежак проблем, посебно када је број података, њихових својстава и класа велики.

У претходном примеру су се помињали појмови попут: тип цвета, дужина латице цвета, правило раздвајања и др. Уопштења свих помињаних релевантних појмова су дата на следећи начин. Скуп података са познатим типовима цветова који се користе у процесу класификације представља *тренинг податке. Тренинг подаци* понекад представљају само подскуп скупа свих познатих података. Мотивација за употребу само подскупа скупа свих података ће бити описана у Секцији 1.3.2. Дужина и ширина латице представљају *атрибут* податка, док се типови цветова iris versicolor и iris irginica називају класама податка. Заједнички, скуп атрибута и класа формира један *податак.* Такође је битно разграничити појам податка, који укључује све информације (атрибути + класа), од податка за који није позната класа. Из тог разлога ће се други појам називати *вектор атрибута* у даљем тексту. Правило које разграничава податке једне и друге класе, о којем је било речи (линеарна функција, полиномијална итд.), се назива *класификациона функција*, а функција додељивања класе се такође назива и *функција одлучивања. Функција одлучивања* је, обично, врло уско повезана са *класификационом функцијом.* У Примеру 1 је та веза заснована на провери са које стране простора подељеног правом или полиномском функцијом се непознати податак налази. Поред ових термина, у даљем излагању биће по потреби уведени још неки термини везани за класификацију.

У литератури је проблем класификације и метода за његово решавање детаљно изучаван, а само неки од детаљних и свеобухватних ресурса су [MST94], [BN06], [DHS12].

Следи општа дефиниција проблема тзв. бинарне класификације, односно класификације у којој постоје само две могуће класе. У даљем тексту су дате и дефиниције проблема класификације прилагођене конкретним класификаторима.

Дефиниција 1. *Нека је дат скуп тренинг података D_{tr} који се састоји од N_{tr} уређених парова облика $(\mathbf{x}^{(i)}, y^{(i)}), i = 1, ..., N_{tr}$, где је $\mathbf{x}^{(i)} \in \mathbf{R}^N$ N-димезионални вектор атрибута, а $y^{(i)} \in \{-1, 1\}$ одговарајућа класа. У*

литератури се вектори са класом 1 називају и позитивни примери, тј. вектори са позитивном класом, а вектори класе -1, се називају негативни примери. Класификација подразумева формирање класификационе функције $f : \mathbf{R}^N \rightarrow \mathbf{R}$. *На основу класификационе функције се одређује функција одлучивања* $c : \mathbf{R}^N \rightarrow \{-1, 1\}$ *која је у стању да за нови вектор атрибута* $\mathbf{x} \in \mathbf{R}^N$, *који није припадао тренинг скупу, одреди класу* $c(\mathbf{x}) = \hat{y}$, *тако да она буде једнака правој класи датог вектора атрибута* $y = \hat{y}$.

Према [MST94], издвајају се три интерпретације проблема класификације, а самим тим и три различита приступа у његовом решавању: 1) статистички приступ; 2) приступ заснован на машинском учењу; 3) и приступ заснован на вештачким неуронским мрежама.

Класичан статистички приступ је заснован на Бајесовом правилу одлучивања, а нешто модернији приступи користе и додатна побољшања, као нпр. мешовите расподеле атрибута. Заједничко за све статистичке приступе је да се класификација не врши директно, већ имплицитно, одређивањем вероватноћа да посматрани податак припада свакој од могућих класа.

У машинском учењу се проблем класификације своди на одређивање аутоматизованих процедура које су у стању да науче да класификују "конзумацијом" довољног броја тренинг података. Овај приступ је обично у потпуности вођен подацима и аутоматизован, те не захтева никакве додатне интервенција човека. Проблем је што количина података потребних за учење класификатора може бити велика. Највећи број метода у овом приступу је заснован на дрветима одлучивања.

Вештачке неуронске мреже представљају комбинацију претходна два приступа и засноване су на структуралној и функционалној имитацији људског мозга. С обзиром да спада у групу универзалних апроксимационих метода, проблем класификације, из перспективе вештачке неуронске мреже, је једноставно постављен као проблем учења класификационе функције. Један од проблема овог приступа је потпуно одсуство транспарентности класификационог модела према кориснику, што не важи за претходна два приступа.

Секција која следи приказује неке концепте Бајесове теорије одлучивања који мотивишу употребу класификационих метода у случају две класе (у

[DHS12] се може наћи детаљан преглед Бајесове теорије одлучивања). Проширења на случај више класа се неће разматрати у статистичком смислу већ само из опште перспективе. У питању су различити начини свођења проблема вишекласне класификације на вишеструке проблеме бинарне класификације и о томе ће бити речи након Секције 1.1.1.

1.1.1 Бајесова теорија одлучивања

Бајесова теорија одлучивања представља основни статистички апарат у решавању проблема класификације. Класификација, тачније доношење одлуке о припадности класи се спроводи оцењивањем вредности вероватноћа за сваку од класа. Нека се разматра проблем бинарне класификације (Дефиниција 1) вектора атрибута означеног са $\mathbf{x} = (x_1, ..., x_N)$, и нека је ω_1 догађај када је припадајућа класа -1, а ω_2 догађај када је класа 1. Нека су $P(\omega_1)$ и $P(\omega_2)$ редом вероватноће које одговарају реализацијама првог и другог догађаја. Када не постоје други могући догађаји, онда важи да је $P(\omega_1) + P(\omega_1) = 1$. Ако не постоје никакве додатне информације, тј. ако су познате само безусловне вероватноће $P(\omega_1)$ и $P(\omega_2)$, логичан избор за функцију одлучивања је:

$$c(\mathbf{x}) = \begin{cases} -1, & P(\omega_1) \geq P(\omega_2) \\ 1, & P(\omega_1) < P(\omega_2) \end{cases} \tag{1.1}$$

Горе приказани класификатор је смислен у случају да не постоје никакве информације о вектору атрибута који се класификује. Са друге стране, може се приметити да ће атрибути зависити од његове класе, тј. да уколико знамо класу неког податка, можемо из тога закључити нешто о његовим атрибутима. Зарад поједностављења нотације, уместо разматрања случајева ω_1 и ω_2 користиће се само ознака за догађај ω_j. Вектор атрибута се може посматрати као вектор случајних променљивих, а $p(\mathbf{x}|\omega_j)$ као густина условне вероватноће. С обзиром да важи $p(\omega_j, \mathbf{x}) = p(\mathbf{x}|\omega_j)P(\omega_j) = P(\omega_j|\mathbf{x})p(\mathbf{x})$, долазимо до познатог резултата, тзв. Бајесове формуле:

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)P(\omega_j)}{p(x)} \tag{1.2}$$

Будући да постоје два могућа догађаја, важи да је $p(\mathbf{x}) = p(\mathbf{x}|\omega_1)P(\omega_1) + p(\mathbf{x}|\omega_2)P(\omega_2)$, па се уврштавањем у (1.2) добија:

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)P(\omega_j)}{p(\mathbf{x}|\omega_1)P(\omega_1) + p(\mathbf{x}|\omega_2)P(\omega_2)} \tag{1.3}$$

У складу са (1.3), претходна функција одлучивања (1.1) се може унапредити тако да узима у обзир и информације из вектора атрибута. Следећа формула се зове Бајесово правило одлучивања:

$$c(\mathbf{x}) = \begin{cases} -1, & P(\omega_1|\mathbf{x}) \geq P(\omega_2|\mathbf{x}) \\ 1, & P(\omega_1|\mathbf{x}) < P(\omega_2|\mathbf{x}) \end{cases} \tag{1.4}$$

Може се показати да Бајесово правило одлучивања минимизује просечну грешку функције одлучивања. Наиме, у складу са датим правилом одлучивања, вероватноћа грешке је дата са:

$$P(\text{greška}|\mathbf{x}) = \begin{cases} P(\omega_1|\mathbf{x}), & \text{ako je odabrana klasa 1} \\ P(\omega_2|\mathbf{x}), & \text{ako je odabrana klasa -1} \end{cases} \tag{1.5}$$

За свако дато $\mathbf{x}$ се може минимизовати грешка тако што се примени Бајесово правило одлучивања. Просечна грешка је представљена изразом:

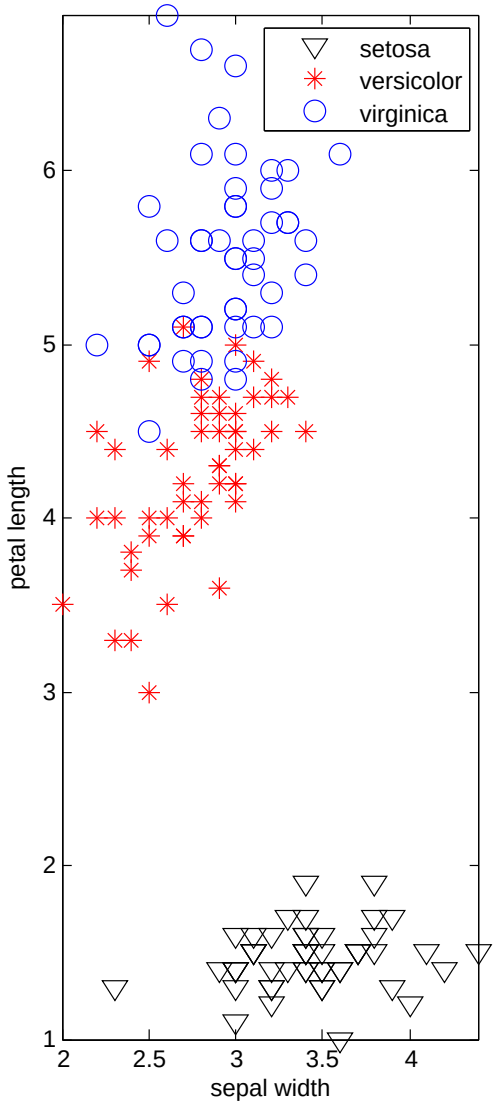
$$P(\text{greška}) = \int_{-\infty}^{\infty} P(\text{greška}, \mathbf{x}) d\mathbf{x} \tag{1.6}$$

Из (1.6) се може закључити да, ако је за свако појединачно $\mathbf{x}$ грешка најмања могућа, онда ће и дати интеграл бити најмањи могућ. Будући да се Бајесовим правилом одлучивања минимизује свака појединачна грешка, закључујемо да ће и интеграл бити најмањи могућ, а самим тим и просечна грешка одлучивања (класификације).

1.1.2 Вишекласни класификациони проблеми

У Примеру 1 је поменуто да је оригинални скуп података цвета Ирис сачињен од података о цветовима три типа. На Слици 1.2 је приказан потпун скуп података са додатим цветовима типа *iris setosa*. Зарад "илустративнијег" распореда података, на хоризонталној оси је сада вредност дужине чашице, а на вертикалној ширина латице. Евидентно је да се не може повући линеарна граница која би раздвајала сва три региона. Међутим, граница је ипак интуитивна, и може се описати неким сложенијим правилом раздвајања. Пре него што се настави

са приказом неких познатих метода за решавање овог проблема, биће уведена формална дефиниција проблема вишекласне класификације.



Слика 1.2: Проблем вишекласне класификације - Ирис

Дефиниција 2. *Нека је дат скуп тренинг података D_{tr} који се састоји од N_{tr} уређених парова облика $(\mathbf{x}^{(i)}, y^{(i)}), i = 1, ..., N_{tr}$, где је $\mathbf{x}^{(i)} \in \mathbf{R}^N$ N-димезионални вектор атрибута, а $y^{(i)} \in \{1, ..., N_c\}$ одговарајућа класа. N_c је укупан број класа, а класе су, без губитка општости, означене природним бројевима од 1 до N_c. Класификација подразумева формирање класификационе функције $f : \mathbf{R}^N \to \mathbf{R}$. На основу класификационе функције се одређује функција одлучивања $c : \mathbf{R}^N \to \{1, ..., N_c\}$, односно функција која је у стању да за нови вектор*

атрибута $\mathbf{x} \in \mathbf{R}^N$*, који није припадао тренинг скупу, одреди класу* $c(\mathbf{x}) = \hat{y}$*, такву да важи* $\hat{y} = y$*.*

Ако се претходна дефиниција упореди са Дефиницијом 1, може се приметити да је једина разлика у кардиналности скупа класа, која сада може бити и већа од 2. Дефиниција вишекласне класификације је, дакле, уопштење бинарне класификације. Постоји врло сродан проблем који се зове вишезначна класификација (енг. multilabel classification), где је циљ доделити векторима атрибута једну или више класа. Код вишекласне класификације (енг. mutliclass classification) сваки објекат припада тачно једној класи.

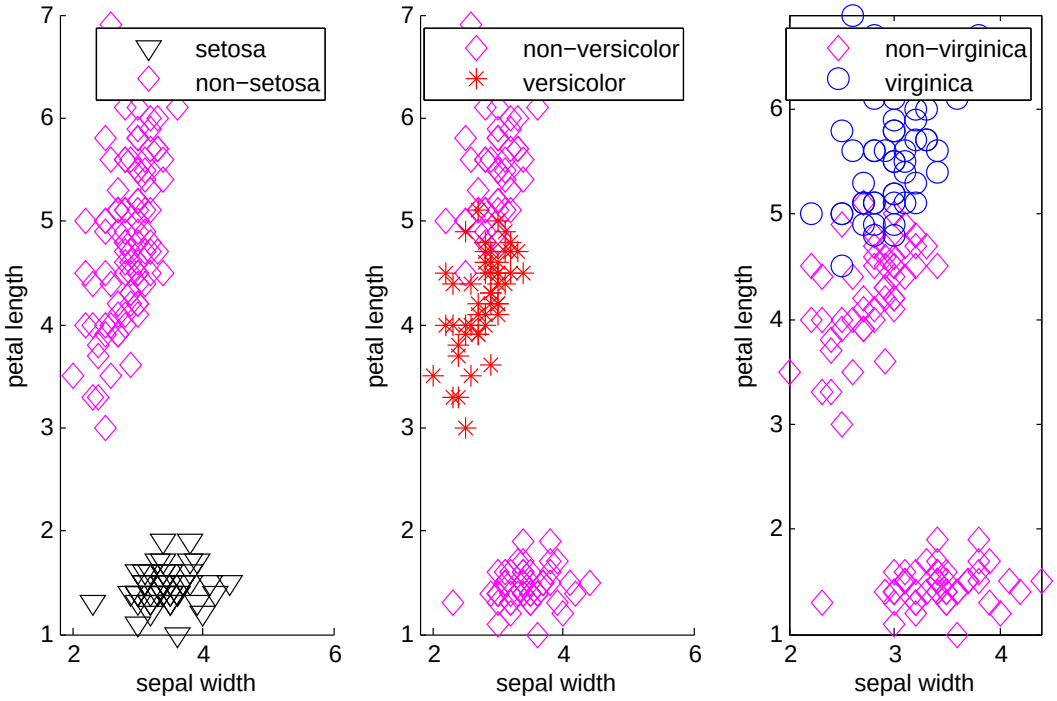
Овде су приказане две основне технике решавања проблема вишекласне класификације. Обе технике су засноване на свођењу проблема вишекласне класификације на већи број проблема бинарне класификације. У литератури су ова два приступа највише коришћена. Постоје и други предложени приступи, али они нису размотрени у овом раду.

Један-против-Свих (енг. One-vs-All) приступ је заснован на следећој идеји. Ако постоји N_c класа, прави се N_c независних бинарних класификатора, где се i-ти класификатор користи за раздвајање i-те класе од свих осталих класа, тј. i-ти класификатор класификује векторе класе i као позитивне примере, а све остале као негативне. Ако се са f_i означи класификациона функција i-тог класификатора, а са $\mathbf{x}$ вектор атрибута који је потребно класификовати, онда се функција одлучивања за проблем вишекласне класификације добија као:

$$c(\mathbf{x}) = arg\max_i f_i(\mathbf{x}),\ i = 1, ..., N_c \tag{1.7}$$

У изразу (1.7), $arg\max_i f_i(\mathbf{x})$ значи да се врши максимизација израза $f_i(\mathbf{x})$, али да се потом као вредност враћа i за које је максимум достигнут. Слика 1.3 илуструје овај приступ. За пример са 3 типа Ирис цветова се формирају 3 класификационе функције. На слици те функције нису приказане, али је јасно да постоји интуитивна граница у сваком од 3 случаја. Новом вектору атрибута $\mathbf{x}$ се додељује класа оног класификатора који постигне највећу моћ дискриминације према осталим класама. На Слици 1.3 се такође види да у случају када се подаци поделе на групе *versicolor* и *non-versicolor*, граница између две новоформиране класе података није линеарна. У таквој

ситуацији би се могла применити нелинеарна класификациона функција, или трансформација скупа улазних података која би омогућила да подаци могу бити раздвојени линеарном функцијом. О овим, и о другим проблемима ће бити речи када се буду разматрали конкретни класификатори.



Слика 1.3: Један-против-Свих приступ

Сви-против-Свих (енг. All-vs-All) се заснива се на формирању $N_c * (N_c - 1)$ класификатора, за сваки пар класа по један. Нека је са f_{ij} означена класификациона функција која све податке класе i класификује као позитивне примере, а све податке класе j као негативне. Примери осталих класа се не разматрају у формирању класификационе функције f_{ij}. Примећује се да је $f_{ij} = -f_{ji}$. Функција одлучивања се рачуна путем следећег израза:

$$c(\mathbf{x}) = arg\max_i \left(\sum_{j \neq i} f_{ij}(\mathbf{x}) \right), \ i = 1, ..., N_c, \ j = 1, ..., N_c \tag{1.8}$$

У литератури се често користе оба приказана приступа, и многобројна емпиријска истраживања су показала да се ова два метода, иако једноставна, добро показују у пракси. Временска ефикасност зависи од величине тренинг скупа података, броја класа и класификатора који се користи. Први приступ користи

само N_c класификатора, али је тренинг скуп сачињен од целог почетног скупа, јер се разматрају све класе. У другом приступу, Сви-против-Свих се формира $N_c * (N_c - 1)$ класификатора. Међутим, сложеност процеса учења сваког од њих је мања, јер је и број података по једном класификатору мањи. Ово је посебно приметно у случајевима када је број класа велик, а расподела класа равномерна.

1.1.3 Подела метода за класификацију

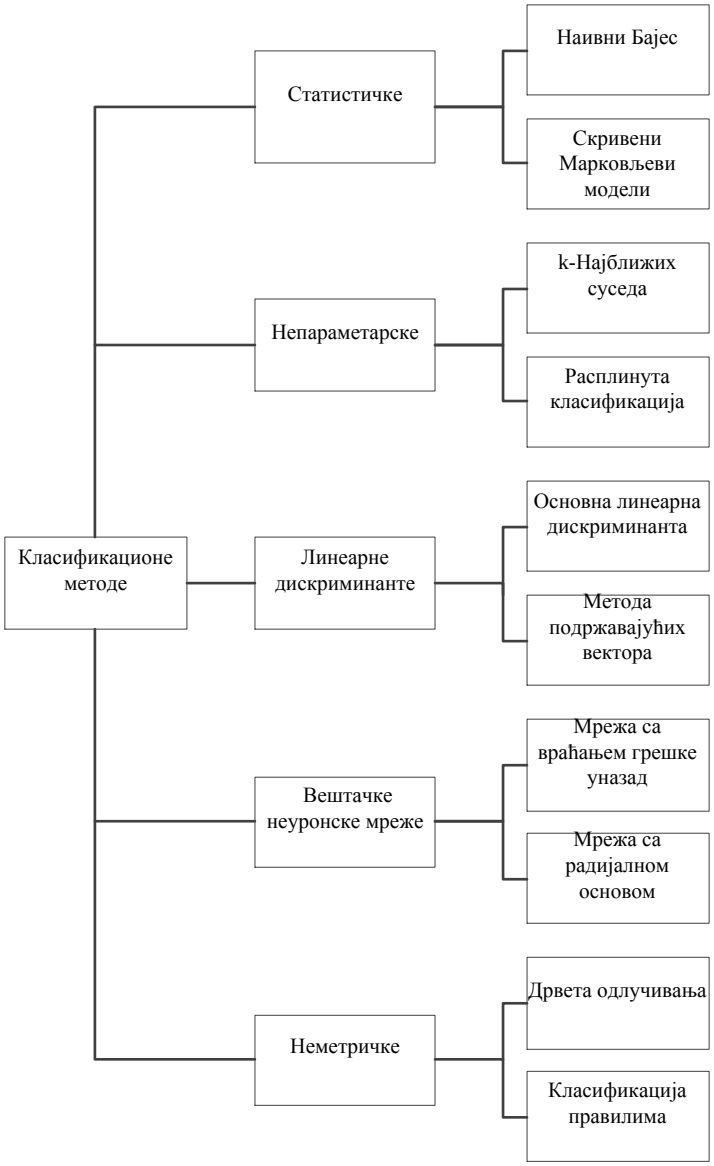
Ова подела је предложена у [DHS12]. На Слици 1.4 је дат њен шематски приказ. За сваку групу метода наведена су по два припадајућа елемента.

Када су у питању методе статистичке класификације, у Секцији 1.1.1 су приказани неки аспекти Бајесове теорије одлучивања. Они су подразумевали да је вероватносна расподела класа унапред позната. Када су такве информације доступне, Бајесов класификатор се може користити као *златни стандард* за поређење са другим методама. У пракси је, међутим, чешћа ситуација да вероватносна расподела класа није доступна, али и тада је могуће применити одређене статистичке технике. На пример, ако је познат тип расподеле (функционална форма), онда се параметри расподеле могу апроксимирати техникама попут методе максималне веродостојности, или другим егзактним и неегзактним оптимизационим техникама.

Када ни функционална форма расподеле није позната, тј. када постоји потпуно одсуство било каквих информација о вероватносној структури класификационог проблема, могу се применити непараметарске технике. Једна од познатијих непараметарских метода је метода k-најближих суседа о којој ће бити речи у наредној секцији овог поглавља.

Следећа група метода се зову линеарне дискриминантне функције. Код њих се користи линеарна функционална форма као класификациони модел. Потом се врши оптимизација параметара те функционалне форме. У наредној секцији овог поглавља ће бити представљене две методе из ове групе, основна линеарна дискриминантна функција и метода подражавајућих вектора, обе на примерима бинарне класификације.

Вештачке неуронске мреже представљају засебну групу метода код којих је функција учења (класификације) нелинеарна. Вештачка неуронска мрежа је



Слика 1.4: Подела класификационих метода

"моћна" апроксимативна метода способна да "научи" функције које у основи имају висок степен нелинеарности. Детаљнији преглед ове технике излази из оквира овог рада.

Неметричке методе немају јасну функционалну форму, нити статистичке елементе. Оне се најбоље могу описати као скупови логичких правила. У наредној секцији је, поред осталих метода, описана и општа структура алгоритма заснованог на дрвету одлучивања.

1.2 Преглед неких метода за класификацију

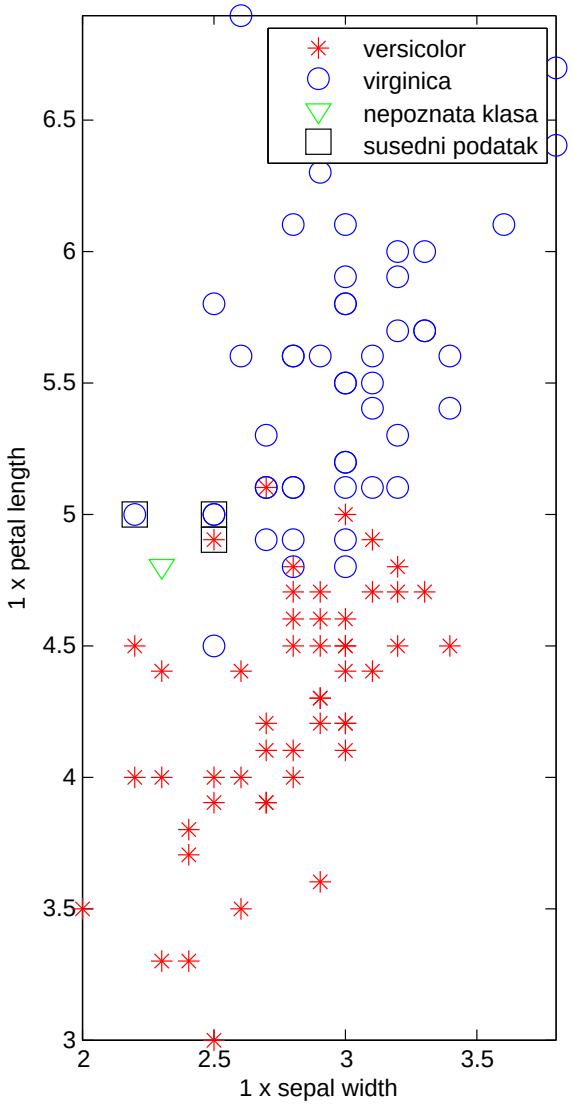
У овој секцији се излажу неке од класификационих техника које припадају различитим класификационим групама из поделе приказане на Слици 1.4.

1.2.1 Метода *k*-најближих суседа (k-NN)

Метода k-најближих суседа (енг. k-nearest neighbors - k-NN) представља непараметарску класификациону технику која класификује задати вектор атрибута на основу скупа од k најближих суседа тог вектора. При том се под најближим суседима мисли на податке из тренинг скупа података који имају највиши степен сличности вектора атрибута са посматраним вектором атрибута. Након што се одреде k таквих података, посматрани вектор атрибута се класификује у складу са класом која преовладава у скупу суседа. На Слици 1.5 је приказана илустрација k-NN методе за $k = 3$. У датом примеру, непознатом цвету би се долелила класа iris virginica. Када се примењује у решавању бинарних класификационих проблема, вредности параметра k обично узимају непарне вредности. Ово омогућава да се увек донесе једнозначна одлука по питању класе.

Формална дефиниција класификационог проблема у контексту методе најближих суседа гласи:

Дефиниција 3. *Нека је дат тренинг скуп података D_{tr} сачињен од N_{tr} парова облика $(\mathbf{x}^{(i)}, y^{(i)}), i = 1, ..., N_{tr}$, где је $\mathbf{x}^{(i)} \in \mathbf{R}^N$ N-димензиони вектор атрибута, а $y^{(i)} \in \{1, 2, ..., N_c\}$ одговарајућа класа. За нови вектор атрибута $\mathbf{x}$, k-NN проналази скуп сачињен од k тренинг података $\{(\mathbf{x}^{(i_1)}, y^{(i_1)}), ..., (\mathbf{x}^{(i_k)}, y^{(i_k)})\}$, који имају најмање вредности функције удаљености $dist : \mathbf{R}^N \times \mathbf{R}^N \to \mathbf{R}$ у односу на вектор атрибута. Функција удаљености (различитости) се рачуна за сваки пар формиран од свих тренинг вектора атрибута са једне стране и новог вектора атрибута са друге: $dist(\mathbf{x}, \mathbf{x}^{(i)})$, $i = 1, ..., N_{tr}$. Након што се утврде све вредности функције удаљености, врши се уређивање по тој вредности у растућем поретку и након тога првих k тренинг вектора атрибута се бира у скуп најближих суседа. Функција одређивања класе новог вектора атрибута се потом добија помоћу следеће формуле:*



Слика 1.5: Пример примене 3-NN методе

$$c(\mathbf{x}) = arg\max_{m} \left(\sum_{j=1}^{k} \mathbb{1}\{y^{(i_j)} = m\} \right), \; m = 1, ..., N_c \tag{1.9}$$

Изразом $\mathbb{1}\{y^{(i_j)} = m\}$ је представљена индикаторска функција која узима вредност 1 ако је $y^{(i_j)} = m$, а у супротном узима вредност 0. Као што се може видети из приказаног, k-NN метода је релативно једноставна за имплементацију и за разумевање. Фаза учења класификатора не постоји у класичном смислу. Све релевантне операције, које чине класификациони модел, се извршавају тек када започне примена класификатора над конкретним вектором атрибута. Ово својство има за последицу да временска ефикасност зависи од димензије

тренинг скупа података и броја атрибута сваког од података, јер је нпр. сложеност извршавања 1-NN за сваки појединачни тренинг вектор једнака $O(N_{tr}N)$. У циљу побољшавања временске ефикасности, у литератури су предложена многобројна побољшања основног k-NN алгоритма, попут паралелизације претраге најближих суседа, парцијалног рачунања функције удаљености и других.

1.2.2 Линеарна дискриминантна функција

У линеарним дискриминантним моделима се прави претпоставка да је адекватна форма функционалног класификатора линеарна функција. Потом се врши оптимизација функције циља по параметрима те функционалне форме. Функција циља је најчешће представљена грешком класификације на тренинг подацима, што може довести до преприлагођавања класификационе функције тренинг скупу. Последица преприлагођавања је да се класификатор не понаша добро на подацима ван тренинг скупа, тзв. тест подацима.

Општа форма линеарне дискриминативне функције је:

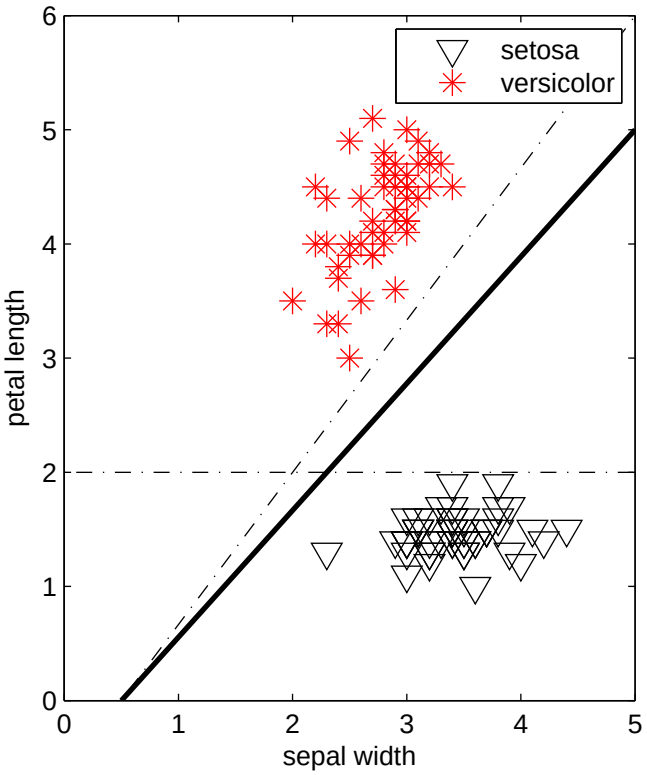
$$f(\mathbf{x}) = w_0 + w_1 x_1 + ... + w_N x_N,\ w_i \in \mathbf{R},\ i = 1, ..., N \tag{1.10}$$

Где су $w_i,\ i = 1, ..., N$ тежине додељене атрибутима вектора, док је w_0 слободни члан или тежински праг (енг. threshold weight). Функција f дефинише хиперраван (маргину) која раздваја области припадности две класе. Уобичајена придружена функција одлучивања за (1.10) је дата са:

$$c(\mathbf{x}) = \begin{cases} -1, & f(\mathbf{x}) < 0 \\ 1, & f(\mathbf{x}) \geq 0 \end{cases} \tag{1.11}$$

Размотримо сада случај када подаци имају својство *линеарне раздвојивости* на тренинг скупу. Ово значи да постоји такав вектор коефицијената $\mathbf{w} = (w_0, ..., w_N)$ за који важи да, након што се уврсти у функцију f, придружена функција одлучивања даје коректне класификације за све тренинг податке. Резултујући вектор коефицијената није јединствен као што се може видети на Слици 1.6. Иако све три раздвајајуће хиперравни коректно класификују све тренинг векторе атрибута, појачана хиперраван испуњава и додатни критеријум јер минимизује суму растојања тренинг вектора од раздвајајуће хиперравни.

Испоставља се да је ово својство кључно за класификаторе који припадају групи класификатора у којима се максимизује маргина (енг. maximal margin classifiers). Најпознатији у овој групи су метода подржавајућих вектора и AdaBoost метод.



Слика 1.6: Примери раздвајајућих хиперравни на Ирис скупу са две класе

У основном моделу линеарне дискриминативне функције додатни услов о максималности маргине не постоји. Проблем оптимизације коефицијената стога разматра само услов линеарне раздвојивости, и решења за овај проблем се могу наћи употребом основних нумеричких алгоритама попут градијентног спуста, Њутнове методе и др.

1.2.3 Класификација методом подржавајућих вектора (SVM)

Метода подржавајућих вектора (енг. Support vector machine - SVM) је техника машинског учења са широким применама. SVM припада групи линеарно дискриминативних метода у којима постоји и додатни услов о

максималности маргине. Два главна домена примене су: 1) класификациони проблеми, у којима се врши предвиђање дискретне променљиве и 2) одређивање функционалних форми у проблемима регресије, у којима се врши предвиђање непрекидне променљиве.

Теоријске основе SVM су дате у [Vap95; Vap99]. Мотивација за примену SVM у класификацији потиче од резултата статистичке теорије учења у којој је развијена горња граница грешке генерализације ове методе. Горња граница грешке је минимална када је растојање између вектора и раздвајајуће хиперравни максимално. Важно практично својство горње границе је њена независност од димензије простора атрибута. Међутим, формирање раздвајајуће хиперравни није увек могуће и у таквим ситацијама се каже да простор атрибута није линеарно раздвојив. Ово својство нераздвојивости се може заобићи пресликавањем оригиналног простора атрибута у други простор који поседује својство линеарне раздвојивости. Трансформација простора доводи до повећања димензије проблема, међутим, то не утиче на ефикасност SVM методе, с обзиром на чињеницу да SVM не користи векторе атрибута директно. Уместо директне употребе тренинг вектора, SVM користи функцију сличности између парова вектора. Функција сличности се назива кернел (језгро), и њено својство од високог практичног значаја је да се може израчунати у оригиналном простору атрибута. Стога, било какво повећање димензије простора нема утицаја на сложеност израчунавања SVM алгоритма.

Сада ће бити приказана математичка формулација проблема бинарне класификације прилагођена решавању SVM алгоритмом [Kec01; PK10; CY11]. За класификационе проблеме са више од две могуће класе, поред општих приступа приказаних у Секцији 1.1.2, у [ASS01] је представљен унификовани приступ за класификаторе засноване на маргинама, међу којима је и SVM.

Дефиниција 4. *Нека је дат тренинг скуп података D_{tr} сачињен од парова облика $(\mathbf{x}^{(i)}, y^{(i)}), i = 1, ..., N_{tr}$, где $\mathbf{x}^{(i)} \in \mathbf{R}^N$ представља N-димензионални вектор вредности по сваком од улазних атрибута, а $y^{(i)} \in \{-1, 1\}$ његову припадајућу класу. SVM користи тренинг скуп D_{tr} у циљу проналажења раздвајајуће хиперравни $\mathbf{w}\cdot\mathbf{x}+b = 0$, $\mathbf{w} \in \mathbf{R}^N$, $b \in \mathbf{R}$ која је максимално удаљена од свих тренинг података (вектора) са обе стране. Раздвајајућа хиперраван (маргина) се лако трансформише у функцију одлучивања $c(\mathbf{x}) = \mathrm{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$*

која рачуна класу за дати улазни вектор атрибута.

Стриктно линеарно раздвојиви подаци су врло ретки у пракси, стога постоји потреба за допуштањем одређене грешке по питању раздвојивости података. Мање стриктна варијанта проблема, тзв. класификација базирана на *мекој* маргини (енг. soft margin classifier) је предложена у [CV95]. Уместо постојања стриктне маргине између тренинг вектора различите класе, мека варијанта проблема допушта векторима да се налазе са погрешне стране, али ипак довољно близу маргине: $y^{(i)}(\mathbf{w}\cdot\mathbf{x}^{(i)}+b) \geq 1-\zeta_i,\ i=1,...,N_{tr}$, а ζ_i је ненегативна променљива која представља грешку преласка вектора $\mathbf{x}^{(i)}$ на погрешну страну хиперравни. На овако модификованој дефиницији раздвајајуће хиперравни, оптималне вредности за $\mathbf{w}$ and b могу бити пронађене решавањем следећег оптимизационог проблема:

$$\min\left(\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N_{tr}}\zeta_i\right) \tag{1.12}$$

уз ограничења:

$$y^{(i)}(\mathbf{w}\cdot\mathbf{x}^{(i)}+b) \geq 1-\zeta_i,\quad i=1,...,N_{tr} \tag{1.13}$$

$$\zeta_i \geq 0,\ i=1,...,N_{tr} \tag{1.14}$$

C је регуларизациони (казнени) параметар који контролише утицај грешака прелажења.

SVM користи погодну дуалну репрезентацију у циљу одређивања маргине. Дуална формулација је дата без извођења (у [BGV92] је детаљно описан поступак њеног добијања на основу прималне формулације).

$$\max\left(\sum_{i=1}^{N_{tr}}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N_{tr}} y^{(i)}y^{(j)}\alpha_i\alpha_j\mathbf{x}^{(i)}\mathbf{x}^{(j)}\right) \tag{1.15}$$

уз ограничења:

$$\alpha_i \in [0,C],\quad i=1,...,N_{tr} \tag{1.16}$$

$$\sum_{i=1}^{N_{tr}} \alpha_i y^{(i)} = 0 \tag{1.17}$$

Вредности α_i, $i = 1, ..., N_{tr}$ представљају Лагранжове множиоце, односно коефицијенте, док C у овој формулацији представља њихову горњу границу. Следствено, C контролише свеукупни максимизациони израз прављењем компромиса између максимизације маргине и минимизације грешке. Функција одлучивања за дату формулацију се рачуна као:

$$c(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{N_{tr}} \alpha_i y^{(i)} \mathbf{x}^{(i)} \mathbf{x} + b\right) \tag{1.18}$$

Формулација за решавање проблема који немају ни приближно својство линеарне раздвојивости (у потпуности су нелинеарни), је детаљно размотрена у Секцији 2.1.1.

1.2.4 Класификација помоћу дрвета одлучивања

Дрво одлучивања класификује посматрани вектор атрибута тако што "пропушта" тај вектор кроз унутрашње чворове све до листова. Сваком од унутрашњих чворова је придружено питање које усмерава вектор атрибута на неки од својих чворова потомака. Питање се односи на вредност неког од могућих атрибута. Неопходно је да одговор на питање буде једнозначан, односно да усмерава улазни вектор атрибута на тачно једног потомка. Вектору атрибута, класа бива додељена када достигне неки од листова дрвета. Неке од познатијих метода из ове групе су: ID3, C4.5, C5.0, CART, CHAID и др. Сви ови алгоритми користе исти принцип рекурзивног формирања дрвета под називом Хантов алгоритам (енг. Hunt's algorithm). Општа структура Хантовог алгоритма се састоји из следећа два корака:

1. **Излазак из рекурзије:** Ако су сви тренинг подаци придружени посматраном чвору дрвета у истој класи, прави се лист дрвета и означава том класом;

2. **Рекурзивни корак:** Ако подаци придружени посматраном чвору дрвета немају јединствену класу, бира се атрибут који најефективније раздваја тренинг скуп података на подскупове. Критеријуми ефективности

раздвајања могу бити различите метрике: ентропија, информациони добитак, Гини коефицијент, тачност класификације и др. Након што се изврши одабир "најбољег" атрибута, врши се подела података према могућим вредностима атрибута, или према дискретизованим интервалима у случају атрибута са реалним вредностима или великим бројем могућих дискретних вредности. Коначно, на добијеним подскуповима се примењује целокупни поступак рекурзивно.

Класификатори из ове групе метода су релативно једноставни за имплементацију и врло ефикасни по питању класификовања нових вектора атрибута. Њихов основни неодстатак лежи у чињеници да је реч о "похлепним" алгоритмима претраге, што може довести до незадовољавајуће тачности класификације код одређених класа проблема.

1.3 Остали аспекти класификације

У овој секцији су описани још неки пратећи елементи класификационог проблема. Прво од питања се тиче проблема утврђивања квалитета класификације. Затим су изложене технике за избегавање преприлагођавања класификатора, односно повећања моћи уопштавања класификационог модела. Овде ће бити речи и о проблему подешавања параметара класификатора, који је један од три проблема разматрана у раду. На крају су изложене процедуре препроцесирања података које могу допринети побољшању ефикасности и квалитета класификације. Посебно су од интереса два проблема која се разматрају у раду: проблем одабира атрибута и подешавања њихових тежина.

На Слици 1.7 је приказана проширена шема класификационог процеса. На почетку процеса је прикупљање података и њихово складиштење у структуираном формату. На пример, ако се врши детекција непожељних порука, потребно је прво обрадити поруке електронске поште и додатне информације о њима. Ако се врши класификација тумора на основу слика добијених путем ултразвука или магнетне резонанце, потребно је обрадити слике и из њих "извући" релевантне информације. Након што су подаци припремљени, прелази се у фазу препроцесирања. Препроцесирање подразумева одабир атрибута, подешавање тежина атрибута, решавање

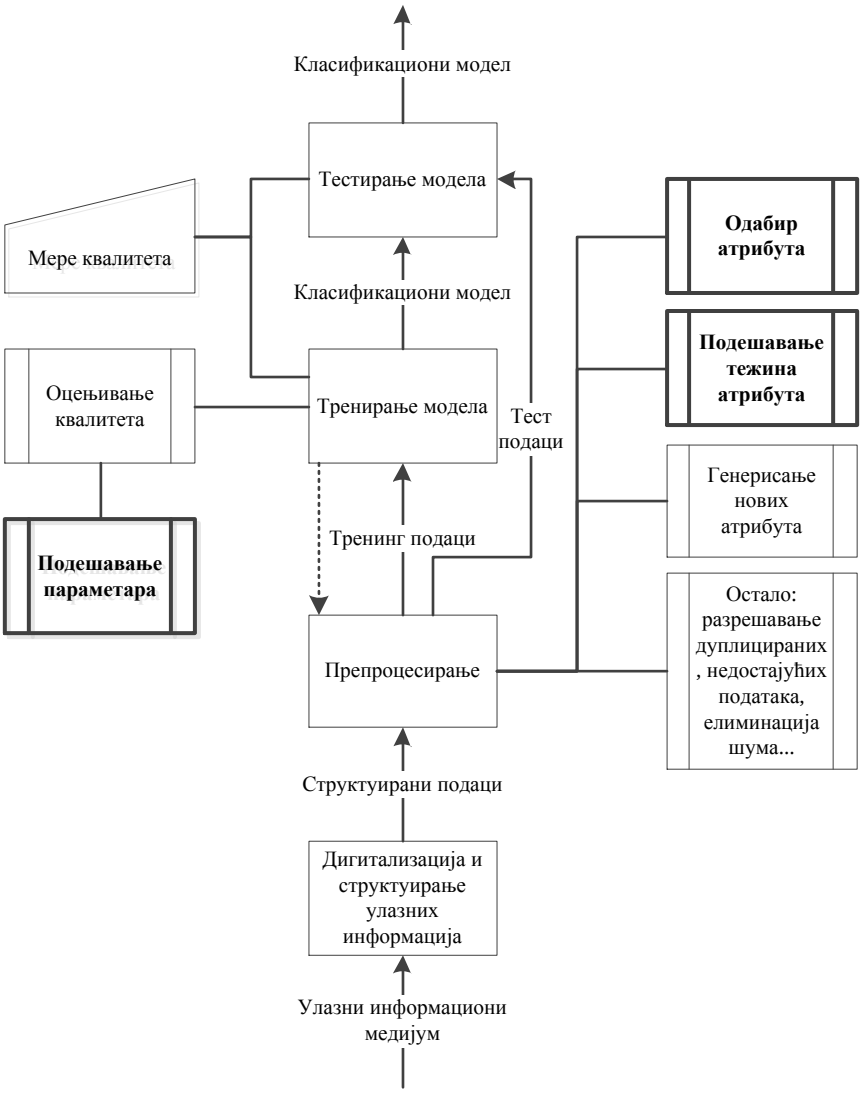
проблема недостајућих вредности, шума итд. После препроцесирања, улази се у фазу формирања класификационог модела. У овом раду се разматра случај када постоји и повратна веза са фазом препроцесирања, јер се одабир атрибута и подешавање њихових тежина извршава у фази тренирања модела. Битан аспекат фазе тренинга је начин оцењивања квалитета класификације на тренинг подацима. Постоје различити приступи оцењивања квалитета који омогућавају висок степен уопштавања класификатора и самим тим могућност његове примене на новим подацима ван тренинг скупа. Неки класификациони модели имају своју интерну параметарску структуру, која, ако се адекватно подеси, може побољшати оцену квалитета у фази тренинга. Ово, касније, уколико је оцена квалитета непристрасна и има довољну моћ генерализације, доводи до побољшања квалитета класификације и на новим подацима. Мера квалитета је функција која се користи за описивање квалитета класификатора, као и за поређење са другим методама. Она се користи и у фази тренинга, као и у фази тестирања класификационог модела. У фази тренинга се оцењује будућа вредност мере квалитета на новим подацима, а у фази тестирања се та будућа вредност мере квалитета проверава.

1.3.1 Мере квалитета

Мера квалитета класификационог модела омогућује његову евалуацију и поређење са другим моделима. Мера квалитета класификације обично представља потенцијал модела да коректно предвиди класу новог податка и/или тренинг података. Матрица конфузије (енг. confusion matrix) представља прегледан и детаљан начин да се тај потенцијал прикаже. Она приказује упоредни однос између броја предвиђених и правих класа за неки скуп вектора атрибута. Претпоставимо да је неки класификатор произвео класификације представљене следећом матрицом конфузије:

Табела 1.1: Матрица конфузије

		Предвиђена класа		
		setosa	versicolor	virginica
Права класа	setosa	43	5	2
	versicolor	5	34	11
	virginica	1	1	48



Слика 1.7: Проширена шема система за класификацију

Редови матрице одговарају правим вредностима класа вектора атрибута, док колоне представљају класе додељене од стране модела. За ред означен називом *versicolor* и колону означену са *virginica*, матрица приказује број вектора атрибута које је модел класификовао као тип *virginica*, а уствари је била реч о цветовима типа *versicolor*. Елементи матрице који се налазе на главној дијагонали приказују коректна предвиђања класификационог модела. У општем случају матрица конфузије има следећу структуру:

Тачност (енг. accuracy) је мера која представља однос укупног броја коректних предвиђања и укупног броја предвиђања. У складу са претходно уведеном нотацијом, може се израчунати као:

Табела 1.2: Матрица конфузије - општа структура

		Предвиђена класа			
		1	2	...	N_c
Права класа	1	n_{11}	n_{12}	...	n_{1N_c}
	2	n_{21}	n_{22}	...	n_{2N_c}
	⋮	⋮	⋮	⋱	⋮
	N_c	n_{N_c1}	n_{N_c2}	...	$n_{N_cN_c}$

$$Acc = \frac{\sum_{i=1}^{N_c} n_{ii}}{\sum_{i=1}^{N_c}\sum_{j=1}^{N_c} n_{ij}} \tag{1.19}$$

Прецизност (eng. precision) је слична мери тачности, али се односи само на једну посматрану класу. Она представља однос тачних позитивних предвиђања (енг. true positives) и укупног број случајева у којима је класификатор предвидео посматрану класу. Рачуна се према формули:

$$P_i = \frac{n_{ii}}{\sum_{j=1}^{N_c} n_{ji}},\ i = 1, ..., N_c \tag{1.20}$$

Одзив (енг. recall) приказује однос коректно предвиђених вектора атрибута неке класе и укупног броја правих појављивања те класе у скупу података:

$$R_i = \frac{n_{ii}}{\sum_{j=1}^{N_c} n_{ij}},\ i = 1, ..., N_c \tag{1.21}$$

F-мера (енг. F-measure) је комбинована мера добијена као хармонијска средина прецизности и одзива:

$$F_i = \frac{2P_iR_i}{P_i + R_i},\ i = 1, ..., N_c \tag{1.22}$$

У Табели 1.3 су приказане вредности ове 4 мере за матрицу конфузије дату у Табели 1.1.

Табела 1.3: Вредности класификационих мера за матрицу конфузије из Табеле 1.1

Класа	P_i	R_i	F_i
i=1	0.88	0.86	0.87
i=2	0.85	0.68	0.76
i=3	0.79	0.96	0.77
Acc:			0.83

1.3.2 Оцене квалитета

Изградња класификатора подразумева две кључне фазе: тренирање класификационог модела и његово тестирање. Тренирање подразумева формирање класификационе функције (класификатора) на основу једног дела података. У фази тестирања, кроз класификатор се пропуштају тест подаци, односно подаци који нису били познати у фази тренинга. На овај начин се симулира права намена класификатора, а то је класификација будућих, тренутно непознатих података. Један од основних проблема у фази тренирања класификатора је оцењивање квалитета класификације изражене претходно приказаним или неким другим мерама. Потребно је да та оцена буде непристрасна и да се понаша добро на тест скупу података. Проблем који се често јавља код класификације је преприлагођавање модела тренинг подацима што узрокује незадовољавајуће понашање класификатора на тест скупу и на будућим непознатим подацима. На пример, уколико бисмо у случају класификације дрветом одлучивања допустили да дрво нарасте док год се не изврши коректна класификација свих тренинг података, постоји велика могућност да би то изазвало преприлагођавање. Евидентно је да је у пракси потребно направити компромис по питању сложености модела. Превише сложени модели могу довести до преприлагођавања, док са друге стране, превише једноставни модели могу довести до потприлагођавања, ситуације у којој модел не научи "довољно" добро циљну класификациону функцију. Решавање проблема оцењивања квалитета се своди на максимизацију "будућег" квалитета класификације, што у пракси значи максимизацију квалитета класификације на тест скупу података. У свим описаним приступима, тест скуп се одваја од тренинг скупа. На крају, када је класификациони модел изграђен, врши се једнократна примена модела над тест скупом што производи коначну меру квалитета изграђеног класификатора. Уобичајене поделе скупа су: 2/3 за тренинг скуп, 1/3 за тест или 1/2 за тренинг и 1/2 за тест. Међутим, тај однос зависи и од типа класификатора. Код методе подржавајућих вектора је довољан релативно мали тренинг скуп, што јој касније омогућава висок степен генерализације на новим подацима. Подаци се бирају на случајан начин, што у теорији, за довољно велике почетне скупове података, омогућава да расподела класа и атрибута буде иста у тренинг и у тест скупу.

Оцена на тренинг скупу података (енг. hold-out estimation) је основни приступ. Оцена квалитета класификације је једноставно једнака оцени квалитета на тренинг скупу. Ово је врло оптимистична претпоставка, и често се показује као пристрасна оцена квалитета. Потребно је направити компромис по питању величине тренинг скупа, јер превелики тренинг скуп може довести до преприлагођавања, а премали до потприлагођавања.

Итеративна подела на тренинг скуп и скуп за проверу (енг. random subsampling) се заснива на вишеструкој случајној подели тренинг скупа података на два дела. Први део се у свакој од тих подела користи за тренирање модела, а други, под називом скуп за проверу, се користи за проверу модела. Коначна оцена квалитета се добија као просечна оцена квалитета над свим скуповима за проверу. Ово има за циљ повећавање способности уопштавања класификатора, али ипак, доводи до других проблема. Очигледни проблем је ефикасност, а други битан недостатак је то што не постоји контрола над бројем појављивања неког податка у скупу за проверу (не постоје ни горња ни доња граница у броју појављивања неког податка у скупу за проверу).

Унакрсна провера (енг. cross-validation) је посебно значајна техника када су полазни скупови података мале или средње величине. Тренинг скуп се дели на k дисјунктних подскупова (скупова за проверу) исте или приближне кардиналности. Устаљене вредности за параметар k су из опсега 2-10. Класа сваког вектора атрибута се одређује "пропуштањем" кроз класификатор који је формиран коришћењем свих компоненти осим оне којој посматрани вектор атрибута припада. Након што се одреди квалитет класификације за све скупове за проверу, укупан квалитет се добија као просек тих вредности. Може се приметити да се овим приступом решава горе поменути проблем са контролом броја појављивања податка у скупу за проверу, јер се сваки податак користи тачно једном за проверу. Међутим, проблем ефикасности и даље је присутан. Гранични сценарио примене унакрсне провере, када је кардиналност скупа за проверу једнака 1, зове се изостави-1-провера (енг. leave-one-out validation - LOO). Иако даје скоро непристрасну оцену квалитета класификације, рачунарски је још интензивнија од унакрсне провере са устаљеним вредностима параметра k.

Бутстреп оцена (енг. bootstrap estimation) за разлику од итеративне

поделе на тренинг и скуп за проверу, и унакрсне провере, користи случајни одабир са враћањем. Ако је тренинг скуп димензије N_{tr}, врши се N_{tr} случајних одабира података са враћањем. Након што се изврше сви случајни одабири, подаци који нису били ниједном одабрани улазе у скуп за проверу, док се преостали користе за тренинг. Процес се потом понавља неколико пута, а укупна оцена квалитета се добија као просечна оцена на свим тако формираним скуповима за проверу. С обзиром да је вероватноћа одабира једног податка $1/N_{tr}$, вероватноћа да податак неће бити одабран ниједанпут у N_{tr} покушаја је $(1 - 1/N_{tr})^{N_{tr}}$. За довољно велико N_{tr}, поменути израз се асимптотски приближава e^{-1}, што значи да ће величина скупа за проверу бити ≈ 0.368.

Оптимизација параметара класификатора. Неке методе за класификацију поседују параметарску структуру, сачињену од једног или више параметара. Вредности параметара често имају висок утицај на квалитет класификације. Традиционални приступи у решавању овог проблема су: 1) итеративно ручно подешавање параметара; 2) систематска претрага параметара по мрежи вредности (енг. grid search). Други приступ је заснован на подели простора могућих параметара на уједначене регионе, а потом провери квалитета класификације у границама тих региона. Ограничење тог приступа је временска неефикасност која је посебно изражена у случају да је број параметара, опсег вредности параметара или прецизност поделе на регионе, висока. Један од познатијих проблема подешавања параметара класификатора је "поткресивање" дрвета у случају класификације дрветом одлучивања. Адекватно подешавање може довести до побољшања квалитета класификације на тест скупу, тј. до спречавања преприлагођавања класификатора тренинг подацима. У контексту вештачких неуронских мрежа, појављује се проблем подешавања броја скривених чворова у унутрашњим слојевима, праг вредности чворова, брзине учења и други. Ако је број скривених чворова сувише мали, постоји могућност да мрежа неће бити у стању да апроксимира класификациону функцију. За превелики број скривених чворова, мрежа ће моћи да научи функцију, али ће јој у том процесу бити потребно много више времена. Код методе подржавајућих вектора, појављује се проблем подешавања параметара кернелске функције, као и регуларизационог параметра. Правилним одабиром ових параметара, квалитет класификације се може значајно побољшати.

1.3.3 Препроцесирање података

Улазни скупови података обично имају недостатке који их чине неадекватним, а понекад и неупотребљивим за намене класификације. Одређеним техникама препроцесирања и оптимизације могуће је надоместити недостатке и побољшати квалитет класификације. Сада ће бити изложени само неки од познатијих проблема који се појављују у фази препроцесирања.

Недостајуће вредности подразумевају непостојање вредности атрибута за подскуп скупа улазних података. Разлози за непостојање неких вредности могу бити различити: грешке у раду мерног инструмента ако се ради о мерењу физичких феномена, вољно ускраћивање одговора у случају да се ради о испитаницима и др. Начини разрешавања ових проблема су такође разноврсни. Најједноставнији приступ је елиминација података. Постоје хоризонтална и вертикална елиминација. Прва подразумева избацивање појединачних података који имају недостајуће атрибуте. Вертикална елиминација се односи на уклањање неког атрибута, и најчешће се примењује у случајевима када су недостајуће вредности сконцентрисане око тог атрибута. Други приступ у решавању проблема недостајућих вредности је додељивање вредности: недостајућа вредност атрибута се може попунити просечном вредношћу тог атрибута на целом скупу у случају реалних вредности, медијаном у случају редних атрибута, или најчешћом вредношћу у случају категоричких. Уместо вредности засноване на целом скупу, може се користити и вредност заснована на скупу најближих суседа. Трећи приступ подразумева задржавање података са недостајућим вредностима у скупу података, али и игнорисање недостајућих вредности у случају да се јави потреба за њиховим коришћењем. Ово значи да се у процесу тренирања или тестирања класификационог модела, мере квалитета и друге потребне вредности израчунавају различито за различите податке. У случајевима када је број оваквих података мали, то не мора довести до нарушавања квалитета класификације.

Одабир атрибута је значајан аспект у препроцесирању података и припада широј класи метода које се користе за димензиону редукцију. Одабир атрибута има двојаку улогу. Прва улога је смањивање димензије улазног проблема што као последицу може имати драстично смањење времена потребног за тренирање класификационог модела. Друга улога је да са смањивањем броја

атрибута и сам класификациони модел постаје интуитивнији. Проблем одабира атрибута подразумева издвајање подскупа атрибута који су релевантни за процес класификације. Један од алтернативних назива овог процеса је и вертикална рестрикција података, јер ако се скуп података посматра табеларно, атрибути су представљени колонама. Постоји велики број метода за решавање проблема одабира атрибута и све оне се могу груписати у три категорије: 1) филтер методе (енг. filter methods), 2) омотач методе (енг. wrapper methods) и 3) угњежедене методе (енг. embedded methods) који формирају подскуп одабраних атрибута у фази тренирања класификационог алгоритма. Филтер методе су врло ефикасне и обично користе само неколико пролаза кроз скуп података како би извршиле елиминацију непотребних атрибута. Често су засноване на својствима улазног скупа података: ентропији, информационом критеријуму, симетричност, нормалности података итд. Филтер методе су препроцесирајуће у правом смислу те речи. То се не може рећи за другу групу метода, тзв. омотач методе. Одабир атрибута код ове групе метода је другачији. Користи се оптимизациона техника као "омотач" око класификационог модела. Оптимизациона техника потом тражи такав подскуп атрибута који, када се проследи као улаз у класификациони модел, максимизује функцију циља. Показује се да су у пракси омотач методе квалитетније од филтер метода, али имају проблем наслеђене (инхерентне) временске неефикасности зобг начина на који решавају проблем одабира атрибута.

Подешавање тежина атрибута (енг. feature weighting) је проблем одређивања оптималног степена утицаја појединачних атрибута. Према стандардној структури класификационог модела, сви укључени атрибути имају исти значај. Међутим, често неки од атрибута нису релевантни, или поседују шум, који може да наруши квалитет класификације. У идеалном случају, подешавањем тежина атрибута, ирелевантни атрибути добијају тежине блиске или једнаке нули, док су вредности тежина релевантних атрибута веће од нуле и усаглашене са њиховим релативним значајем. Према приступу решавања овог проблема, постоје две групе метода ([WAM97]): једнопролазни методи (енг. online optimization) који узимају у обзир својства скупа података и вишепролазни (енг. batch optimization) који користе повратну информацију класификатора. Првом групом метода вредности тежина атрибута се ажурирају секвенцијалним

проласком кроз тренинг скуп. Проблем који се у том процесу јавља је велика зависност коначних тежина од редоследа података у тренинг скупу. Другом групом метода врше се вишеструки проласци кроз тренинг скуп, што елиминише претходни проблем зависности од редоследа података по цени мање ефикасности. У литератури су предложени различити приступи засновани на градијентним техникама, симулираном каљењу, генетским алгоритмима итд.

Одабир података (енг. instance selection) подразумева одабир најмањег подскупа података на основу којег је класификатор у стању да произведе виши или бар исти квалитет класификације. У литератури се овај проблем назива још и проблем хоризонталне редукције улазног скупа, јер се врши елиминација ирелевантних података који су у табели података представљени као редови. Добит од процеса одабира података је двојака: 1) димензија улазног скупа података је смањена што може довести до великог побољшања временске ефикасности класификатора; 2) елиминисани су подаци са шумом, грешкама, подаци ван граница (енг. outliers) који могу умањити квалитет класификације. Према [BM02], уобичајена су два основна приступа у решавању овог проблема:

1. *Побољшавање уклањањем.* Уклањањем одређених података често је могуће повећати квалитет класификације. Ово је могуће јер се уклањају подаци који имају шум, грешку или једноставно нису довољно репрезентативни.

2. *Ненарушавање уклањањем.* Уклањањем податка задржава се квалитет класификације. Ово значи да је податак непотребан, односно редудантан, тако да се уклањањем доприноси смањењу димензије улазног проблема.

У литератури су предложени и неки приступи ([KJ99; FML02]) у којима се интегрисано посматра хоризонтална и вертикална рестрикцију скупа података, односно проблем одабира атрибута и проблем одабира података.

Неравномерна расподела класа. За скуп података са две класе се каже да има неравномерну расподелу класа (енг. class imbalance) ако је однос броја података једне класе значајно мањи од броја података друге класе. Ово је чест проблем у реалним скуповима података: подаци о ретким болестима, откривање превара, непожељних порука, итд. Према [MS07] постоје четири основна приступа у решавању овог проблема:

1. приступ заснован на поновном узорковању (енг. resampling) у циљу балансирања класа;
2. измена постојећег алгоритма учења (измена на алгоритамског нивоу);
3. измена у начину мерења квалитета класификације (прилагођавање мере квалитета);
4. приступи засновани на везама између неравномерности расподеле класа и осталих карактеристика сложености.

1.4 Метахеуристичка оптимизација

У научним и индустријским применама, израчунавање оптималних решења оптимизационих проблема није увек могуће. Често се у таквим ситуацијама могу применити технике метахеуристичке оптимизације које дају "довољно добра" решења. Метахеуристике чине фамилију апроксимативних метода и за разлику од егзактних алгоритама не гарантују оптималност добијеног решења. Са друге стране, њихова ефикасност често надомешћује непостојање потврде оптималности и чини их прихватљивим за велики број проблема који се појављују у индустрији, инжењерским, природним и другим дисциплинама. Такође, за разлику од стандардних апроксимативних алгоритама, метахеуристике не дефинишу колико је решење близу оптималног. Реч хеуристика потиче од старе Грчке речи *heuriskein*, што значи уметност проналажења нове стратегије (правила) у решавању проблема. Префикс *meta* такође представља Грчку реч, која значи методологија вишег нивоа. Метахеуристичке оптимизационе методе (или метахеуристичке методе претраге) се стога могу дефинисати као хеуристичке методе вишег нивоа, односно високог степена генерализације, што им омогућава да буду применљиве на широкој класи оптимизационих проблема.

Када користити метахеуристике? Својства разматраног оптимизационог проблема се морају узети у обзир када се одговара на ово питање. Најпре је потребно оценити временску и просторну сложеност проблема, а потом и домен примене, односно тежину тест проблема (инстанци) над којима ће се оптимизациона метода применити. Увек је боље дати предност егзактној

методи уколико њоме може да се реши проблем у разумном времену. На пример, може се десити да проблем припада класи NP-тешких проблема, али да су улазне димензије проблема мале, и проблем се може ефикасно решити неком егзактном методом. Чак и када су улазне димензије средње, или велике, може се десити да проблем има специфичну структуру која омогућава да се он реши егзактном техником. Са друге стране, може се десити да проблем припада класи полиномско решивих проблема, а да су све егзактне методе врло неефикасне. У многим случајевима од практичног значаја се дешава да је проблем тешко решити због великог броја локалних оптимума. Број локалних оптимума може при том да расте експоненцијално са растом димензије проблема. У тим ситуацијама се може десити да чак и за проблеме средњих димензија, методе које гарантују оптималност решења, користе превише времена за извршење. Тада се предност може дати метахеуристици.

Подела оптимизационих метода према [Tal09] је представљена Сликом 1.8.

Слика 1.8: Подела оптимизационих метода

Када су у питању метахеуристике, једна од могућих подела је пре-

ма броју решења које се користе у процесу претраге простора решења. Према том критеријуму, метахеуристике су подељене на две основне категорије: 1) метахеуристике чија је претрага вођена једним решењем, и 2) метахеуристике чија је претрага вођена популацијом могућих решења (популационе метахеуристике). Метахеуристике вођене једним решењем трансформишу једно решење током читавог процеса претраге, док популационе користе скуп могућих решења која међусобно интереагују и еволуирају. Детаљнији приказ свих метахеуристика превазилази оквир овог рада. Овде ће укратко бити описане само неке од њих.

Метахеуристике вођене једним решењем. У решавању оптимизационог проблема, метахеуристике вођене једним решењем побољшавају то једно решење итеративно. Процес претраге би се могао представити као "пут" кроз простор решења вођен праћењем перспективних суседних решења. У том својству, постоје два основна оператора претраге: 1) генерисање нових кандидат решења и 2) замена старог решења одабраним новим. Скуп нових кандидат решења се обично добија локалном претрагом, тј. истраживањем суседства (енг. neighborhood) активног решења. Процес генерисања и замена се итеративно извршава док се не достигне критеријум завршетка алгоритма, и он може бити дат максималним бројем итерација, минималним степеном побољшања решења, временским ограничењем и др. Процес претраге може бити са памћењем и без памћења. Први случај подразумева да се користи меморијска структура за памћење претходних или неких одабраних претражених делова простора претраге. На Слици 1.9 је приказан шематски општи механизам примене метахеуристике засноване на једном решењу.

Надаље ће, у кратким цртама, бити описане три препознатљиве методе из ове групе метахеуристика.

Једна од првих метода заснованих на једном решењу је *симулирано каљење* (енг. simulated annealing). Метода је предложена у раду [KV+83], и спада у подгрупу техника инспирисаних природом. Идеја је потекла од једног од принципа статистичке механике према којем се у процесу каљења метала, најпре врши загревање, обликовање, а потом контролисано хлађење које омогућава достизање чврсте кристалне решетке. По аналогији са тим процесом, код методе симулираног каљења, чврстина кристалне решетке одговора функцији

```
   улаз  : Улазни проблем P
   излаз: Најбоље решење s_n
1  n = 0;
2  s_n =креирајИницијалноРешење(P);
3  while ниje задовољен критеријум завршетка do
4  |  C = генеришиСкупКандидатРешења(s_n, P);
5  |  s_{n+1}=одабериНовоРешење(C);
6  |  n = n+1;
7  end
```

Слика 1.9: Општа шема метахеуристике вођене једним решењем

циља која се оптимизује, а начин "хлађења" осликава процес претраге простора могућих решења.

Табу претрага (енг. tabu search), метода предложена у [Glo89], представља једну од најчешће коришћених метахеуристика у литератури. У многим аспектима, табу претрага личи на обичну локалну претрагу. Међутим, суштинска разлика је да табу претрага понекад поставља за ново решење и неко које не побољшава претходно. Локална претрага и бирање бољег решења се извршава док год се не достигне локални оптимум. Након тога, метода покушава да побегне из локалног оптимума одабиром решења које може бити и горе од тренутног. Додатно својство методе је да памти листу претходних решења и на тај начин избегава враћање у њих.

Метода променљивих околина (енг. variable neighborhood search) је предложена у [MH97], и заснива се на итеративној локалној претрази при чему се суседства за локалну претрагу мењају. Под суседствима се мисли на скупове суседних решења. У најједноставнијем случају, ако је решење представљено низом бинарних вредности, једно суседство за неко посматрано решење може бити скуп свих бинарних низова који су на Хаминговом растојању један. Пракса је да се креће са једноставнијим суседствима која су обично и мање захтевна за претраживање, па се, уколико то не даје резултате, прелази на сложенија, а самим тим и временски захтевнија суседства.

Популационе метахеуристике. Популационе метахеуристике врше итеративно побољшавање скупа (популације) решења. Најпре се поставе иницијална решења, а након тога се врши итеративна замена или спајање активних

популација решења у нове популације. Процес итеративног побољшавања популације решења се завршава када се достигне критеријум завршетка. Као и код метахеуристика вођеним једним решењем, и овде се могу препознати два општа механизма: 1) генерисање нове популације решења и 2) замена популације новом. Битно је приметити да замена не мора бити потпуна, већ може бити делимична, јер се нека решења могу задржати. Такође, замена се не мора вршити у класичном смислу, већ је могуће вршити еволуцију решења из претходне итерације. Замена може бити "са памћењем", што значи да ће се у обзир узимати и претходне популације решења, а не само активна. Општа шема примене популационе метахеуристике је приказана на Слици 1.10.

улаз : Улазни проблем P
излаз: Најбоље решење из популације pop_n

```
1 n = 0;
2 pop_n = креирајИницијалнуПопулацијуРешења(P);
3 while није задовољен критеријум завршетка do
4 |   pop'_n =генеришиНовуПопулацију(pop_n, P);
5 |   pop_{n+1}=замениПопулацију(pop_n ⋃ pop'_n);
6 |   n = n+1;
7 end
```

Слика 1.10: Општа шема популационе метахеуристике

Већина популационих техника је инспирисана процесима који се појављују у природи. Једна од познатијих подгрупа популационих метахеуристика су *еволутивни алгоритми* и њихова основна мотивација је симулирање еволуције врста. Популације решења су представљене јединкама, а квалитет решења који је њима дат се оцењује степеном прилагођености јединке. Боље прилагођене јединке, као и у еволуцији, имају већа права по питању остављања потомства. Једна од најпопуларнијих метода из групе еволутивних алгоритама је *генетски алгоритам*. Најчешће коришћена варијанта алгоритма је заснована на бинарној репрезентацији јединке и природно инспирисаним механизмима селекције, укрштања и мутације.

Следећа велика група популационих метахеуристика су оне инспирисане колективним деловањем врста, као што су мрави, пчеле, рибе, птице, и др. Будући да су све поменуте врсте групишу у ројеве, тј. јата, често се овај

вид оптимизације назива и оптимизација ројевима (енг. swarm optimization). Јединке у ројевима су једноставни агенти, неспособни да сами доносе одлуке. Међутим, у групи, индиректном сарадњом и комуникацијом, ројеви као целина дејствују на интелигентан начин, и показује се, успевају да се изборе са врло тешким оптимизационим задацима. Оптимизација колонијама мрава (енг. ant colony optimization - ACO) је популарна метода из ове групе. Предложио ју је M. Dorigo, 1992. године у раду [Dor92]. Свака јединка система је представљена једним мравом. Током кретања мрава, на земљу се испушта феромон, који сугерише правац кретања другим мравима. Временом, мрави успевају да нађу најкраћи пут између две локације (колоније и извора хране), једноставним корекцијама своје путање према локацијама где је количина феромона већа. Сви ови природни аспекти, са одређеним додацима, су уврштени у механизме ACO методе.

1.5 Метахеуристика заснована на електромагнетизму - EM

Метахеуристика заснована на електромагнетизму (енг. electrogmagnetism-like algorithm - EM), предложена у [BF03], представља оптимизациону технику инспирисану механизмима интеракције наелектрисаних честица. EM је популациона метода, чију популацију чини скуп EM тачака (EM честица). Свака EM тачка представља једно решење проблема који се решава. EM тачке које представљају (кодирају) боља решења су *награђене* вишим степеном наелектрисања, што је кључно за даље вођење процеса претраге, јер EM тачке са вишим набојем привлаче остале честице јаче ка себи. Веза привлачења EM честица је слична Кулоновом закону, према којем је интензитет привлачења сразмеран наелектрисањима честица, а обрнуто сразмеран њиховој удаљености.

У [BFS04] су дата разматрања која се тичу конвергенције EM методе. EM се показао као успешна техника у решавању великог броја проблема са практичним и теоријским значајем: у [AG10] је предложен EM метод за решавање проблема глобалне оптимизације са ограничењима; у [SL11] се EM користи за решавање проблема одабира атрибута; хибридни алгоритам заснован на EM и симулираном каљењу за решавање проблема распоређивања послова

```
улаз: N_it, M
1  p = креирајИницијалнеEMТачке(M);
2  for iter ← 1 to N_it do
3  |  for i ← 1 to M do
4  |  |  рачунајФункцијуЦиља(p_i);
5  |  end
6  |  примениЛокалнуПретрагу(p);
7  |  рачунајНаелектрисања(p);
8  |  рачунајСиле(p);
9  |  помериEMТачке(p);
10 end
11 испишиРешење();
```

Слика 1.11: Основни EM метод

(енг. job-shop problem) је описан у [TMKN09]. Хибридни EM за решавање проблема рутирања возила са капацитетима (енг. capacitated vehicle routing problem) се предлаже у [YE10]. У раду [Fil11] се уводи EM метод за решавање вишеструког хаб локацијског проблема без капацитета (енг. uncapacitated multiple allocation hub location problem), док се у [Cue+12] EM користи за аутоматску детекцију кружних облика у сликама које поседују шум. Поред ових, разматрано је још неколико проблема у литератури: проблем прекривања скупа са једном ценом (енг. unicost set covering problem) [NATG10], проблем одређивања топологије мреже са минималном потрошњом енергије (енг. strong minimum energy topology problem) [Kar12], проблем максималне пермутације са датим скупом ограничења (енг. maximum betwenness problem) [FKM13], проблем подешавања параметара машине подржавајућих вектора (енг. support vector machine parameter tuning problem) [Kar+13], проблем подешавања тежина атрибута [KŠC14] (енг. feature weighting problem) и други.

Главне компоненте основног EM алгоритма су приказане на Слици 1.11.

EM користи само 2 контролна параметра: N_{it} представља број итерација главног циклуса, док је M укупан број EM тачака, тј. величина популације. Свакој тачки се најпре додељује иницијално решење, а након тога, алгоритам улази у главни циклус. У основној варијанти EM методе, иницијализација се врши на случајан начин тако што се i-тој тачки $\mathbf{p_i}$ додељује низ од N случајних вредност из интервала $[0, 1]$. N представља димензију проблема

који се решава. Број итерација главног циклуса је N_{it} и при том у свакој итерацији, свака EM тачка бива подвргнута израчунавању функције циља. Функција циља представља метрику квалитета једне EM тачке, тј. решења које та тачка представља. Након завршетка главног циклуса, примењује се процедура локалне претраге (енг. local search - LS). У основној варијанти EM методе локална претрага се примењује над свим тачкама, али то не мора увек бити случај. Понекад се, зарад убрзања алгоритма или избегавања локалних оптимума, врши селективна примена LS процедуре над подскупом скупа свих тачака. Сам процес локалне претраге подразумева тражење локалних побољшања у околини активног решења. Постоји више начина на који се побољшање тражи или примењује. Обично се користи варирање појединачних вредности из низа од N вредности посматране EM јединке. Ако се врши систематска претрага свих N вредности, примена побољшања може бити тренутна (енг. first improvement strategy) или одложена са применом најбољег побољшања (енг. best improvement strategy). Постоје и несистематски обиласци засновани на хеуристици, случајни обиласци, итд. Следећи корак у извршавању EM методе је израчунавање наелектрисања и сила које производе појединачне EM тачке. Функције циља EM тачке, индиректно, кроз наелектрисања и силе, дефинишу начин померања тачака кроз простор решења. При рачунању наелектрисања, узима се у обзир вредност функције циља посматране тачке, али и функције циља осталих тачака (1.23).

$$q_i = exp\left(-N\frac{p_i^{obj} - p_{best}^{obj}}{\sum_{k=1}^{M}(p_k^{obj} - p_{best}^{obj})}\right) \tag{1.23}$$

Интуиција иза додељивања виших наелектрисања бољим EM тачкама је заснована на идеји да боље EM тачке треба да имају значајнију улогу у процесу претраге простора решења. У формули (1.23), q_i означава наелектрисање, p_i^{obj} функцију циља i-те EM тачке. Може се приметити да најбоља тачка добија наелектрисање у вредности 1, док остале добијају неалектрисања са вредностима из интервала $(0, 1]$.

Након израчунавања наелектрисања, рачунају се силе интеракције међу паровима тачака. За одабрану EM тачку $\mathbf{p_i}$, $\mathbf{F_i}$ представља укупну силу која

утиче на њу, а тачан израз по коме се укупна сила добија је дат у (1.24). Приказани израз се односи на случај минимизације, али се једноставно може прилагодити и проблемима у којима се врши максимизација.

$$\mathbf{F_i} = \begin{cases} \sum_{j=1, j\neq i}^{M} (\mathbf{p_j} - \mathbf{p_i}) \frac{q_i \times q_j}{||\mathbf{p_j} - \mathbf{p_i}||^2}, & p_j^{obj} < p_i^{obj} \\ \sum_{j=1, j\neq i}^{M} (\mathbf{p_i} - \mathbf{p_j}) \frac{q_i \times q_j}{||\mathbf{p_j} - \mathbf{p_i}||^2}, & p_j^{obj} \geq p_i^{obj} \end{cases} \tag{1.24}$$

Као што се може видети, сила $\mathbf{F_i}$ се добија слагањем сила узајамних интеракција међу паровима које јединка $\mathbf{p_i}$ гради са осталим јединкама из популације. Појединачна интеракција између јединке $\mathbf{p_i}$ и $\mathbf{p_j}$ се рачуна по изразу који је сличан Кулоновом закону, а који гласи да је интензитет интеракције између две наелектрисане честице сразмеран наелектрисањима честица и обрнуто сразмеран њиховој удаљености. Као завршни корак у овој процедури, извршава се померање јединки. При рачунању правца, смера и интензитета помераја користе се претходно израчунате силе:

$$p_i^k = \begin{cases} p_i^k + \lambda \frac{F_i^k}{||\mathbf{F_i}||}(1 - p_i^k), & F_i^k > 0 \\ p_i^k + \lambda \frac{F_i^k}{||\mathbf{F_i}||} \cdot p_i^k, & F_i^k < 0 \end{cases} \tag{1.25}$$

Додатно, елемент случајности се додаје путем променљиве λ. Ова променљива се бира случајно на интервалу $[0, 1]$ у свакој итерацији и за сваку ЕМ тачку. У (1.25), F_i^k представља k-ту координату вектора силе која одговара i-тој ЕМ јединки.

1.6 Преглед примена метахеуристика у класификацији

У овој секцији је направљен преглед неких примена метахеуристика у класификацији. У поглављима која следе, биће направљена детаљнија анализа литературе за сваки од разматраних проблема.

Проблем одабира гена (енг. gene selection problem) подразумева одабир подскупа задатог скупа гена. Решавање овог проблема може довести до повећања тачности класификације у различитим класификационим

проблемима из домена рачунарске генетике. Проблем је само специјализација проблема општег проблема одабира атрибута у класификацији. У [DH10] је направљен преглед неких метахеуристичких приступа за решавање проблема одабира гена. У раду се појашњава како се рангирајући коефицијенти линеарних класификатора, нпр. SVM, могу искористити у спрези са локалном претрагом и еволутивним техникама зарад побољшања тачности класификације. Рад [Tal+08] се такође бави проблемом одабира гена. У њему је предложена платформа заснована на методи подржавајућих вектора (SVM), генетском алгоритму (GA) и оптимизацијиројевима (PSO) за класификацију високодимензионалних низова гена. GA и PSO су употребљени за проналажење малих подскупова гена који су довољно информативни. Након тога је SVM са техником унакрсне провере употребљен за класификацију. Метода је упоређена са различитим техникама из литературе, и добијен је напредак у погледу резултата класификације над 6 јавно доступних скупова података.

Аутори рада [YWD05] користе технике машинског учења као што су Наивни Бајес, дрво одлучивања и SVM, у циљу аутоматског филтрирања непожељних (енг. spam) порука. Претходни приступи су користили кључне речи из порука електронске поште како би издвајали бинарне атрибуте из посматраног корпуса. Проблем са овим приступом је што се кључне речи мењају временом, тако да су у овој студији, својства понашања пошиљаоца непожељних порука (енг. spammers) коришћена као атрибути. Метахеуристике су употребљене за намену издвајања укупно 113 нових атрибута. Показало се да је овај приступ допринео одређеном унапређењу квалитета и ефикасности класификације. У [UM10] се такође разматра проблем одабира атрибута над бинарним класификационим проблемима из јавно доступних база података. Они предлажу дискретну методу оптимизацијеројевима, а као класификатор користе логистичку регресију. Тестирања спроведена на јавно доступним тест проблемима су показала да је предложена PSO метода упоредива по питању тачности класификације и ефикасности са методама из литературе.

Y. Marinakis је у неколико радова демонстрирао потенцијал применљивости класификационих техника поптпомогнутих метахеуристикама у решавању финансијски оријентисаних проблема. У [Mar+09], аутор се бави развојем класификационог алата за решавање проблема одређивања кредитног ризика и

оцена ревизора. Показана побољшања у погледу класификације су обезбеђена применом оптимизације колонијом мрава (ACO) и оптимизације роjевима (PSO) на проблем одабира атрибута. Слично тој студији, у [MMZ10] је решаван исти проблем, али нешто другачијом оптимизационом техником. Овде је примењена варијанта оптимизације роjевима пчела (енг. honey bees mating optimization) у спрези са методом најближих суседа. Метод је тестиран на проблемима утврђивања кредитног ризика и упоређен са различитим оптимизационим техникама: оптимизације роjевима, колонијом мрава, генетским алгоритмом и табу претрагом. У раду [Mar+08] су представљена побољшања методе најближих суседа заснована на примени различитих метахеуристика: табу претрази, генетском алгоритму и оптимизацији колонијом мрава. Квалитет класификације је истестиран на подацима добијеним од водеће Грчке Комерцијалне Банке о зајмовима 1411 фирми. Циљ је био извршити класификацију фирми на неколико класа које описују ниво кредитног ризика. Направљена су поређења са различитим методама из литературе: UTADIS, SVM, CART, и др. Још једна од примена у финансијама је описана у [Wan+12]. Овде се користи метода која комбинује грубе скупове (енг. rough set) и расејану претрагу (енг. scatter search) за проблем одабира атрибута. Тестирање је спроведено над три класификациона модела: дрвету одлучивања J48, вештачкој неуронској мрежи и логистичкој регресији. Показано је да предложена метода значајно побољшава ефикасност и тачност класификације.

Папаниколау метод је медицинска дијагностичка техника којом се омогућава утврђивање пре-малигног стања људске ћелије. Правовременом дијагнозом се може спречити напредовање ћелија у форму инвазивног карцинома. У раду [MDJ09] је предложен метахеурички метод који помаже у класификацији ћелија Папаниколау техником. Коришћене су две базе података, прва са 917 слика, а друга са 500 слика изолованих људских ћелија. Свака ћелија је описана са 20 нумеричких својстава и постоји 7 дијагностичких степена (класа). Ослабљени захтев је, међутим, извршити бинарну класификацију на нормалне и ненормалне ћелије. У том циљу, метода најближих суседа је потпомогнута одабиром атрибута, а одабир атрибута је оптимизован генетским алгоритмом. Показано је да техника надмашује претходне приступе из литературе по питању тачности класификације. Проблеми медицинске дијагностике су

обрађени и у [SR07]. У том раду, проблем одабира атрибута је решен оптимизацијом колонијом мрава, а као класификациони модел је коришћена вештачка неуронска мрежа. У [Zuo+08] се предлаже модификација k-NN методе, под скраћеницом KDF-WKNN (енг. kernel difference-weighted k-nearest neighbor), за дијагностиковање срчане аритмије. Експериментални резултати су показали да је предложени метод супериоран у односу на обичну методу најближих суседа и упоредив са најбољим методама за класификацију из литературе.

Преглед неколико различитих техника одабира атрибута је направљен у [Yus09]. GRASP метода (енг. greedy randomized adaptive search procedure), табу претрага и меметички алгоритам (енг. memetic algorithm) су упоређени са једном од највише коришћених метода, генетским алгоритмом, и још неким класичним методама за одабир атрибута: SFFS (енг. sequential forward floating selection) и SBFS (енг. sequential backward floating selection). Показано је да су GRASP и табу претрага статистички значајно бољи од других метода. Још један преглед метода везаних за класификацију је направљен у [AMBM13]. Контекст су проблеми медицинске дијагностике. Описане су различите метахеуристике за решавање проблема подешавања параметара и одабира атрибута.

У [Hor+12] се оптимизација појем свитаца (енг. firefly algorithm - FA) примењује за подешавање вредности параметара између унутрашњег и излазног слоја неурона радијалне неуронске мреже (енг. radial basis function network). Метод је упоређен са градијентним спустом (енг. gradient descent - GD), генетским алгоритмом, оптимизацијом појевима (PSO) методом оптимизације појевима пчела (енг. artificial bee colony algoritm - ABC). Тестирање квалитета је извршено над јавно доступним UCI тест проблемима [BL13], и показано је да FA надмашује GD и GA по тачности класификације, али не и PSO и ABC методе. Аутори закључују да су технике засноване на оптимизацији појевима (FA, PSO, ABC) добар избор када се решава проблем подешавања параметара радијалне неуронске мреже. У раду [BL07] се предлаже хибридна класификациона техника заснована на дрвету одлучивања, оптимизацији колонијом мрава и еволутивним стратегијама под називом ACO-DTree. Генерисање дрвета и подешавање параметара у процесу генерисања се на тај начин контролише

метахеуристикама. Проблем подешавања параметара се разматра и у [BS10]. Аутори показују да су технике засноване наројевима (ACO и PSO) погодне, јер се њиховом применом тачност класификације машине подржавајућих вектора унапређује. A. Candelieri у својој докторској дисертацији [Can11] предлаже радни оквир за класификацију побољшану адекватним одабиром параметара. У том циљу, A. Candelieri примењује неколико метахеуристика: генетски алгоритам, табу претрагу и оптимизацију колонијом мрава, у подешавању параметара методе подржавајућих вектора.

У [KCB04], аутори се баве класификацијом типа земљишта тако што анализирају хиперспектралне снимке земљишта. За ту намену користе класификацију бинарним дрветом одлучивања и одабир атрибута који је оптимизован табу претрагом. Резултати су показали да је одабир атрибута, побољшан табу претрагом, боље решење од одабира атрибута заснованог на похлепној техници или Фишеровој дискриминанти. Још две примене табу претраге и методе променљивих околина су предложене у [GT+04] и [PCN07]. Домен примене у првом раду је проблем одабира атрибута, а у другом истовремено решавање проблема одабира атрибута и проблема подешавања тежина атрибута за линеарну дискриминантну функцију. У првом раду се показало да је предложени дует метахеуристика бољи од генетског алгоритма за исти проблем, док су у другом, аутори експериментално утврдили да је тачност класификације предложене методе статистички значајно боља него код класичних дискриминантних линеарних класификатора (класичне дискриминантне анализе и логистичке регресије). Још један сличан допринос је дат у [TBK07]. Овде се разматра истовремени проблем одабира атрибута и подешавања њихових тежина у оквиру k-NN методе за класификацију. За оптимизацију овог процеса је употребљена табу претрага, а резултати тестирања су показали значајно унапређење у погледу тачности класификације. Нешто другачија примена табу претраге је предложена у [Leb+05]. Овде је она употребљена за подешавање параметара методе подржавајућих вектора. Након тога, подешени SVM је употребљен за класификацију пиксела, проблема који се јавља као иницијална фаза у процесу сегментације слика у боји. Експериментално је показано да одабир атрибута побољшан табу претрагом значајно доприноси повећању ефикасности процеса класификације пиксела.

Поглавље 2

Примена EM у одређивању параметара SVM

2.1 Проблем подешавања параметера SVM

У овом поглављу се разматра проблем подешавања параметара SVM, тачније параметара кернела који се после користе у SVM алгоритму. Резултати из овог поглавља су приказани у раду [Kar+13]. Подешавање SVM параметара може бити веома захтеван задатак са аспекта временске сложености с обзиром да број параметара може бити велики и зато што параметри припадају домену реалних бројева. Стандардни приступ у решавању проблема подешавања параметара је тзв. претраживање решетке (енг. Grid search - GS). GS је заснован на дискретизацији домена реалних вредности, што се обично своди на одабир дискретног интервала и одређивање чворова на узастопним удаљеностима за одабрани интервал. У случају да је проблем дводимензионалан (два параметра), добија се решетка вредности, код тродимензионалног се претражује коцка вредности, итд. Након трансформације у дискретни домен, врши се исцрпна претрага свих чворова, тј. комбинација дискретних параметара. GS је врло захтевна метода када су у питању рачунарски ресурси, и обично не успева да произведе решења када је број параметара или/и степен дискретизације висок. Сходно томе, хеуристички приступи који не користе тако исцрпну претрагу простора решења су погодни.

У наставку ове секције описана је нелинеарна класификација код SVM, а затим и употреба кернелског учења заснованог на једном или више

кернела. У другој секцији је направљен преглед литерауре везане за проблем одређивања параметара у SVM. После тога су описани елементи предложеног EM алгоритма: иницијализација, функција циља, локална претрага, итд. Последње две секције приказују експерименталне резултате, закључке базиране на њима, као и правце даљег развоја.

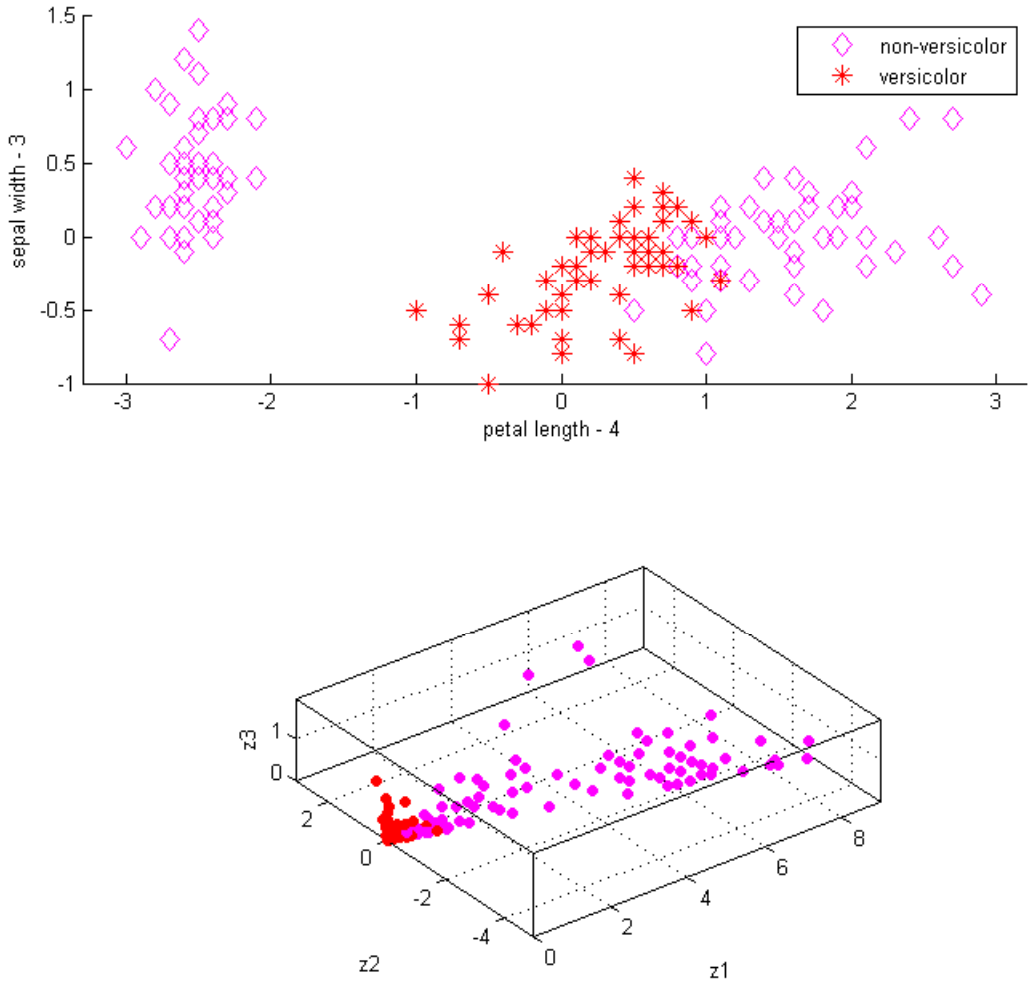
2.1.1 Нелинеарна SVM класификација

У секцији 1.2.3 су описани основни елементи методе подржавајућих вектора са пратећим формулацијама класификационих проблема у случајевима да су подаци линеарно раздвојиви, или "скоро" линеарно раздвојиви (случај *меке* маргине). У случају да подаци нису линеарно раздвојиви и да употреба *меке* маргине није довољна, примењује се трансформација над простором атрибута. У том својству се врши замена сваког вектора $\mathbf{u}$ са $\mathbf{\Phi}(\mathbf{u})$, где $\mathbf{\Phi} : \mathbf{R}^N \rightarrow \mathbf{R}^{N'}$ пресликава оригинални простор атрибута у простор више димензије, у којем је линеарна раздвојивост података могућа. На Слици 2.1 је приказано пресликавање простора атрибута на примеру скупа података Ирис.

У горњем делу слике је приказан бинарни проблем класификације за тип цвета *versicolor* са једне стране, и остала два типа са друге. Са p је означена хоризонтална оса (petal length), а са s вертикална (sepal width). Пре нелинеарног пресликавања, примењена је и транслација атрибута дуж обе осе $(p, s) \rightarrow (p - 4, s - 3)$, у циљу поравнавања података типа *versicolor* са координатним почетком. Након тога је извршено пресликавање:

$$\mathbf{\Phi}(p, s) = (z1, z2, z3) = (p^2, \sqrt{2}ps, s^2). \tag{2.1}$$

Може се приметити да су, након примене пресликавања, подаци постали линеарно раздвојиви јер се може дефинисати раван која дели податке на оне који су у класи *versicolor* и оне који припадају другој класи, у којој су преостала два типа цвета. Пресликани простор атрибута је веће димензије, па се намеће питање ефикасности. Најпре је потребно пресликати оригиналне векторе атрибута у нови простор, а потом у новом простору, који је нужно веће димензије, извршити сва потребна израчунавања. Сва израчунавања над подацима у пресликаном простору су заснована на скаларном производу. Ово представља битну полазну претпоставку за увођење механизма који се назива



Слика 2.1: Нелинеарно пресликавање простора атрибута

кернелски трик (енг. kernel trick). Кернелски трик је механизам који омогућава да се пресликавање и димензија пресликаног простора занемари из аспекта временске ефикасности. Суштина лежи у употреби функционалне форме која се назива кернел: $K(\mathbf{u}, \mathbf{v}) = \mathbf{\Phi}(\mathbf{u}) \cdot \mathbf{\Phi}(\mathbf{v})$. Кључно својство кернела је да се он примењује над векторима у пресликаном простору атрибута, али се истовремено може израчунати и у оригиналном простору атрибута. Ово омогућава да никад не долази од директне употребе функције $\mathbf{\Phi}$ и, самим тим, димензија циљног простора не игра улогу у ефикасности израчунавања. Важење овог својства у пресликавању (2.1) се показује на следећи начин:

$$
\begin{aligned}
K(\mathbf{u},\mathbf{v}) &= \mathbf{\Phi}(\mathbf{u})\cdot\mathbf{\Phi}(\mathbf{v}) \\
&= (p_u^2, \sqrt{2}p_u s_u, s_u^2)\cdot(p_v^2, \sqrt{2}p_v s_v, s_v^2) \\
&= (p_u^2 p_v^2, 2p_u s_u p_v s_v, s_u^2 s_v^2) \\
&= (p_u p_v + s_u s_v)^2 \\
&= ((p_u, s_u)\cdot(p_v, s_v))^2 \\
&= (\mathbf{u}\cdot\mathbf{v})^2
\end{aligned}
\tag{2.2}
$$

Из практичне перспективе, SVM кернел представља меру сличности (различитости) између вектора атрибута $\mathbf{u}$ и $\mathbf{v}$. Формални услови, да би функција представљала кернел, су дати тзв. Мерсеровим условом, односно Мерсеровом теоремом [STC04], чији је доказ, због значаја за овај рад, изложен у потпуности. Пре формулисања Мерсерове теореме, потребно је дефинисати неколико појмова.

Дефиниција 5. *Матрица $M \in \mathbf{R}^d \times \mathbf{R}^d$, $d > 0$ је позитивно-семи-дефинитна ако за свако z такво да $z \in \mathbf{R}^d$ важи $z^T M z \geq 0$. На пример, може се показати да је $M = I$ позитивно-семи-дефинитна:*

$$
z^T I z = \sum_{i=1}^{d}\sum_{j=1}^{d} z_i z_j I_{ij} = \sum_{i=1}^{d} z_i^2 \geq 0. \tag{2.3}
$$

Дефиниција 6. *За тренинг скуп података D_{tr} (раније описан у Дефиницији 1) и функцију $K(\mathbf{u},\mathbf{v})$,* матрица кернела K_D *је матрица димензије $N_{tr} \times N_{tr}$ где је $(K_D)_{ij} = K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ за $i = 1, ..., N_{tr}$ и $j = 1, ..., N_{tr}$.*

Теорема 1. *(Мерсерова теорема - [STC04]). Кернел $K(\mathbf{u},\mathbf{v})$ је исправан (валидан) ако и само ако је његова придружена матрица кернела позитивно-семи-дефинитна над целим тренинг скупом података.*

Доказ. ($\Longrightarrow$) Нека је $K(\mathbf{u},\mathbf{v})$ исправан кернел. Онда за његово придружено пресликавање простора атрибута $\mathbf{\Phi}$ важи $K(\mathbf{u},\mathbf{v}) = \mathbf{\Phi}(\mathbf{u})\cdot\mathbf{\Phi}(\mathbf{v})$, па важи и: $(K_D)_{ij} = \mathbf{\Phi}(\mathbf{x}^{(i)})\cdot\mathbf{\Phi}(\mathbf{x}^{(j)})$ за $i = 1, ..., N_{tr}$ и $j = 1, ..., N_{tr}$. Нека је C матрица $[\mathbf{\Phi}(\mathbf{x}^{(1)})...\mathbf{\Phi}(\mathbf{x}^{(N_{tr})})]$, где је сваки $\mathbf{\Phi}(\mathbf{x}^{(i)})$ колонски вектор. Онда важи $K_D = C^T C$. Из овога следи да матрица K_D мора бити позитивно-семи-дефинитна, јер за сваки вектор $z \in \mathbf{R}^{N_{tr}}$, $(z^T C^T)(Cz) \geq 0$.

($\Longleftarrow$) Нека је D_{tr} скуп свих могућих података (претпоставка је да је коначан). Како је придружена кернел матрица K_D позитивно-семи-дефинитна, она има ненегативне сопствене вредности, па се може декомпоновати: $K_D = V^T V$. Нека је $\mathbf{\Phi}(\mathbf{x}^{(i)}) = \mathbf{r_i}$, где је $\mathbf{r_i}$ i-ти ред матрице R. Матрица R на тај начин дефинише следеће пресликавање: $K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \mathbf{r_i} \cdot \mathbf{r_j}$, што представља кернел. □

Мерсерова теорема гарантује да је матрица кернела позитивно-семи-дефинитна, што је кључно за даљи процес формирања SVM дуала, јер се на тај начин гарантује да ће дуал бити конкаван, и самим тим релативно једноставан за оптимизацију.

Након пресликавања оригиналног простора атрибута, добија се другачија дуална формулација проблема од оне дате у (1.15) јер сада максимизациони израз има следећу форму:

$$\max \left(\sum_{i=1}^{N_{tr}} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N_{tr}} y^{(i)} y^{(j)} \alpha_i \alpha_j K(\mathbf{x}^{(i)}), \mathbf{x}^{(j)}) \right). \tag{2.4}$$

уз ограничења:

$$\alpha_i \in [0, C], i = 1, ..., N_{tr} \tag{2.5}$$

$$\sum_{i=1}^{N_{tr}} \alpha_i y^{(i)} = 0 \tag{2.6}$$

Коначно, придружена функција одлучивања је дата са:

$$c(\mathbf{x}) = \operatorname{sgn} \left(\sum_{i=1}^{N_{tr}} \alpha_i y^{(i)} K(\mathbf{x}^{(i)}, \mathbf{x}) + b \right) \tag{2.7}$$

Показано је да вредност параметра C и тип кернела имају значајан утицај на квалитет класификације [LD06]. Адекватну вредност параметра C је потребно тражити на скупу позитивних реалних бројева. Уколико се употребљава више кернела (вишекернелско учење) проблем постаје још тежи с обзиром да је за сваки кернел потребно одредити параметар C.

2.1.2 Учење једног кернела

Учење једног кернела (енг. single kernel learning - SKL) се заснива на подешавању јединственог скупа параметара SVM, тј. скупа параметара у којем фигурише само једна кернелска функција. Ова техника је погодна у ситацији када атрибути не формирају кластере. Као што је раније напоменуто, потребно је подесити регуларизациони параметар C и евентуално додатне параметре кернелске функције K. У овом раду се разматрају два модела кернела: линеарни (2.8) и кернел са радијалном основом (2.9).

$$K(\mathbf{u}, \mathbf{v}) = \mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{N} u_i v_i \tag{2.8}$$

$$K^{\sigma}(\mathbf{u}, \mathbf{v}) = \exp\left(-\sum_{i=1}^{N} \frac{(u_i - v_i)^2}{2\sigma_i^2}\right) \tag{2.9}$$

Линеарни кернел нема унутрашњих параметара пошто је потребно извршити једино скаларни производ између улазних вектора. Стога се процес параметризације заснива на подешавању SVM регуларизационог параметра C. Код кернела са радијалном основом постоје и унутрашњи параметри кернела, тако да је скуп параметара једнак $\{C, \sigma_1, \sigma_2, ..., \sigma_N\}$, где σ_i представља фактор скалирања за i-ти атрибут. Скалирајући фактори су корисни у раду са подацима који имају реалне примене, јер је већина њих сачињена од атрибута који имају различите домене или својства [Cha+02]. У овом поглављу се разматра релаксирана верзија простора параметара за SKL, која се добија подешавањем $\sigma_1 = \sigma_2 = ... = \sigma_N = \sigma$. Следствено, проблем се своди на претрагу простора датог комбинацијама вредности параметара $\{C, \sigma\}$.

2.1.3 Вишекернелско учење

Учење више кернела (енг. multpile kernel learning - MKL) се заснива на сложенијој формулацији проблема него SKL. У MKL моделима, атрибути су распоређени у скуп кластера $\mathbf{G} = \{G_1, ..., G_{N_G}\}$ кардиналности N_G, који представљају непреклапајуће групе атрибута раздвојене према сродности. Сваки од кластера добија придружени кернел $K_1, ..., K_{N_G}$ са себи својственим скупом параметара. Кернели се потом могу груписати линеарним, конусним,

конвексним комбинацијама, или неким другим које користе нелинеарне функционалне форме попут множења, степеновања, итд. У [GA11] је дат детаљан преглед различитих MKL модела. У овом поглављу се разматрају конусне комбинације линеарних кернела (2.10) и кернела са радијалном основом (2.11).

$$\begin{aligned} K^{\alpha}(\mathbf{u},\mathbf{v}) &= \sum_{j=1}^{N_G} \alpha_j K_j(\mathbf{u},\mathbf{v}) \\ &= \sum_{j=1}^{N_G} \alpha_j \sum_{i\in G_j} u_i v_i, \\ \alpha_j &> 0,\ j = 1,...,N_G \end{aligned} \tag{2.10}$$

$$\begin{aligned} K^{\alpha,\sigma}(\mathbf{u},\mathbf{v}) &= \sum_{j=1}^{N_G} \alpha_j K_j^{\sigma_j}(\mathbf{u},\mathbf{v}) \\ &= \sum_{j=1}^{N_G} \alpha_j \exp\left(-\sum_{i\in G_j} \frac{(u_i - v_i)^2}{2\sigma_j^2}\right), \\ \alpha_j &> 0,\ j = 1,...,N_G \end{aligned} \tag{2.11}$$

Јасно је да се у MKL користи простор параметара виших димензија, него што је то случај са SKL моделима. Ово је последица увођења нових коефицијената $\alpha_j, j = 1,...N_G$ у оквиру конусне комбинације и скалирајућих фактора $\sigma_j, j = 1,...,N_G$ у случају да се користи кернел са радијалном основом. Стога се број подешавајућих параметара повећава и износи $H = N_G$ у случају линеарног модела, односно $H = 2N_G$ у случају модела са радијалном основом (овај број параметара такође важи и у SKL моделима где је $N_G = 1$).

2.2 Претходни резултати

Одабир одговарајућег кернела и подешавање његове унутрашње структуре има велики утицај на квалитет предвиђања SVM. Кернелска трансформација је заснована на симетричној позитивно семидефинитној матрици која представља

матрицу сличности за све парове вектора. У раду [CG10] се посебно наглашава значај одабира функција кернела за задати тренинг скуп података. Разлог лежи у чињеници да кернел-базирани методи остварују висок степен генерализације тако што уграђују претходно знање у кернел. У [CG10] аутори долазе до квалитетних кернелских матрица формирањем линеарних комбинација над кернелским матрицама употребом семидефинитног програмирања. Након тога, успешно примењују SVM алгоритам базиран на добијеним комбинацијама матрица у медицинској дијагностици (проблем класификације). У неким случајевима ни одабир адекватног кернела и његове унутрашње параметерске структуре, не доводи до задовољавајућег побољшања класификације. Ово се дешава када је потребно извршити класификацију или регресију над простором хетерогених атрибута, тј. атрибута који нису сродни. У тим ситуацијама, атрибути су обично груписани у кластере сродних атрибута и самим тим их је немогуће описати јединственим кернелом. Срећом, SVM се може применити и у тим случајевима тако што ће користити различите кернеле (вишекернелски приступ) за различите кластере атрибута. Сваки кернел ће на тај начин имати подешену параметарску структуру која одговара кластеру атрибута на који се односи и тиме индиректно повећати моћ предвиђања SVM алгоритма. У [GA11] је дат преглед неколико вишекернелских SVM метода и закључено је да употреба више кернела може допринети квалитету класификације.

У литератури су предложене различите стратегије обиласка простора параметара, тј. њиховог подешавања.

У [KL03] је предложен побољшани GS алгоритам за SKL модел који користи разматрања асимптотског понашања кернела са радијалном основом. У [Kee02] је употребљена горња граница маргине као мера квалитета, што је омогућило решавање проблема великих димензија са преко 10000 подржавајућих вектора. У [Kee02] се користи *углађена* (енг. smoothed) k-компонентна унакрсна провера (енг. k-fold cross-validation) и градијент у правцу SVM параметара како би се добила што ближа подоптимална решења. У [BLJ04] је предложена техника секвенцијалне минималне оптимизације (енг. sequential minimal optimization - SMO) у циљу решавања проблема квадратног програмирања са квадратним ограничењима (енг. quadratically constrained quadratic pro-

gram -QCQP). Ова техника одговара оптимизацији коефицијената конусне комбинације у MKL моделу. У [Son+06] је показано да се QCQP може реформулисати у семиинфинтни линеарни програм и потом ефикасно решити коришћењем стандардне SVM имплементације.

У [FI05] је предложен метод који користи еволутивну стратегију за прилагођавање матрице коваријансе (енг. covariance matrix adaptation evolution strategy - CMA-ES). Овај приступ, поред тога што је применљив у случају високо димензионалних простора параметара, показује и боља решења него GS на малим инстанцама проблема. У [BJLLD09] су представљена два фокусирана GS алгоритма (енг. focused grid search - FGS). Прва варијанта, тзв. детерминистички FGS, врши систематску претрагу и показује се као доста бржи у поређењу са класичним GS. Други, прекаљени FGS (енг. annealed FGS - AFGS) користи елементе случајне претраге у циљу смањивања трошкова претраге. Оба алгоритма су конкурентна са CMA-ES, и при том лака за коришћења с обзиром да не захтевају подешавање било каквих контролних параметара.

За решавање овог проблема, у литератури се може пронаћи неколико метахеуристичких приступа. У [IL04] су предложени генетски алгоритам и алгоритам симулираног каљења за подешавање параметара SVM. Оба алгоритма су се показала робусним и достигла решења блиска оптималним. Генетски алгоритам је нешто бржи, али захтева подешавање већег броја контролних параметара. У [SSA10] је предложен генетски алгоритам који надмашује традиционални GS алгоритам по питању квалитета класификације. У [PK10] је употребљена еволутивна стратегија за подешавање SVM параметара. Аутори овог рада користе различите комбинације RBF кернела у циљу добијања високог квалитета класификације. У [ZCH10] је предложена техника оптимизације мрављим колонијама (енг. Ant colony optimization - ACO). ACO алгоритам се показује успешним у решавању 5 проблема из домена реалне примене, преузетих са UCI репозиторијума машинског учења (енг. UCI Machine Learning Repository, [FA10]). Алгоритам заснован на идеји вештачког имуног система (енг. artificial immune algorithm), за подешавање параметара SVM, је предложен у [AKA11]. Овај алгоритам је такође успешно примењен у решавању неких класификационих проблема, као нпр. у

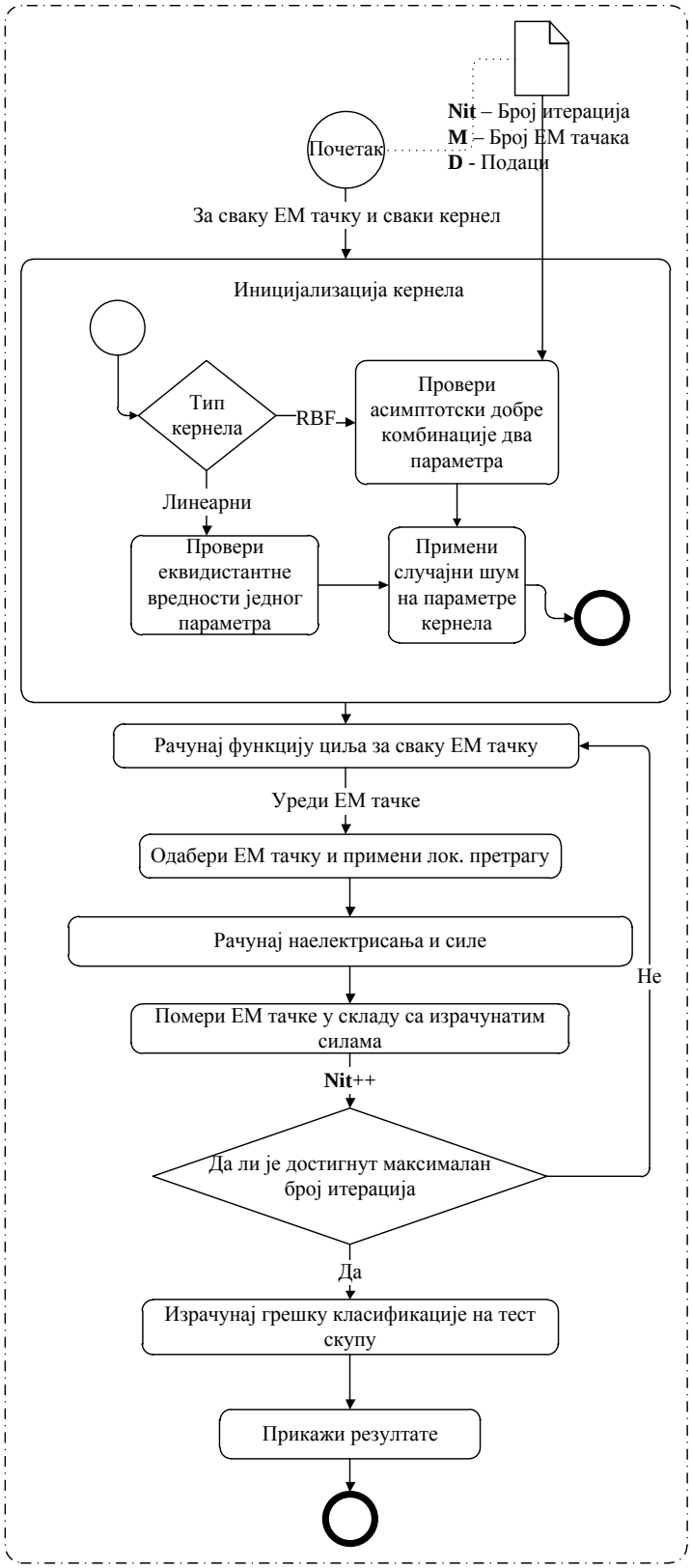
дијагностици отказа индукционих мотора и проблему детекције аномалија. У [GM+11] је описана вишепараметарска метода регресије заснована на методи подржавајућих вектора (енг. support vector machine regression - SVMr) у којем су параметри оптимизовани употребом еволутивне технике. Алгоритам се показао корисним у проблемима прогнозирања (енг. forecasting problems). Аутори рада су израчунали и нове границе за вишепараметарске кернеле које успевају да смање простор претраге параметара SVMr алгоритма. Слично, у радовима [ZHQ13] и [GM+13], SVM се оптимизује применом метахеуристичких алгоритама и, као такав, после користи за потребе прогнозирања. У [ZHQ13] се примењује оптимизација ројевима (енг. particle swarm optimization - PSO), након чега се SVM користи за предвиђање цена акција на берзи, док се у [GM+13] користи еволутивни алгоритам за подешавање параметара, а након тога се оптимизовани SVM модел примењује за решавање различитих регресионих проблема. Недавно је предложен и алгоритам променљиве претраге околина (енг. Variable neighboorhod search - VNS) за решавање овог проблема [CMBRM12]. Аутори рада су извели неколико експеримената над моделима учења са једним и више кернела. Резултати су упоређени са неким од тренутно најбољих метода описаних у [GA11].

2.3 Предложени EM метод

Основни EM алгоритам је описан у поглављу 1.5 и стога се сада излажу само елементи који прилагођавају основни алгоритам примени у решавању проблема подешавања параметара SVM. Слика 2.2 садржи шематски приказ EM алгоритма за подешавање параметара SVM.

2.3.1 Репрезентација решења и иницијализација

У фази иницијализације, популација од M EM тачака $\mathbf{p}$ бива формирана, што има за последицу и подешавање иницијалних вредности кернелских параметара. Свака EM тачка је H-димензионални вектор реалних вредности. У SKL моделу са кернелом радијалне основе i-та EM тачка је кодирана са $(p_i^1, p_i^2) \rightarrow (C, \sigma)$. У линеарном MKL моделу i-та EM тачка је кодирана параметрима на следећи начин: $(p_i^1, ..., p_i^{H=N_G}) \rightarrow (\alpha_1, ..., \alpha_{N_G})$, док су у RBF MKL моделу



Слика 2.2: Шематски приказ EM методе за подешавање параметара SVM

коефицијенти конусне комбинације и скалирајући фактори унакрсно кодирани: $(p_i^1, ..., p_i^{H=2N_G}) \rightarrow (\alpha_1, \sigma_1, ..., \alpha_{N_G}, \sigma_{N_G})$.

Процедура за постављање иницијалних вредности параметара се састоји из два корака. У првом се користи идеја детерминистичке хеуристике описане у [KL03]. Други корак је процедура која врши насумично разбацивање параметара добијених у првом кораку. Кроз даље излагање ће бити коришћена логаритамска нотација представљања вредности параметара. Разлог томе је њена погодност у случају рада са вредностима из домена великих реалних бројева којима припадају параметри који се подешавају.

У [KL03], аутори анализирају асимптотско понашање кернела са радијалном основом, тј. разматрају како параметри C и σ^2 интерагују када један од њих тежи бесконачности. Они такође дефинишу границе простора *добрих* решења према следећој формули:

$$\log \sigma^2 = \log C - \log \widetilde{C} \tag{2.12}$$

На тај начин, за сваку фиксирану вредност $\log \widetilde{C}$ постоји линеарна веза између $\log \sigma^2$ и $\log C$. Аутори доказују да када $\sigma^2 \rightarrow \infty$ SVM класификатор конвергира ка линеарном класификатору са казненим параметром $\log \widetilde{C}$. Сходно уведеној вези, аутори представљају једноставну хеуристику за претрагу простора свих могућих комбинација два поменута параметра. Техника из [KL03] је примењена у предложеној методи као основа за процедуру иницијализације. С обзиром на чињеницу да ова техника производи увек исто решење (поступак је детерминистички), предложени EM алгоритам укључује и корак који насумично помера (разбацује) добијено решење, тј. комбинацију два параметра тако што сваком од параметара додељује случајни шум. Следствено, уместо детерминистичке јединствене јединке, добија се читава популација различитих јединки разбацаних на случајан начин око простора добрих решења.

Према [KL03], за $\log \widetilde{C}$ се узима најбоља вредност параметра $\log C$ која се добија применом линеарног SVM класификатора. На сличан начин, у предложеном EM алгоритму се рачуна грешка класификације у 6 еквидистантних вредности параметра $\log C$ на интервалу $[-8, 2]$. Вредност $\log C$ која производи најмању грешку класификације се бира за $\log \widetilde{C}$.

Вредности које кодирају EM јединке се потом одређују на следећи начин:

- У случају да се користи линеарни класификатор, i-та EM тачка се иницијализује вредностима: $p_i^j = \log \widetilde{C} X_j$, $j = 1, ..., H$, где је $X_j \sim \mathcal{N}(1, 1)$ (X_j је случајна променљива из нормалне расподеле са наведеним параметрима).

- У случају да се користи кернел са радијалном основом, најпре се врши претрага комбинације параметара $(\log C, \log \sigma^2)$ која нуди највећу тачност класификације. Овај поступак се врши систематском провером вредности $\log \sigma^2$ дуж низа еквидистантних чворова на интервалу $[-8, 8]$. Вредност параметра $\log C$ се потом рачуна коришћењем формуле (2.12).

  Комбинација параметара која производи највећу тачност, означена је са $(\overline{\log C}, \overline{\log \sigma^2})$. Коначно, i-та EM тачка бива инцијализована на следећи начин: $p_i^{2j} = \log \widetilde{C} + \overline{\log \sigma^2} X_j$, $p_i^{2j+1} = \overline{\log \sigma^2} X_j$, $j = 1, ..., N_G$, где је као и пре, X_j случајна променљива из $\mathcal{N}(1, 1)$ расподеле.

Зарад поједностављеног извршавања EM процедура, иницијалне вредности параметара се скалирају на интервал $[0, 1]$. Касније, приликом рачунања функције циља, вредности се трансформишу назад на оригинални интервал и као такве се користе у тренингу SVM класификатора.

2.3.2 Функција циља

У процесу претраге оптималног скупа SVM параметара EM алгоритам је вођен функцијом циља. Она је мера квалитета појединачног решења представљеног EM тачком. Природан избор за функцију циља у случају SVM класификатора је оцена грешке генерализације, тј. оцена грешке класификације добијене на непознатим (новим) подацима. Општеприменљиви типови оцена су описани у Секцији 1.3.2. Детаљнији преглед оцена грешке генерализације специфичних за методу SVM се може наћи у [Joa00], док се у [DKP03] може видети преглед једноставних оцена прилагођених проблему подешавања параметара.

Слика 2.3 приказује шему процедуре рачунања функције циља. У прва два корака се врши трансформација координата EM тачке у SVM параметре и употреба параметара у рачунању вредности кернелске матрице. Након тога, потребно је израчунати функцију циља. Да би се извршило релевантно поређење са другим техникама, функција циља је за већину експеримената

рачуната коришћењем унакрсне провере, осим у експерименту где се EM алгоритам пореди са ACO техником, где је рачуната као грешка на засебном скупу података за проверу (погледати [ZCH10] и [Cha+02]). За инстанце проблема мањих димензија са хомогеним атрибутима, користи се техника унакрсне провере са 5 компоненти, док се за велике инстанце грешка рачуна као 5 пута поновљена 2-компонентна унакрсна провера. У првом случају, тренинг скуп се дели на 5 дисјунктних делова једнаких димензија, и онда се у 5 итерација сваки део користи за проверу, тј. за рачунање грешке класификације. На крају се узме просечна вредност од добијених 5 грешака. У другом случају се врши 2-компонента унакрсна провера 5 пута, а потом се укупна оцена рачуна као просек за тих пет извршавања.

улаз: $\mathbf{p_i}$, D_{tr}

```
1  r = преведиУПараметреКернела(p_i);
2  израчунајМатрицуКернела(r, D_tr);
3  p_i^obj = 0;
4  if укључена унакрсна провера then
5  |  if N_G == 1 then
6  |  |  делова = 5;
7  |  |  p_i^obj = унакрснаПровера(делова, D_tr);
8  |  else
9  |  |  делова=2;
10 |  |  for k ← 1 to 5 do
11 |  |  |  p_i^obj=p_i^obj+унакрснаПровера(делова, D_tr);
12 |  |  end
13 |  |  p_i^obj = p_i^obj/5;
14 |  end
15 else
16 |  p_i^obj = грешкаНаСкупуЗаПроверу();
17 end
```

Слика 2.3: Рачунање функције циља

2.3.3 Локална претрага

Процедура локалне претраге (енг. local search - LS) је подељена у две фазе (Слика 2.4): прва је одабир EM тачке на којој ће бити примењена LS, а друга је сам процес примене LS.

улаз: $\mathbf{p}$, D_{tr}

```
1  сортирајПоГрешциКласификацијеОпадајуће(p);
2  ind = -1;
3  if применљиваЛокалнаПретрага(p_1) then
4  |  ind = 1;
5  else if применљиваЛокалнаПретрага(p_2) then
6  |  ind = 2;
7  end
8  if ind== -1 then
9  |  return;
10 end
11 for k ← 1 to H do
12 |  for знак ← 0 to 1 do
13 |  |  корак = (знак-p_ind^k)/10;
14 |  |  побољшање = true;
15 |  |  while побољшање = true do
16 |  |  |  побољшање=false;
17 |  |  |  стараФункцијаЦиља=p_ind^obj;
18 |  |  |  староКоордината=p_ind^k;
19 |  |  |  p_ind^k = p_ind^k + корак;
20 |  |  |  новаФункцијаЦиља=функцијаЦиља(p_ind, D_tr);
21 |  |  |  if новаФункцијаЦиља < стараФункцијаЦиља then
22 |  |  |  |  побољшање = true;
23 |  |  |  else
24 |  |  |  |  p_ind^k = староКоордината;
25 |  |  |  |  p_ind^obj = стараФункцијаЦиља;
26 |  |  |  end
27 |  |  end
28 |  end
29 end
```

Слика 2.4: Локална претрага

Фаза одабира је слична оној описаној у [FKM13]. Кандидати за LS су најбоља и друга најбоља ЕМ тачка. Стога се ЕМ тачке најпре уређују у растућем поретку према вредности функције циља. Процедура "применљиваЛокалнаПретрага" проверава да ли ЕМ тачка испуњава неопходне услове да би на њој била примењена LS: да ли на дату тачку никад раније није био примењен поступак LS, или је LS био примењен раније, али се вредност функције циља у међувремену променила. У случају да најбоља тачка не испуњава неопходне услове, проверава се друга најбоља тачка. У случају да

ниједна не испуњава услове, LS се у тој итерацији алгоритма не примењује. Мотивација за овако стриктну селекцију тачака, на којима ће бити примењиван LS, је следећа:

1. Број рачунања функције циља је значајно смањен, што као последицу има и краће време извршавања;

2. Примена LS над свим, или већини тачака, би могла да смањи степен различитости тачака у оквиру популације па самим тим смањи и потенцијал за проналажење глобалног оптимума.

Друга фаза ове процедуре је примена локалне претраге на одабраној тачки. Алгоритам покушава да пронађе побољшање у оба смера почев од тренутне вредности сваке координате одабране тачке. Најпре се покушава према левој (вредности 0), а онда и према десној граници (вредности 1). Померање се врши повећавањем вредности дате координате за 1/10 преосталог интервала вредности на левој или десној страни. Када је побољшање пронађено, оно се одмах примењује, а потом се претрага наставља у истом смеру. Ако побољшање није пронађено, процес претраге се наставља у супротном правцу, али само једанпут. Након што се заврши побољшавање свих кодираних параметара, алгоритам LS се завршава.

2.4 Експериментални резултати

Процена квалитета предложеног приступа је направљена кроз пет засебних експеримената. Прва три су заснована на три колекције инстанци проблема малих и средњих димензија, са највише 60 атрибута. Број подешавајућих SVM параметара H је 2 у сва три случаја. Последња два експеримента разматрају примену линеарних кернела, кернела са радијалном основом и MKL модел на колекцији проблема великих димензија. Као што је описано у Секцији 2.1.3, број параметара је $H = N_G$ за линеарне, а $H = 2N_G$ за кернеле са радијалном основом.

Експеримент 1 Табела 2.1 садржи информације о колекцији од 13 тест проблема (означена као *прва експериментална колекција* у даљем тексту) на којој је базирано поређење са резултатима из [KL03] и [CMBRM12]. Табела

садржи следеће колоне: име тест проблема (*проблем*), број тренинг података (N_{tr}), број података за тестирање (N_{ts}) и број атрибута (N). Ове инстанце се могу наћи на UCI репозиторијуму [BL13]. У овом експерименту се, међутим, користи прочишћена колекција истих тест проблема описана у [ROM01], за коју је дато по 100 подела података на тренинг и тест скуп. На тај начин је омогућено непристрасно поређење резултата са другим предложеним алгоритмима у литератури. Као у [KL03] и [CMBRM12], користи се прва од датих 100 подела по свакој инстанци проблема[1].

У раду [Mul+01] је прва експериментална колекција коришћена при поређењу неколико класификационих метода заснованих на кернелском учењу, и то: методи подржавајућих вектора (SVM), кернелској фишеровој дискриминантној анализи (енг. Kernel fisher discriminant analysis - KFD), класификатору заснованом на функцији са радијалном основом (енг. radial basis function classifier), AdaBoost (AB) и регуларизованом AdaBoost (AB_R). Ниједна од ових метода се није показала систематски најбољом, што је и било очекивано, с обзиром да сви ови алгоритми имају сличне основе у виду кернелског учења. Међутим, KFD и SVM су произвели резултате вишег квалитета. У једном од каснијих истраживања [FH03] је показао да је SVM бољи од KFD на подскупу *прве експерименталне колекције*. Велики број истраживања, чији је опис ван научног фокуса овог рада, наводи на закључак да SVM често надмашује класификационе алгоритме са којима се пореди, чак и у својој чистој варијанти, без коришћења оптимизације параметара. Супериорност SVM класификатора, потврђена у тим истраживањима, подржава одлуку спроведену у овом раду да се предложени ЕМ алгоритам пореди само са другим алгоритмима за подешавање параметара SVM класификатора, а не и са другим класификационим методама.

С обзиром на хомогеност атрибута у *првој експерименталној колекцији*, примењен је RBF модел са једним кернелом. Процес претраге ЕМ алгоритма је вођен оценом грешке добијеном путем 5-компонентне унакрсне провере. На крају процеса претраге, када се достигну критеријуми завршетка, SVM параметри, кодирани најбољом ЕМ тачком, се користе у изградњи модела предвиђања. Модел се потом примењује над тест скупом како би се израчунала

[1] Интернет адреса: http://mldata.org/repository/tags/data/IDA_Benchmark_Repository/

Табела 2.1: Мали тест проблеми коришћени у првом и делом у другом експерименту

проблем	N_{tr}	N_{ts}	N
Banana	400	4900	2
Diabetes	468	300	8
Image	1300	1010	18
Splice	1000	2175	60
Ringnorm	400	7000	20
Twonorm	400	7000	20
Waveform	400	4600	21
German	700	300	20
Heart	170	100	13
Thyroid	140	75	5
Titanic	150	2051	3
Solar	666	400	9
Breast Cancer	200	77	9

грешка тестирања, која се касније користи у поређењу са другим методама.

Експеримент 2 Други експеримент је извршен на колекцији тест проблема која је подскуп прве колекције, с обзиром да се разматрају следећи проблеми: *Breast Cancer, Diabetes, Heart, Thyroid* и *Titanic*. Резултати над ових 5 тест проблема су упоређени са ACO техником за подешавање параметара SVM [ZCH10]. Као и у *првој експерименталној колекцији*, употребљена је прва од 100 подела на тренинг и тест скуп (погледати [ROM01]). Као што је поменуто у претходној секцији, начин рачунања функције циља је нешто другачији него у првом експерименту, јер се овде користи грешка на скупу података за проверу.

Експеримент 3 Трећа колекција малих тест проблема је коришћена у раду [PK10], и добијена је на захтев, директно од аутора рада. Сваки од 15 тест проблема из ове колекције је подељен на 5 различитих начина на тренинг и тест скуп (5 делова где је сваки коришћен један пут као тест скуп). Као што се може видети из Табеле 2.2, димензије проблема су врло сличне као у првом експерименту. Методологија провере и тестирања је иста као и у [PK10], што значи да је коришћена 5-компонентна унакрсна провера над тренинг скупом као функција циља. Након завршетка EM алгоритма, класификациони модел је изграђен помоћу најбоље EM тачке и после је тај модел коришћен за рачунање

Табела 2.2: Мали тест проблеми коришћени у трећем експерименту

проблем	N_{tr}	N_{ts}	N
Checkers	153	39	2
Spiral	465	117	2
Liver Disorders	276	69	6
Indians Diabetes	614	154	8
Three Of Nine	410	102	9
Tic Tac Toe	766	192	9
Breast Cancer	559	140	10
Parity Bits	819	205	10
Solar Flare	853	213	10
Cleveland Heart	216	54	13
Australian	552	138	14
German-org	800	200	24
Ionosphere	280	71	34
Tokyo	767	192	44
Sonar	166	42	60

грешке на тест скупу. Укупна оцена грешке је добијена као просек добијених грешака на 5 тест скупова, формираних поделом на 5 различитих начина.

Експерименти 4 и 5 Четврти и пети експеримент су засновани на великим тест проблемима[2] и експерименталној методологији описаној у [CMBRM12] и [GA11]. Насупрот малим тест проблемима, овде су атрибути хетерогени, тј. груписани у кластере атрибута. Последња колона Табеле 2.3 приказује кардиналности кластера (G). Примењено је вишекернелско учење са линеарним кернелима и кернелима са радијалном основом. Сваки кернел се бави појединачним кластером атрибута. Функција циља је рачуната као 5x2-компонентна унакрсна провера. С обзиром да подела оригиналног скупа података на тренинг и тест скуп није доступна, као и у [GA11] и [CMBRM12] и овде је подела извршена на случајан начин са односом 2:1 у корист тренинг скупа. Просечна грешка тестирања за већи број извршавања, са различитим иницијалним стањем генератора случајних бројева, је употребљена при поређењу са другим методама.

При одређивању експерименталног окружења било је битно одржати релативну униформност контролних параметара алгоритма кроз различите

[2]Интернет адресе: http://archive.ics.uci.edu/ml/datasets/Multiple+Features и http://archive.ics.uci.edu/ml/datasets/Internet+Advertisements

Табела 2.3: Велики тест проблеми у четвртом и петом експерименту

проблем	N_{tr}	N_{ts}	N	G
mfEO4	1333	667	427	(76, 64, 240, 47)
mfEO6	1333	667	649	(76, 64, 240, 47, 216, 6)
mfSL4	1333	667	427	(76, 64, 240, 47)
mfSL6	1333	667	649	(76, 64, 240, 47, 216, 6)
adv	2186	1093	1554	(457, 495, 472, 111, 19)

експерименте, али и истовремено реплицирати сличне или исте услове извршавања које су користили и алгоритми са којима се пореди. Посебно је важно користити исти или приближан број евалуација функције циља у свим алгоритмима који се пореде [ČLM12]. У неким случајевима, када не постоје додатне процедуре, нпр. локална претрага, ово је могуће постићи једноставним подешавањем исте величине популације и истог броја итерација. Овде то није случај, будући да предложени EM алгоритам користи и локалну претрагу.

Први скуп тест проблема је решен постављањем $M = 5$ и $N_{it} = 5$. Ови контролни параметри су подешени са жељом да се достигне приближан број извршавања функције циља (54) као што је то случај и у [KL03] и [CMBRM12]. Овај број подразумева (урачунава) и израчунавања локалне претраге. Преостала два експеримента на мањим инстанцама користе доста већи број израчунавања функције циља. У [ZCH10] је коришћена популација од 80 мрава. Следствено, предложени алгоритам користи $M = 80$ у овом експерименту. Пошто код ACO технике критеријум завршетка није заснован на максималном броју итерација, а такође у поменутом раду није приказан ни укупан број евалуација, предложени EM користи $N_{it} = 1000$, док времена извршавања су приказана за све упоређене алгоритме. У [PK10] је употребљен велики број извршавања функције циља у тзв. еволутивној стратегији (ES). ES је извршавао 1000 итерација, и у свакој итерацији је рачуната функција циља за 10 кандидатских решења, што у збиру даје 10000 израчунавања. Имајући ово на уму, експериментално окружење за EM је подешено на следећи начин: $M = 100$, $N_{it} = 100$. Прави број извршавања је обично био нешто виши од 10000 него у [PK10] због LS. У случају великих тест проблема, тј. четвртог и петог експеримента, M је постављен на 8, док је N_{it} једнак 15. Број израчунавања је у овим експериментима осцилирао око 600, што је укупан број израчунавања

коришћен у [CMBRM12].

Предложени EM алгоритам је написан у програмском језику C, док је изворни код преведен Visual Studio 2010 преводиоцом. Сви тестови су извршени на Intel Xeon E5410 @2.34GHz са 4GB RAM и Windows 7 оперативним системом.

Табела 2.4 приказује резултате првог експеримента, тј. поређење између предложеног метода и још два метода из литературе. Колоне означене са KL и VNS реферишу на резултате добијене у [KL03] и [CMBRM12] након једног извршавања. Број израчунавања функције циља, време извршавања, и број итерације у којој је пронађено решење је приказано у последње три колоне. Поред ових, у табели су приказане још три колоне које представљау скалиране грешке за сваки од алгоритама: $\overline{KL}$, $\overline{VNS}$ и $\overline{EM}$. Ове вредности су употребљене у статистичкој анализи резултата. Оне су добијене скалирањем грешака класификације сва три метода на интервал [0,1], за сваку инстанцу проблема засебно. На тај начин, најгори од три алгоритма добија вредност 1, док најбољи добија вредност 0. Да би се избегло дељење нулом, скалирање није извршено када су сва три алгоритма достигла исту грешку класификације. У тим случајевима, скалирана грешка је постављена на 0 за сва три алгоритма (погледати инстанцу Solar). Мотивација за скалирањем грешке лежи у чињеници да за различите тест проблеме грешка класификације може имати драстично различите вредности. Стога, без скалирања, неки тест проблеми би имали већи значај од других у целокупном поређењу. Резултати показују да предложени EM алгоритам даје најбоља решења на 10 од 13 тест проблема, и производи друго најбоље решење на преостала 3.

Резултати другог експеримента су представљени у Табели 2.5. Прве 3 колоне приказују грешку класификације за SVM подешен претрагом мреже (GS), мрављим колонијама (ACO) и предложеним EM алгоритмом. Резултати за EM и ACO су преузети из [ZCH10]. Као и у претходном експерименту, дате су и вредности скалираних грешака, док су на крају дата времена извршавања (у секундама). Може се видети да EM даје боља решења него GS и ACO на свим разматраним инстанцама. Времена извршавања иду такође у прилог EM алгоритму. Ово је последица тога што EM и ACO имају различите критеријуме завршетка, тј. EM завршава извршавање након одређеног броја итерација, док се ACO зауставља када достигне одређени ниво прецизности.

Табела 2.4: RBF кернел на малим тест проблемима - Експеримент 1

проблем	KL	VNS	EM	$\overline{KL}$	$\overline{VNS}$	$\overline{EM}$	$Eval_{EM}$	$t_{EM}(sec)$	$Iter_{found}$
Banana	11.59	11.61	11.57	0.48	1	0	57	7.35	1/5
Diabetes	24	24.67	23.33	0.50	1	0	69	9.65	1/5
Image	5.84	2.38	2.18	1	0.06	0	60	34.66	3/5
Splice	10.53	9.93	10.16	1	0	0.38	49	39.34	3/5
Ringnorm	1.44	1.7	1.63	0	1	0.73	61	8.44	1/5
Twonorm	2.47	2.77	2.36	0.27	1	0	53	10.48	2/5
Waveform	11.39	10.46	11.22	1	0	0.81	58	13.6	2/5
German	21.33	21.33	20.33	1	1	0	50	21.88	2/5
Heart	21	20	19	1	0.5	0	35	0.61	1/5
Thyroid	5.33	5.33	4	1	1	0	47	0.52	2/5
Titanic	22.92	22.92	22.57	1	1	0	40	13.82	1/5
Solar	34.5	34.5	34.5	0	0	0	47	11.52	2/5
Breast Cancer	29.87	28.57	28.57	1	0	0	41	4.88	2/5

Табела 2.5: RBF кернел на малим тест проблемима - Експеримент 2

проблем	GS	ACO	EM	$\overline{GS}$	$\overline{ACO}$	$\overline{EM}$	t_{GS}	t_{ACO}	t_{EM}
Breast Cancer	25.97	25.97	23.38	1	1	0	2547.3	1437.8	270.44
Diabetes	23.33	23	22.67	1	0.5	0	29078	19298	1837.46
Heart	19	16	15	1	0.25	0	1446.4	519.58	270.71
Thyroid	4	2.67	1.33	1	0.5	0	702.89	666.2	163.43
Titanic	22.57	22.57	21.84	1	1	0	639.95	429.22	1437.42

Резултати трећег експеримента су приказани у Табели 2.6. Друга и трећа колона представљају просечну грешку класификације GS алгоритма и еволутивне стратегије (ES) из [PK10]. Просечне вредности грешке EM алгоритма су приказане у четвртој колони. Као и у прва два експеримента, приказане су и скалиране грешке. Последње три колоне приказују број израчунавања функције циља ($Eval_{EM}$), време извршавања (t_{EM}) и просечан број итерација за достизање коначног решења ($Iter_{found}$). Резултати сугеришу да EM алгоритам систематично производи најбоља решења у 13 од 15 случајева. На једном од преосталих тест проблема, *Parity Bits*, EM даје решење подједнако добро као и GS, а само код једног тест проблема *Checkers*, EM је лошији од друга два алгоритма. Просечан број израчунавања функције циља је нешто виши од 10000 (што је број евалуација у [PK10]), али се испоставља да ово мало прекорачење у броју израчунавања нема значајан ефекат на квалитет решења с обзиром да су решења у просеку пронађена за мање од 33 итерације (од укупно 100 итерација).

Четврти експеримент је заснован на тест проблемима великих димензија где

Табела 2.6: RBF кернел на малим тест проблемима - Експеримент 3

проблем	GS	ES	EM	$\overline{GS}$	$\overline{ES}$	$\overline{EM}$	$Eval_{EM}$	$t_{EM}(sec)$	$Iter_{found}$
Checkers	16.68	16.18	43.09	0.02	0	1	10247.8	53.2	9.2/100
Spiral	0	0	0	0	0	0	11543.4	603.2	44.4/100
Liver Disorders	38.26	33.33	26.96	1	0.56	0	11510.8	383.47	32/100
Indians Diabetes	35.03	26.7	22.4	1	0.34	0	11413.8	1749.21	52.4/100
Three Of Nine	46.49	0	0	1	0	0	11616.4	495.95	14.8/100
Tic Tac Toe	34.66	0.31	0	1	0	0	11335.6	2475.52	14/100
Breast Cancer	13.59	5.44	3.86	1	0.16	0	11526	663.49	25.2/100
Parity Bits	51.95	24.22	51.95	1	0	1	10232.4	2598.03	1/100
Solar Flare	19.13	19.04	17.26	1	0.95	0	110861.4	6173.75	11.4/100
Cleveland Heart	44.44	21.85	14.81	1	0.24	0	10622.8	132.02	9.4/100
Australian	44.49	44.49	28.12	1	1	0	10747	1150.56	2.2/100
German-org	29.9	29.7	24	1	0.97	0	10687.8	2360.07	5/100
Ionosphere	33.9	4.84	4.84	1	0	0	11489.4	199.12	22/100
Tokyo	18.98	8.34	7.92	1	0.04	0	11283	2182.48	25.4/100
Sonar	29.33	11.08	11.07	1	0	0	11455	93.29	56.2/100

се решења добијају подешавањем параметара MKL модела. Просечни резултати предложеног EM алгоритма су упоређени са најбољим (IGA_b), медијаном (IGA_m), и најгорим решењем (IGA_w) добијеним при извршавању 12 различитих модела описаних у [GA11]. Поред тога, предложени EM алгоритам је упоређен са угњежденим VNS моделом предложеним у [CMBRM12]. Табела 2.7 приказује добијене резултате, који потврђују да је предложени алгоритам конкурентан са осталим алгоритмима, посебно код проблема mfEO4 и mfSL4.

Табела 2.7: Више линеарних кернела - Експеримент 4

проблем	$IGA_{(b,m,w)}$	VNS	EM	$Eval_{EM}$	$t_{EM}(sec)$	$Iter_{found}$
mfEO4	(1.99, 2.15, 4.22)	2.34	2.14	539.7	1567	3.4/15
mfEO6	(1.61, 1.76, 3.1)	1.46	2.02	654.8	2076.5	4/15
mfSL4	(4.82, 5.11, 9.46)	6.19	5.56	575.2	1698	3.7/15
mfSL6	(2.19, 2.54, 10.82)	2.3	5.04	721.5	2510.6	3.1/15
adv	(3.41, 3.72, 4.9)	3.22	4.76	336.4	4391.8	2.6/15

Последњи, пети експеримент је заснован на употреби кернела са радијалном основом (2.11). Као у претходном експерименту, EM решење се пореди са најбољим, медијаном и најгорим решењем од 12 алгоритама предложених у [GA11]. Додатно, дато је и поређење са две варијанте угњеждених VNS модела, VNS_1 и VNS_2 [CMBRM12]. Резултати показују да је EM бољи од осталих метода на 3 од 5 инстанци, док се понаша конкурентно на преостале две.

Зарад показивања статистичке значајности квалитета резултата EM

Табела 2.8: Више RBF кернела - Експеримент 5

проблем	$IGA_{b,m,w}$	VNS_1	VNS_2	EM	$Eval_{EM}$	$t_{EM}(sec)$	$Iter_{found}$
mfEO4	(0.67, 0.96, 2.18)	0.74	0.72	0.6	519.8	1958.6	5.9/15
mfEO6	(0.58, 0.67, 7.22)	0.65	0.53	0.48	633.6	2525.6	2.6/15
mfSL4	(1.43, 1.63, 4.6)	1.58	1.4	1.39	507	2001.2	4.2/15
mfSL6	(0.97, 1.25, 7.25)	0.99	0.95	1.45	624.5	2660.9	7.9/15
adv	(3.81, 4.34, 11.88)	3.24	3.39	4.68	530	7608.6	6/15

алгоритма, извршена је статистичка анализа. У литератури су предложене многобројне технике за поређење резултата извршавања различитих алгоритама. На пример, у [HH09] је SVMr модел (SVM за регресију) оптимизован техником мравље колоније, а потом је добијени модел употребљен за предвиђање курса валута, и притом упоређен са другим алгоритмима. У том раду су аутори користили статистичку технику уведену у [DM02], која је специјализована за одређивање разлика у моћи предвиђања када су у питању проблеми прогнозирања. Нажалост, ова статистичка методологија не одговара разматраном проблему класификације. Праћена је друга методологија, слична описаној у [FKM13]. Најпре је извршен Shapiro-Wilk тест како би се испитала нормалност скалираних грешака. У све три мале експерименталне колекције, Shapiro-Wilk тест је показао да подаци не прате нормалну расподелу. Ово је искључило ANOVA тест из даљег разматрања, тако да је примењен Kruskal-Wallis H тест за поређење квалитета алгоритама.

У првом експерименту, нулта хипотеза је гласила да не постоји значајна разлика између квалитета KL, VNS и EM алгоритама. Тест је показао да се нулта хипотеза одбацује са H(2)=11.68, p=0.003. Такође су израчунати следећи рангови алгоритам: ранг 25.46 за KL, 22.69 за VNS и 11.85 за EM. Из даљег разматрања је потом избачен најлошији, KL алгоритам, а нови Kruskal-Wallis тест је примењен на VNS и EM. Као и раније, нулта хипотеза је одбачена са H(1)=6.357, p=0.012, и просечним ранговима од 17 за VNS и 10 за EM. Слично је у другом експерименту утврђена значајна статистичка разлика између GS, ACO и EM, са H(2)=12.149, p=0.002 и ранговима од 12, 9 и 3 за GS, ACO и EM редом. Даље поређење ACO и EM је показало значајну разлику међу алгоритмима са H(1)=7.813, p=0.005 и просечним ранговима од 3 за EM и 8 за ACO. Коначно, у трећем експерименту, нулта хипотеза је такође одбачена

са H(2)=22.200, p=0.00002, и просечним ранговима: 34.50 за GS, 21.20 за ES и 13.30 за ЕМ. Друга фаза тестирања је показала да је ЕМ значајно бољи од ES са H(1)=6.957, p=0.008, и просечним ранговима од 19.33 за ES и 11.67 за ЕМ. На основу извршених статистичких тестова може се закључити да је ЕМ статистички значајно бољи него сви алгоритми са којима је поређен, у оквиру сва три експеримента.

За експерименте са инстанцама великих димензија, вредности са којима се алгоритам ЕМ пореди представљају здружене резултате више алгоритама, тј. њихове максималне, минималне, и вредности медијане. Стога, статистичка анализа није извршена за последња два експеримента.

2.5 Завршна разматрања

Моћ предвиђања SVM је високо зависна од унутрашње параметарске структуре SVM модела. Традиционални приступ у решавању проблема подешавања параметара, тзв. претрага мреже (GS), се понаша добро у случају да је скуп подешавајућих параметара мале кардиналности. С обзиром на то да су параметри реалне вредности, ефикасност GS драстично опада увођењем новог параметра или повећањем прецизности претраге. У овом поглављу је представљен ефикасан алгоритам за подешавање параметара SVM.

Интерна репрезентација ЕМ тачака је заснована на вектору реалних вредности, што се показало као врло практично својство у решавању проблема подешавања параметара. Ово је омогућило *гладак* прелаз из простора ЕМ тачака у простор параметара, с обзиром да су оба простора над скупом реалних вредности. Предложени ЕМ алгоритам користи хеуристичку иницијализацију која значајно смањује време потребно за налажење квалитетних региона претраге. Функција циља је рачуната техником унакрсне провере, што је омогућило добру оцену грешке над тест скупом података. Специјализована локална претрага се примењује селективно само на најбоље тачке, што смањује време извршавања и спречава *заглављивање* алгоритма у локалном оптимуму.

У експерименталној евалуацији коришћени су различити скупови података и различити модели учења кернела. Три колекције малих и средњих инстанци са највише 60 атрибута су коришћене у поређењу ЕМ са неким од тренутно најбољих алгоритама из литературе. Спроведена три експеримента над ове

три колекције су заснована на учењу једног кернела и кернелским моделима са линеарном функцијом и функцијом са радијалном основом. Предложени EM алгоритам је надмашио резултате GS алгоритма и методе променљивих околина на 10 од 13 инстанци проблема, док се показао као бољи од GS и еволутивне технике на 13 од 15 инстанци. У поређењу са алгоритмом мравље колоније, EM је био бољи у свих 5 од 5 посматраних случајева.

Тестирање је извршено и на инстанцама великих димензија са хетерогеним скупом атрибута, са максималном кардиналношћу од 1554. Резултати су показали да је предложени метод бољи него 14 успешних алгоритама из литературе на 3 од 5 инстанци, у случају када се користи RBF кернел и конкурентан са осталим алгоритмима у случају да се користи линеарни кернел. Ово је довело до закључка да се EM понаша боље у раду са RBF кернелима.

Као правци даљег развоја, могући су експерименти са различитим процедурама локалне претраге, нпр. претрага која би користила претрагу мреже (GS) са адаптирајућим степеном прецизности, случајна претрага, итд.

Поглавље 3

Примена ЕМ у одабиру атрибута

3.1 Проблем одабира атрибута

Проблем одабира атрибута (енг. feature selection problem - FS) подразумева тражење оптималног подскупа атрибута из датог скупа свих атрибута. Оптималност се мери према циљном критеријуму, који у случају класификационих проблема може бити грешка (тачност) класификације, ефикасност модела, комбинација неких од ових критеријума, итд. Методе за одабир атрибута (енг. feature selection algorithm - FSA) су примењене као подршка у решавању великог броја проблема са реалним применама. Неколико прирена FSA су: FSA заснован на расплинутим-грубим скуповима (енг. fuzzy-rough set) са применом у анализи микросеквенци и слика [PVP11], FSA који користи оптимизацију ројевима у циљу решавања проблема препознавања лица [AMM12], за одређивање тумор маркера [MAT10], двофазни FSA за категоризацију текста [MLY11], у [Hua+12] FSA оптимизован мрављим колонијама се користи за класификацију електромиографских сигнала, док се у [Yan+11] FSA заснован на биномном тестирању хипотеза, користи у циљу откривања непожељних порука електронске поште. Према начину решавања проблема, постоје три типа FS алгоритама [GE03]:

1. *омотач*-методе (енг. wrapper-methods) које користе класификациони алгоритам као "црну кутију" у циљу одређивања квалитета подскупа одабраних атрибута;

2. *филтер*-методе (енг. filter-methods) које формирају подскуп одабраних

атрибута у фази препроцесирања, тј. пре извршења класификационог алгоритма, па самим тим ове методе и не зависе од конкретне имплементације класификатора;

3. *угњежедене*-методе (енг. embedded-methods) које формирају подскуп одабраних атрибута у фази тренирања класификационог алгоритма.

У овом поглављу се разматра алгоритам који се заснива на првој групи метода, тзв. омотач-методама. За класификацију се користе метода најближих суседа са 1 суседом (енг. 1-nearest-neighbor 1-NN) и метода подржавајућих вектора. Основе k-NN су уведене у Секцији 1.2.1. Када је у питању SVM, основе су дате у Секцији 1.2.3, док се напреднији аспекти разматрају у Секцији 2.1.1.

Пример У овом примеру се приказује начин функционисања 1-NN класификатора када се примењује одабир атрибута, односно какав је ефекат подскупа употребљених атрибута на тачност класификације. За ту намену се користи познати тест проблем који садржи информације о цвету Ирис [BL13]. У примеру се користи 30 од укупно 150 података о цвету Ирис и они су приказани у Табели 3.1. 24 податка се користе у фази учења као тренинг подаци, док се преосталих 6 користи за тестирање квалитета класификације. Сваки податак се састоји од 4 атрибута, односно својстава цвета: ширина и висина чашице, и ширина и висина латице, редом означених са *slen*, *swidth*, *plen* и *pwidth*. Сваки податак садржи и тип цвета која узима 3 различите вредности (колона под називом *class*): 1 одговара типу *iris setosa*, 2 типу *iris versicolour*, а 3 типу *iris virginica*. 1-NN додељује сваком од 6 тест података класу најближег суседа из тренинг скупа података, при чему се као мера блискости користи Еуклидско растојање. За сваки тест податак је приказана предвиђена класа за различите подскупове употребљених атрибута, где се са 0 означава да атрибут на датој позицији није одабран, а са 1 да јесте одабран. На пример, 1101 означава подскуп атрибута сачињен од свих атрибута осим трећег (*plen*). Тачност класификације се рачуна као број података за које је предвиђена класа била једнака правој класи подељен са укупним бројем тест података. Може се видети да различити подскупови атрибута производе различите тачност, на пример, одабир само другог атрибута производи тачност од 66%, док одабир свих атрибута или само последњег атрибута даје тачност од 100%. Иако пун

скуп атрибута производи максималну тачност, обично се преферирају мањи скупови атрибута који омогућавају највећу или довољно велику тачност. У овом примеру је то подскуп атрибута који се састоји само од последњег атрибута.

Табела 3.1: Ирис тест проблем

тренинг подаци									
slen	swidth	plen	pwidth	класа	slen	swidth	plen	pwidth	класа
4.9	3.0	1.4	0.2	1	5.7	2.8	4.5	1.3	2
4.7	3.2	1.3	0.2	1	6.3	3.3	4.7	1.6	2
4.6	3.1	1.5	0.2	1	4.9	2.4	3.3	1.0	2
5.0	3.6	1.4	0.2	1	5.2	2.7	3.9	1.4	2
4.6	3.4	1.4	0.3	1	6.3	3.3	6.0	2.5	3
5.0	3.4	1.5	0.2	1	5.8	2.7	5.1	1.9	3
4.4	2.9	1.4	0.2	1	6.3	2.9	5.6	1.8	3
4.9	3.1	1.5	0.1	1	6.5	3.0	5.8	2.2	3
7.0	3.2	4.7	1.4	2	7.6	3.0	6.6	2.1	3
6.4	3.2	4.5	1.5	2	7.3	2.9	6.3	1.8	3
6.9	3.1	4.9	1.5	2	6.7	2.5	5.8	1.8	3
5.5	2.3	4.0	1.3	2	7.2	3.6	6.1	2.5	3
тест подаци					предвиђена класа за одабране атрибуте				
slen	swidth	plen	pwidth	класа	1100	0100	1101	1111	0001
5.1	3.5	1.4	0.2	1	1	1	1	1	1
5.4	3.9	1.7	0.4	1	1	1	1	1	1
6.5	2.8	4.6	1.5	2	3	2	2	2	2
6.6	2.9	4.6	1.3	2	3	1	2	2	2
7.1	3.0	5.9	2.1	3	2	3	3	3	3
4.9	2.5	4.5	1.7	3	3	1	2	3	3
тачност класификације:					0.50	0.66	0.83	1.00	1.00

Сада ће бити приказана нека теоријска разматрања везана за тежину проблема одабира атрибута [LV91; VHM94; AK98]. Формална дефиниција проблема минималног скупа атрибута [VHM94] из перспективе *омотач* методе је следећа:

Дефиниција 7. *Нека је дат скуп тренинг података D_{tr} који се састоји од N_{tr} уређених парова облика $(\mathbf{x}^{(i)}, y^{(i)}), i = 1, ..., N_{tr}$, где је $\mathbf{x}^{(i)} \in \mathbf{Q}^N$ N-димезионални вектор атрибута, а $y^{(i)} \in \{-1, 1\}$ одговарајућа класа. Скуп података се може описати матрицом $\mathbf{A}$ рационалних вредности димензије $N_{tr} \times (N+1)$. Проблем налажења минималног скупа атрибута (енг. Mimimum feature set - MIN FS) се своди на одређивање вектора коефицијената $(w_0, ..., w_N) \in \mathbf{Q}^{N+1}$ који описује хиперраван и раздваја све податке из полазног тренинг скупа према класи. При том важи да MIN FS користи минималан број ненула коефицијената w_j, $j =$*

$1, ..., N$ *за представљање хиперравни. У матричној нотацији, проналажење хиперравни одговара решавању система* $\mathbf{Aw} > 0$.

У [LV91] је доказано да је варијанта MIN FS проблема, у којој су атрибути бинарне вредности, NP-тежак. У [VHM94] је показано тврђење да је MIN FS дат претходном дефиницијом NP-тежак и тежак за апроксимацију бар колико је тежак и проблем покривања скупа. С обзиром на значај овог проблема, у наставку је приказан доказ да је MIN FS NP-тежак [VHM94]. Пре тога, потребно је увести неколико дефиниција.

Дефиниција 8. *Проблем минимизације* $\mathcal{M}$ *има следећу форму. За дати пар предиката P и Q, целобројну функцију трошка k и за задато x које задовољава предикат P(x) (тест проблем), потребно је пронаћи такво y за које важи Q(x,y) (y је решење тест проблема x) и при том је вредност k(x,y) минимална.*

Дефиниција 9. *Полиномски апроксимативни алгоритам (енг. Polynomial-time approximation algorithm - PTAA)* $\mathcal{A}$ *за проблем минимизације* $\mathcal{M}$ *је полиномски алгоритам који на улазу прихвата тест проблем x, за који важи P(x), а на излазу производи y за које важи Q(x,y). Каже се да* $\mathcal{A}$ *апроксимира* $\mathcal{M}$ *са фактором* $\phi(x)$ *ако важи да* $k(x, y) \leq \phi(x)k(x, z)$ *за било које z такво да важи Q(x,z). Проблем* $\mathcal{M}$ *се може апроксимирати са фактором* $\phi(x)$ *ако постоји PTAA који га апроксимира са фактором* $\phi(x)$.

Дефиниција 10. *Полиномско пресликавање проблема* $\mathcal{M}$ *у проблем* $\mathcal{M}'$*, које задржава трошак (полиномска редукција), је пар функција* (t_1, t_2) *са следећим својствима (са x је означен тест проблем за* $\mathcal{M}$*):*

1. t_1 *пресликава све тест проблеме* $\mathcal{M}$ *у тест проблеме за* $\mathcal{M}'$. t_2 *пресликава све парове (y,x), где у представљају решења за* $t_1(x)$ *(и проблем* $\mathcal{M}'$*) у решења за x (проблем* $\mathcal{M}$*).*

2. t_1 *и* t_2 *се могу израчунати у полиномском времену.*

3. *Ако решење x има трошак k, онда* $t_1(x)$ *има решење са трошком од најмање k.*

4. *Ако је y решење за* $t_1(x)$ *са трошком k, онда* $t_2(y, x)$ *има трошак од највише k.*

Ако постоје полиномске редукције из $\mathcal{M}$ у $\mathcal{M}'$ и PTAA, који апроксимира $\mathcal{M}'$ са фактором $\phi(x')$, онда постоји и PTAA који апроксимира $\mathcal{M}$ са фактором $\phi(t_1(x))$, где је $x' = t_1(x)$. Из овога се може закључити да се редукцијом почетног проблема на неки циљни проблем може одредити фактор апроксимације за почетни проблем ако је он познат за циљни. За ту намену ће се користити проблем минималног покривања скупа, за који је показано да је NP-тежак проблем.

Дефиниција 11. *Минимално покривање скупа (енг. Minimum set cover MIN SC) је проблем дефинисан на следећи начин. Нека је дат коначни скуп S и колекција подскупова скупа S означена са C. Минимално покривање скупа S је скуп $C' \subseteq C$ такав да $\cup C' = S$ и при том је кардиналност C' минимална.*

Теорема 2. *Постоји полиномска редукција (свођење) која задржава трошак између проблема минималног покривања скупа и проблема минималног скупа атрибута. Редукција је таква да се тест проблеми MIN CS означени са (S, C) пресликавају у тест проблеме MIN FS означене са $(N_{tr}, N, \mathbf{A})$ где је $N_{tr} = |S| + 1$.*

Доказ. Нека је (S, C) тест проблем за MIN SC. При том је $N_{tr} = |S| + 1$, а $N = |C|$. Елементи скупа S су означени са $s_1, s_2, ..., s_{N_{tr}-1}$, а елементи скупа C са $c_1, c_2, ..., c_N$. Funkcija $t_1(S, C) = (N_{tr}, N, \mathbf{A})$ пресликава тест проблем MIN SC у матрицу $\mathbf{A}$ тако да су елементи матрице $A_{ij}, i = 1, ..., N_{tr}, j = 1, ..., N + 1$ одређени на следећи начин:

$$A_{ij} = \begin{cases} 1, & i < N_{tr}, j \leq N, s_i \in c_j \\ 0, & i < N_{tr}, j \leq N, s_i \notin c_j \\ 0, & i = N_{tr}, j \leq N \\ -1, & i < N_{tr}, j = N + 1 \\ 1, & i = N_{tr}, j = N + 1 \end{cases} \tag{3.1}$$

Решавање система $\mathbf{Aw} > 0$ је стога еквивалентно решавању:

$$\sum_{j=1}^{N} \mathbb{1}\{s_i \in c_j\} w_j > w_{N+1} > 0, \; i = 1, ..., N_{tr} - 1 \tag{3.2}$$

Ознаком $\mathbb{1}$ је представљена индикаторска функција која узима вредност 1 у случају да је аргумент израз тачан, односно вредност 0 у супротном.

Претпоставимо да (S, C) има решење са трошком k, што значи да постоји $C' \subseteq C$ који је покривач скупа S, и $|C'| = k$. Дефинишимо вектор $\mathbf{w}$ на следећи начин:

$$w_j = \begin{cases} 1, & j \leq N, c_j \in C' \\ 0, & j \leq N, c_j \notin C' \\ 0.5, & j = N+1 \end{cases} \tag{3.3}$$

С обзиром да је C' покривач скупа S, за сваки елемент тог скупа $s_i, i = 1, ..., N_{tr}$ важи да припада неком $c_j \in C'$. Зато се, након уврштавања (3.3) у (3.2), добија да за свако $i = 1, ..., N_{tr} - 1$ важи:

$$\sum_{j=1}^{N} \mathbb{1}\{s_i \in c_j\} w_j = \sum_{j=1}^{N} \mathbb{1}\{s_i \in c_j\} \mathbb{1}\{c_j \in C'\} > 1 > w_{n+1} = 0.5 \tag{3.4}$$

На тај начин је показано да за тест проблем $t_1(S, C) = (N_{tr}, N, \mathbf{A})$ постоји решење са трошком k.

Може се показати да важи и други смер. За све $\mathbf{w} \in Q^{N+1}$, нека је дефинисана функција:

$$t_2(\mathbf{w}, S, C) = \{c_j | w_j > 0, j = 1, ..., N\}. \tag{3.5}$$

Претпоставимо да је $\mathbf{w}$ решење тест проблема $t_1(S, C) = (N_{tr}, N, \mathbf{A})$ са трошком k. Нека је $C' = t_2(\mathbf{w}, S, C)$. Из дефиниције t_2 је јасно да $C' \subseteq C$ и да је $|C'| \leq k$. Пошто важи $\mathbf{Aw} > 0$, то значи да су задовољене све неједнакости дефинисане системом (3.2), па за свако i важи да постоји неко j такво да $s_i \in c_j$ и $w_j > 0$. Када је $w_j > 0$ то значи да $c_j \in C'$, па је сваки елемент скупа S покривен неким c_j. То значи да је C' покривач скупа S, па је $\mathbf{w}$ решење за тест проблем (S, C) са највише трошком k. □

Последица 1. *Проблем минималног скупа атрибута (MIN FS) је NP-тежак.*

Доказ. С обзиром да је проблем покривања скупа NP-тежак проблем, на основу Теореме 2 се закључује да је и проблем минималног скупа атрибута NP-тежак проблем. □

3.2 Претходни резултати

Омотач методе за одабир атрибута се често сматрају за најинтуитивније приступе, будући да се поступак одређивања подскупа атрибута заснива на утврђивању његове мере квалитета и класификационог алгоритма који се користи као "црна кутија". С обзиром да је проблем NP-тежак, при његовом решавању се могу очекивати изазови у погледу временске сложености. У циљу савладавања сложености, предложене су многобројне рачунарски ефикасне технике. У [NF77], предложен је егзактан метод заснован на гранању са одсецањем. Аутори су показали да метод прави значајну рестрикцију простора претраге док гаранција оптималности и даље важи. Међутим, гаранција оптималности је условљена монотоношћу мере удаљености коју претрага користи. Са порастом простора за реалне примене, ефикасност је постала још већи изазов, тако да су почели да се употребљавају и неегзактни решавачи. Предност неегзактних решавача је у већини случајева њихова ефикасност. Са друге стране, недостатак је што неегзактни решавачи не гарантују оптималност решења.

У [SS89] и [YH98] су предложени генетски алгоритми за одабир атрибута. Још један генетски алгоритам који врши истовремену вертикалну редукцију (одабир атрибута) и хоризонталну редукцију, или тзв. одабир слогова је предложен у [KJ99]. Неколико метахеуристичких приступа, попут табу претраге [ZS02] и оптимизацијаројевима [AMM12; CSC12; MAT10], је предложено у литератури. У [Lin+08], аутори су развили метод заснован на симулираном каљењу за истовремено подешавање параметара SVM и одабир атрибута. У [KI+10] је предложена *омотач* техника заснована на вештачкој неуронској мрежи. У [VSK12] је дата дискусија о вишециљној (енг. multiobjective) природи проблема одабира атрибута. Направљен је преглед неколико могућих циљева у процесу одабира атрибута попут дискриминишуће моћи скупа атрибута, перформанси модела, кардиналности скупа одабраних атрибута итд. Аутори такође предлажу алат који користи расплинуте логике и на тај начин избегава класичан приступ у вишециљној оптимизацији заснован на додељивању тежина сваком од циљева, а након тога примењују оптимизацију мрављим колонијама како би решили проблем.

У [SL11] аутори предлажу метахеуристику заснована на електромагнетизму за одабир атрибута, и показују да је предложени метод значајно тачнији од неколико других метода из литературе, а упоредив са методом подржавајућих вектора и *омотач* методом која користи генетски алгоритам и 1-NN класификатор. Аутори су показали да њихова метода има мање просечно време извршавања од генетског алгоритма. Међутим, време извршавања њихове ЕМ методе је и даље значајно дуже у односу на већину *филтер* техника. У [UMC11] је предложена хибридна *филтер-омотач* метода која користи оптимизацију ројевима и методу подржавајућих вектора. Даљи ток излагања у овом поглављу је организован на следећи начин: наредна секција приказује дизајн ЕМ алгоритма за проблем одабира атрибута, након тога су изложени експериментални резултати који демонстрирају квалитет предложене методе. У последњој секцији су дата завршна разматрања и правци даљег развоја у контексту проблема одабира атрибута и оптимизације.

3.3 Предложени ЕМ метод

Шема предложеног ЕМ метода је дата на Слици 3.1. Алгоритам захтева 3 контролна параметра, и то два која су присутна и у основном ЕМ алгоритму, N_{it} - број итерација, и M - величину популације, и додатни параметар α, који описује ефекат скалирања решења. Вредности скалирајућег параметра припадају интервалу $[0, 1]$.

У фази иницијализације се врши припрема улазног скупа података и генерисање иницијалних ЕМ тачака. Након тога, алгоритам улази у главни (спољни) циклус у којем се извршавају сви најбитнији елементи оптимизације. Најпре се врши рачунање функција циља за све ЕМ тачке. Након тога се бира једна ЕМ тачка над којом ће се применити локална претрага и скалирање решења. На крају се рачунају налектрисања и силе које дејствују на тачке, и врши померање у складу са добијеним силама. Главни циклус се завршава када се достигне критеријум завршетка, у овом случају је то максимални број итерација.

```
   улаз : N_it, M, α
 1 p = креирајИницијалнеЕМТачке(M);
 2 for iter ← 1 to N_it do
 3 |   for i ← 1 to M do
 4 |   |   рачунајФункцијуЦиља(p_i);
 5 |   end
 6 |   примениЛокалнуПретрагу(p);
 7 |   примениСкалирањеРешења(p, α);
 8 |   рачунајНаелектрисања(p);
 9 |   рачунајСиле(p);
10 |   помериЕМТачке(p);
11 end
12 испишиРешење();
```

Слика 3.1: Шема ЕМ методе за одабир атрибута

3.3.1 Репрезентација решења и иницијализација

У фази иницијализације се врши алокација меморије потребне за улазне податке, као и подела тих података на делове који се касније користе у унакрсној провери. Затим се генерише M-димензиони вектор ЕМ тачака означен са $\mathbf{p}$. Свака ЕМ тачка $\mathbf{p_i}$, $i = 1, ..., M$ представља N-димензиони вектор реалних вредности које узимају вредности из интервала $[0, 1]$, где N представља кардиналност скупа атрибута, тј. димензију разматраног проблема.

Генерисање иницијалних вредности ЕМ тачака се обавља тако што свака координата p_i^j, $j = 1, ..., N$ сваке ЕМ тачке $i = 1, ..., M$ узима вредност равномерне случајне промењиве из интервала [0,1]. p_i^j се касније, у добијању решења за проблем одабира атрибута, интерпретира на следећи начин: ако је вредност већа од 0.5, онда је j-ти атрибут укључен у решење представљено i-том тачком, а иначе није укључен. То имплицира да на самом почетку оптимизационог процеса, сваки атрибут има једнаке шансе да буде у решењу или ван њега.

3.3.2 Функција циља

Слика 3.2 приказује псеудокод процедуре за израчунавање функције циља.

Најпре се све ЕМ тачке $\mathbf{p_i}$, $i = 1, ..., M$ преводе у одговарајуће скупове атрибута, тј. бинарне векторе $\mathbf{s_i}$, $i = 1, ..., M$ на следећи начин:

```
улаз : pi
si = преведиУСкупОдабранихАтрибута(pi);
пронађено = претражиКеш(si);
if пронађено==NULL then
    if класификатор == 1NN then
        делова=5;
        p_i^obj = унакрснаПровераБалансираноТачност(делова, si);
    else if класификатор==SVM then
        делова=2;
        p_i^obj = унакрснаПровераОбичнаТачност(делова, si);
    end
    додајУКеш(si,p_i^obj);
else
    p_i^obj=пронађено.obj;
    поставиВиђеноУКешу(si);
end
```

Слика 3.2: Рачунање функције циља

$$s_i^k = \begin{cases} 1, & p_i^k \geq 0.5 \\ 0, & p_i^k < 0.5 \end{cases} \tag{3.6}$$

При том важи $k = 1, ..., N$. Вредност 1/0 означава да ли је атрибут укључен/искључен из скупа одабраних атрибута. Функција циља се рачуна на два начина у зависности од класификатора који се користи. У случају да је то 1-NN, користи се 5-компонентна унакрсна провера, а функција циља се добија по формули за балансирану тачност класификације (енг. balanced classification accuracy - BCA). BCA се користи са циљем умањења ефекта неизбалансираности података по питању класа.

$$BCA = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{n_{ii}}{\sum_{k=1}^{N_c} n_{ik}} \tag{3.7}$$

Вредношћу n_{ik} представљен је број података класе i класификованих класом k. У случају SVM класификатора, користи се 2-компонентна унакрсна провера, а функција циља се рачуна као тачност класификације: просечан удео коректно класификованих података у свим компонентама, што се, у случају да су величине компоненти једнаке, своди на рачунање укупног удела коректно

класификованих података у оквиру целог тренинг скупа.

Пре рачунања функције циља, вектор $\mathbf{s_i}$ се најпре тражи у кеш структури. Рачунање функције циља се извршава само у случају да вектор није пронађен. Детаљан опис поступка кеширања решења је дат у Секцији 3.3.5.

3.3.3 Локална претрага

Након рачунања функције циља, разматра се позивање локалне претраге LS. LS се не позива увек из два разлога: 1) локална претрага је обично временски захтеван процес, тако да је уштеда по питању броја LS позива значајна из аспекта ефикасности; 2) адекватно постављен критеријум за позивање LS може допринети побољшању експлоративних својстава претраге простора решења. Имајући на уму ова два разлога, предложени EM метод позива LS највише једном у свакој итерацији главног циклуса. Прецизније, да би се LS применио на некој EM тачки, потребно је да буде задовољена конјункција следећих услова:

1. EM тачка је најбоља или друга најбоља по вредности функције циља (први случај има предност);

2. LS није никад раније био примењен на посматраној EM тачки, или се вредност функције циља променила од последње примене LS;

3. Вредност најбоље EM тачке се није променила у најмање 10 последњих итерација.

Горе поменути предуслови за извршавање LS смањују трошкове претраге у ситуацијама у којим је мало вероватно да ће LS довести до побољшања решења. Истовремено се смањују и шансе да ће се претрага "заглавити" у локалном оптимуму. Треба приметити да примена LS на најбољу или другу најбољу јединку индиректно утиче и на друге EM тачке кроз касније рачунање наелектрисања и сила.

Ако су критеријуми позивања LS на посматраној тачки задовољени, LS се примењује на следећи начин. Постоје две процедуре локалне претраге: прва (LS1) је заснована на једној размени атрибута и примењује се одмах; а друга (LS2) користи двоструку замену атрибута и примењује се тек након што се нађе најбоља двострука замена. Поступак прве LS је следећи: у свакој

итерацији LS, врши се комплементирање једног бита вектора $\mathbf{s_i}$. То значи да се атрибут који није био у решењу укључује, или обрнуто, атрибут који је био у решењу сада се искључује из њега. Уколико се открије побољшање изазвано комплементирањем, оно се истог момента примењује, а процес побољшавања се наставља. LS1 се зауставља када се више не може постићи побољшање комплементирањем било ког од посматраних битова, односно атрибута.

Након што се заврши LS1, примењује се LS2 на истој EM тачки. Поступак је следећи: искључује се атрибут који припада решењу и тражи атрибут који није у решењу, али такав да се са његовим укључивањем добија побољшање. Када се провере сви парови искључених и укључених атрибута, примењује се пар који производи најбоље побољшање. Процес LS2 се зауставља када ниједан пар више не може да произведе побољшање.

3.3.4 Скалирање решења

У ситуацијама када координате EM тачака имају вредности блиске 0.5, појављују се непотребне флуктуације у претрази проузроковане честим прелажењем атрибута из стања укључен у стање неукључен и обрнуто. Под таквим околностима, конвергенција методе бива нарушена и целокупан систем претраге постаје непоузданији. Како би се ово избегло, стандардни EM алгоритам је проширен процедуром скалирања решења које омогућава бољу контролу померања EM тачака. Међутим, скалирање може довести и до смањења простора претраге и преране конвергенције. Зато је битно направити компромис у одабиру скалирајућег фактора и усагласити интензитет претраге, тј. брзину конвергенције са једне стране и њену свеобухватност са друге. Формула (3.8) приказује трансформацију скалирања која се примењује над појединачном EM тачком (слична оној која је коришћена у [Kra12] и [Kar12]).

$$\mathbf{p_i} = \alpha \mathbf{s_i} + (1 - \alpha)\mathbf{p_i} \tag{3.8}$$

Скалирајући фактор α узима вредности из интервала $[0, 1]$. Најбоља илустрација ефекта скалирајућег фактора се може направити посматрањем граничних вредности. Када је $\alpha = 1$, све координате тачке $\mathbf{p_i}$ се заокружују на ближе целобројне границе, односно на вредности 0 или 1: $(\mathbf{p_i} = \mathbf{s_i})$. Када је $\alpha = 0$, из формуле се види да неће бити никакве трансформације. Више

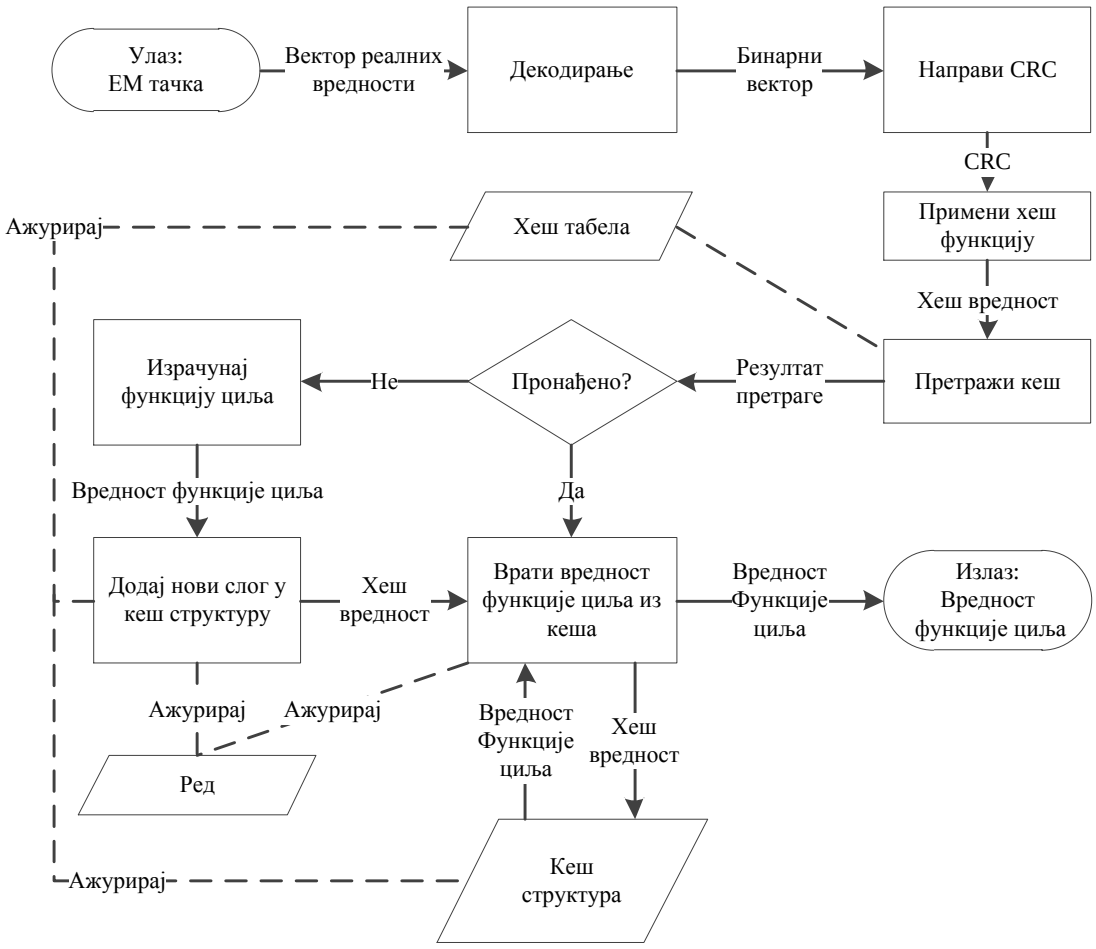
вредности α имају већу шансу да доведу претрагу до локалног оптимума, а мале вредности немају довољан ефекат на убрзање конвергенције. Емпиријски је утврђено да $\alpha = 0.1$ нуди добар компромис између брзине претраге и њене свеобухватности.

3.3.5 Кеширање решења

У предложеном ЕМ алгоритму је имплементирана ефикасна техника кеширања која значајно убрзава процес претраге решења. Кеширање се користи током рачунања вредности функције циља за ЕМ јединке. Основна идеја је да се решење које кодира ЕМ тачка (бинарни низ) најпре потражи у кеш структури. Уколико се пронађе, вредност функције циља се не рачуна, већ се преузима из кеш структуре. Само у случају да решење које кодира ЕМ тачка није пронађено, извршава се процедура за израчунавање функције циља, и потом се та вредност смешта у кеш структуру. Детаљан шематски приказ кеширања је приказан на Слици 3.3.

При кеширању се посматра бинарни низ који се добија дискретизацијом низа реалних вредности ЕМ тачке. Дискретизација се, као што је раније описано, извршава коришћењем "праг" вредности 0.5. Иако би кеширање било могуће и на реалним низовима ЕМ тачака, оно не би произвело задовољавајуће резултате, јер би број кеш "погодака" био занемарљив.

Зарад побољшања процеса кеширања, користе се још два механизма: CRC код (енг. cyclic redundancy check) и хеш функција. Када се генерише или мења ЕМ тачка, одмах се израчуна и њен одговарајући 32-битни CRC код. Ово омогућава да се ЕМ тачке са сличним бинарним низовима пресликају у "релативно" различите CRC кодове, тј. да се цела улазна популација ЕМ тачака равномерно распореди по 32-битном опсегу могућих вредности. На тај начин се гарантује и исто време извршавања помоћних процедура у процесу кеширања за различите димензије улазног проблема, јер је CRC код фиксне дужине. Над CRC кодом се потом примењује хеш функција и добија хеш вредност. Хеш вредност се потом тражи у хеш табели. Уколико се пронађе одговарајући елемент у хеш табели проверава се листа вредности на коју тај елемент показује. Листа вредности је заправо списак свих ЕМ тачака које производе добијену хеш вредност. Да би се утврдило која од тих јединки одговара траженој, пролази се



Слика 3.3: Кеширање решења

секвенцијално кроз ову листу и упоређује најпре CRC тражене јединке. Ако је он различит, прелази се одмах на следећи елемент из листе, иначе се проверава и бинарни низ како би се недвосмислено потврдило да је реч о траженој јединки. Адекватан одабир хеш функције омогућава да листе за дату хеш вредност у просеку буду релативно кратке. Поред тога, употреба CRC вредности, пре проверавања бинарних низова, додатно побољшава ефикасност претраге. Ако се јединка пронађе, очитава се вредност функције циља и враћа као резултат, а у супротном се за тражену јединку рачуна хеш вредност и функција циља и све то убацује у кеш структуру. И у овом случају се на крају враћа вредност функције циља као резултат.

Будући да кеш структура има ограничену меморију, било је потребно развити механизам ослобађања сувишних кеш слогова. Употребљена је LRU

(енг. least recently used) стратегија, која предвиђа избацивање најмање коришћеног скоријег слога из меморије. Имплементација је заснована на реду, јер одговара LRU приоритету избацивања и убацивања елемената.

3.4 Експериментални резултати

У овој секцији су представљени резултати тестирања предложене EM методе. Имплементација је написана у програмском језику C, и преведена Visual Studio 2010 преводиоцом. Сви тестови су извршени на рачунару са следећим карактеристикама: Intel i5 2430M @2.4GHz са 4GB RAM, под Windows 7 оперативним системом.

Извршена су два експериментална тестирања, на две засебне групе тест проблема који су преузети са UCI репозиторијума [BL13]. Прва група се састоји од 13 тест проблема, и на њој су резултати предложене EM методе упоређени са претходно предложеним EM методом (EM^{stc}) и генетским алгоритмом (GA) из литературе. Друга група која садржи 6 тест проблема је решена у циљу поређења са два метода заснована на оптимизацијиројевима (енг. particle swarm optimization - PSO), означена као PSO^1 и PSO^2.

У првом експерименту је поређење метода засновано на уделу изостављених атрибута, $RF = |\{s^k_{best}|s^k_{best} = 0,\ k = 1, ..., N\}|/N$, док се у другом користи мера која представља број задржаних атрибута. Суштински су ове две мере једнаке, једино што је у случају прве циљ извршити максимизацију, а у случају друге минимизацију вредности. Функције циља су у оба случаја рачунате техником унакрсне провере. Важно је напоменути да подела података на компоненте које се користе при унакрсној провери није идентична оној која се користи у методама са којима се пореди. Разлог је што оригинална подела није била доступна од стране аутора радова упоређиваних метода. Стога је упоредивост мера тачности класификације под знаком питања, посебно у случају тест проблема са малим бројем података, јер је у њима ефекат поделе скупа јаче изражен. Како би се решио овај проблем, имплементиран је егзактни алгоритам заснован на потпуној претрази скупа могућих подскупова атрибута (FS).

Експеримент 1 За сваки тест проблем, EM алгоритам је извршен 10 пута. За свако од тих 10 извршавања је коришћена другачија иницијална вредност

генератора случајних бројева. То је имало за последицу различиту поделу полазног скупа података на компоненте које су се касније користиле у унакрсној провери. За сваки тест проблем израчунате су просечне вредности BCA, RF, и укупног времена извршавања алгоритма.

Како би се извело поређење метода у "фер" околностима, предложени EM метод је извршен под истим условима као у [SL11]: број итерација алгоритма N_{it} је постављен на 600, док је број EM тачака $M = 150$. Скалирајући фактор α који контролише интензитет скалирања је постављен на 0.1. При рачунању функције циља коришћена је 5-компонентна унакрсна провера.

У [SL11] је направљено поређење неколико различитих метода у погледу тачности класификације, тачније BCA вредности. Показало се да не постоји значајна статистичка разлика између перформанси две *омотач* засноване методе, EM методе, и генетског алгоритма, и једне *филтер* методе, SVM заснованог на LIBSVM имплементацији [CL11]. Такође је показано да су све три методе значајно боље од неколико других метода за класификацију (погледати [SL11] за детаљан опис свих поређених метода). Предложена EM метода је упоређена са EM и GA методама, предложеним у [SL11], означеним са EM^{stc} и GA, респективно. С обзиром на другачију природу процеса класификације, предложена EM метода није упоређена са SVM методом.

Табела 3.2 приказује резултате поређења. Прве четири колоне представљају: име тест проблема, број атрибута, број класа и величину скупа података. Након тога следе просечне BCA вредности EM^{stc} и GA изражене у процентима. Следећа колона, FS приказује просечно оптимално решење достигнуто потпуном претрагом након 10 извршавања, а наредна колона је просечна BCA вредност предложене EM методе. У случају да се та вредност поклапа са оптималним просечним решењем, у колони је записана вредност *opt*. То значи да је у тим случајевима EM достигао оптимално решење у свих 10 извршавања. Последње три колоне приказују проценат изостављених атрибута.

Резултати из Табеле 3.2 показују да је EM бољи од EM^{stc} и GA у 10 од 13 случајева према вредности RF. У преостала 3 случаја, предложени EM је два пута други најбољи, и само једном гори од EM^{stc} и GA. Такође се показује да EM достиже просечна оптимална решења у решавању 10 од 13 тест проблема. Само у случају тест проблема *waveform* проценат успешности достизања

оптималног решења је био нижи (50%), што је произвело и нешто нижи просечни BCA. У преостала два проблема, *spambase* и *water*, FS алгоритам није успео да заврши извршавање за мање од 3 дана, па је стога био прекинут. Може се приметити да је у та два случаја, EM бољи по питању BCA од друга два *омотач* метода.

Табела 3.3 приказује времена извршавања изражена у секундама за све методе. Тест проблеми су уређени растуће према дужини времена извршавања алгоритма потпуне претраге. Може се приметити да су првих 5 тест проблема лако решени од стране FS, сваки за мање од 10 секунди. Предложени EM метод се понаша подједнако брзо на овим проблемима. Ово је последица кеширања, јер је број могућих комбинација одабраних атрибута релативно мали, па је и проценат кеширања (EM_{cache}) изузетно висок. Проценат кеширања се добија као однос броја успешних и укупног броја претрага кеш структуре. Успешна претрага значи да је тражени елемент пронађен у кеш структури, па израчунавање функције циља није било непоходно. Може се видети да EM^{stc} и GA захтевају много више времена при решавању ових тест проблема. То је посебно изражено у тест проблему *abalone* где је EM^{stc} 200 пута спорији од EM. EM је бржи од осталих алгоритама на свим преосталим тест проблемима, изузев на проблему *spambase*.

Табела 3.2: Поређење метода према проценту изостављених атрибута

проблем	N	N_c	N_r	EM^{stc}	GA	FS	EM	EM^{stc}_{RF}	GA_{RF}	EM_{RF}
abalone	8	11	3842	24.35	24.37	23.99	*opt*	52.50	50.00	<u>57.50</u>
glass	9	6	214	79.51	79.72	78.69	*opt*	<u>51.11</u>	42.22	43.33
iris	4	3	150	98.00	98.00	99.39	*opt*	55.00	<u>60.00</u>	50.00
letter	16	26	20000	96.39	95.24	96.39	*opt*	27.50	21.25	<u>28.75</u>
shuttle	9	7	58000	91.65	91.56	94.89	*opt*	46.67	46.67	<u>47.78</u>
spambase	57	2	4601	94.30	91.55	-	94.35	42.81	36.00	<u>51.93</u>
tae	5	3	151	65.47	65.47	62.26	*opt*	40.00	36.00	<u>44.00</u>
vehicle	18	4	846	78.10	74.64	79.50	*opt*	52.22	45.56	<u>52.78</u>
water	38	4	513	73.34	66.28	-	80.03	54.21	47.89	<u>63.16</u>
waveform	21	3	5000	80.84	77.36	80.76	80.45	21.90	33.33	<u>34.76</u>
wine	13	3	178	98.57	98.57	99.80	*opt*	58.46	61.54	<u>72.31</u>
wisconsin	9	2	683	98.25	98.04	98.62	*opt*	<u>53.33</u>	40.00	48.89
yeast	8	9	1479	47.07	47.03	51.15	*opt*	17.50	12.50	<u>22.50</u>

Табела 3.3: Времена извршавања и проценат кеширања

проблем	N	N_r	FS_t	EM_t^{stc}	GA_t	EM_t	EM_{cache}
tae	5	151	0.0	3.9	261.8	0.9	99.96
iris	4	150	0.0	7.5	288.6	0.9	99.98
glass	9	214	0.1	7.5	410.8	1.7	99.55
wisconsin	9	683	0.6	70.6	2096.8	2.0	99.52
yeast	8	1479	1.2	234.7	2252.3	2.4	99.74
wine	13	178	1.9	153.0	269.9	2.5	97.48
abalone	8	3842	8.4	1376.1	7097.2	7.2	99.74
vehicle	18	846	627.8	355.3	2948.3	21.5	91.21
shuttle	9	58000	6595.6	10287.3	362608.3	3876.7	99.56
letter	16	20000	79921.7	21953.0	67707.9	2189.7	97.62
waveform	21	5000	174682.1	8189.4	9646.2	1061.3	88.78
water	38	513	>3 дана	262.8	1574.5	55.7	70.80
spambase	57	4601	>3 дана	5530.8	7396.2	14237.3	52.95

Експеримент 2 У другом експерименту, предложени EM метод је упоређен са две варијанте PSO метода, предложене у [UMC11]. Поређење је засновано на 6 тест проблема из UCI репозиторијума [BL13]. Експериментално окружење је исто као у [UMC11]. LIBSVM имплементација SVM је коришћена као класификациони алгоритам [CL11]. Подешавања SVM су следећа: параметар цене грешке је постављен на 100, док се радијална функција са $\sigma = 2$ користи као кернел. За вишекласне проблеме класификације, користи се *Један-против-Осталих* стратегија. Максималан број итерација EM алгоритма је постављен на 300, док је 2-компонентна унакрсна провера употребљена за израчунавање функције циља. Подаци за поређење су тачност класификације и број задржаних атрибута. С обзиром да се и у овом експерименту EM извршава 10 пута са различитим иницијалним стањем генератора случајних бројева, подаци за поређење представљају просечне вредности добијене из тих 10 извршавања за сваки тест проблем понаособ.

Будући да времена извршавања нису дата у [UMC11], поређење по питању ефикасности извршавања није направљено. Табела 3.4 приказује следеће информације: име тест проблема, број атрибута, број класа, тачност класификације за две варијанте PSO методе, просечну оптималну тачност класификације добијену потпуном претрагом (уколико се извршавање завршава у разумном времену), тачност класификације EM методе и просечан број

задржаних атрибута за сва три алгоритма: PSO_d^1, PSO_d^2 and EM_d. Иако није направљено поређење времена извршавања, зарад комплетности информација, последње три колоне приказују времена извршавања алгоритма потпуне претраге, EM алгоритма, и проценат кеширања EM алгоритма.

Табела 3.4: Поређење метода према просечном броју задржаних атрибута

проблем	N	C	PSO^1	PSO^2	FS	EM	PSO_d^1	PSO_d^2	EM_d	FS_t	EM_t	EM_{cache}
glass	10	7	78.50	80.28	70.00	*opt*	4.9	5.9	5.1	29.6	19.9	95.19
wine	13	3	99.19	99.72	97.30	*opt*	8.3	8.6	6.7	354.1	42.2	85.09
breast-cancer	30	2	96.83	97.66	-	95.92	11.1	12.2	6.4	>3 дана	1412.0	51.07
ionosphere	34	2	95.44	95.54	-	96.72	9.6	9.25	12.5	>3 дана	548.2	32.97
sonar	60	2	84.28	85.67	-	91.68	14.3	13.9	21.8	>3 дана	1868.1	14.85
heart	13	2	84.30	86.01	83.74	*opt*	8.6	7.5	3.5	551.4	66.8	86.35

EM је достигао сва оптимална решења на проблемима које је FS успео да реши у разумном времену (за мање од 3 дана). Може се видети да је тачност класификације EM методе виша од тачности PSO алгоритама на 2 од 3 тест проблема. Број задржаних атрибута је комплементарна мера оној која је коришћена у првом експерименту. Из тог разлога, овде је пожељно да тај број буде што мањи. Може се видети да је EM достигао најмањи просечан број задржаних атрибута у 3 од 6 посматраних тест проблема.

3.5 Завршна разматрања

У овом поглављу је представљен побољшани EM метод за решавање проблема одабира атрибута. Метод је унапређен у неколико аспеката. Најпре, примењена је процедура скалирања EM решења која је омогућила бољу контролу над процесом претраге решења. Уведена је ефикасна локална претрага која се примењује "пажљиво" ради што веће уштеде рачунарских ресурса. Локална претрага се састоји из две фазе, прве која је рачунски јефтинија и заснована је на једној замени атрибута са тренутном применом у случају побољшања. Друга је нешто скупља, јер захтева двоструку замену атрибута, а примењује се тек након што се нађе најбоља таква замена. Овако пажљиво осмишљена локална претрага омогућава ефикасно локално побољшавање EM решења уз минималну потрошњу рачунарских ресурса. Последње побољшање које је

допринело драстичном убрзању алгоритма је кеширање решења. Показало се да је проценат кеширања у неким тест проблемима довео до убрзања од приближно 200 пута.

Експериментални резултати над UCI тест проблемима су показали предност предложене EM методе у односу на друге *омотач* засноване приступе када је у питању тачност класификације и степен смањења потребних атрибута.

Правци даљег развоја, по питању решавања проблема одабира атрибута и EM методе, могу бити хибридизација предложеног метода са *филтер* методама за класификацију. Висок научни и практични потенцијал би имала и паралелизација методе јер би то довело до додатног убрзања алгоритма. Развијање паралелног модела израчунавања за проблем одабира атрибута би имало велики значај код ширег скупа *омотач* заснованих метода, јер би то решило проблем њихове наслеђене временске неефикасности.

Поглавље 4

Примена EM у подешавању тежина атрибута

4.1 Проблем одређивања тежина атрибута

Одређивање тежина атрибута (енг. feature weighting) подразумева додељивање реалних тежинских фактора атрибутима у циљу побољшања функције циља. Функција циља у контексту класификационих примена може бити тачност класификације, одзив и др. У овом поглављу ће се разматрати два домена примене одређивања тежина атрибута. Први је класификација, а други је закључивање на основу претходних случајева (енг. case-based reasoning - CBR). Јасно је да ће се и функције циља у те две примене разликовати, па ће сходно томе, проблем закључивања на основу претходних случајева бити детаљно дефинисан у наставку поглавља.

Одређивање тежина атрибута представља генерализацију проблема одабира атрибута јер, уколико се изврши рестрикција могућих тежинских фактора на скуп $\{0, 1\}$, проблем се своди на одабир атрибута. Слично као и код проблема одабира атрибута, методологија решавања може бити заснована на: 1) примени информација о скупу података у подешавању тежина (једнопролазни или константан број пролаза кроз скуп података), 2) примени повратних информација о вредности функције циља (вишепролазни). Јасно је да постоји аналогија између методологије 1) и *филтер* метода код одабира атрибута. Са друге стране, методологија 2) је слична тзв. *омотач* техникама код одабира атрибута. У овом поглављу се проблем анализира применом методе

најближих суседа. Ово не нарушава општост предложене методе, јер је могуће, уз мале модификације, сценарио одабира тежина пренети на неке друге класификационе методе, попут методе подржавајућих вектора, линеарне дискримантне функције, неуронске мреже и др.

У Секцији 1.2.1 је изложен основни k-NN метод. Формулом $dist : \mathbf{R}^N \times \mathbf{R}^N \rightarrow \mathbf{R}$ је представљена функција удаљености која се рачуна при одабиру најближих суседа. У Секцији 1.2.1, тачна форма ове функције није дата, јер постоји већи број функција које се могу користити као мере удаљености. Неке од познатијих функција удаљености су:

$$euklidska\_udaljenost(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \tag{4.1}$$

$$minkovski\_udaljenost(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{N} (x_i - y_i)^r \right)^{1/r} \tag{4.2}$$

$$mahalanobis\_udaljenost(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})\Sigma^{-1}(\mathbf{x} - \mathbf{y})^T \tag{4.3}$$

Где су ознакама $\mathbf{x}$ и $\mathbf{y}$ представљени вектори атрибута за које се удаљеност рачуна, r у (4.2) представља параметар чије вредности могу бити $1, 2, ..., \infty$, а Σ је матрица коваријанси. У (4.1) је приказано Еуклидско растојање. Оно се користи у овом поглављу, али у модификованој форми. Друга мера удаљености (4.2) је уопштење Еуклидског растојања, тј. Еуклидско је специјални случај ове удаљености за вредност параметра $r = 2$. Последње је такође уопштење Еуклидског и посебно је корисно у применама када су атрибути корелисани, имају различите опсеге вредности и/или када је расподела атрибута блиска нормалној расподели. Ниједна од приказаних функција удаљености не узима у обзир значај атрибута у рачунању удаљености. Уопштена варијанта Еуклидске удаљености, тзв. тежинска Еуклидска удаљеност, у којој је то могуће, је дата формулом:

$$teuklidska\_udaljenost(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{N} h_i (x_i - y_i)^2} \tag{4.4}$$

Вредности h_i, $i = 1, ..., N$ су тежински фактори. С обзиром да се разматра

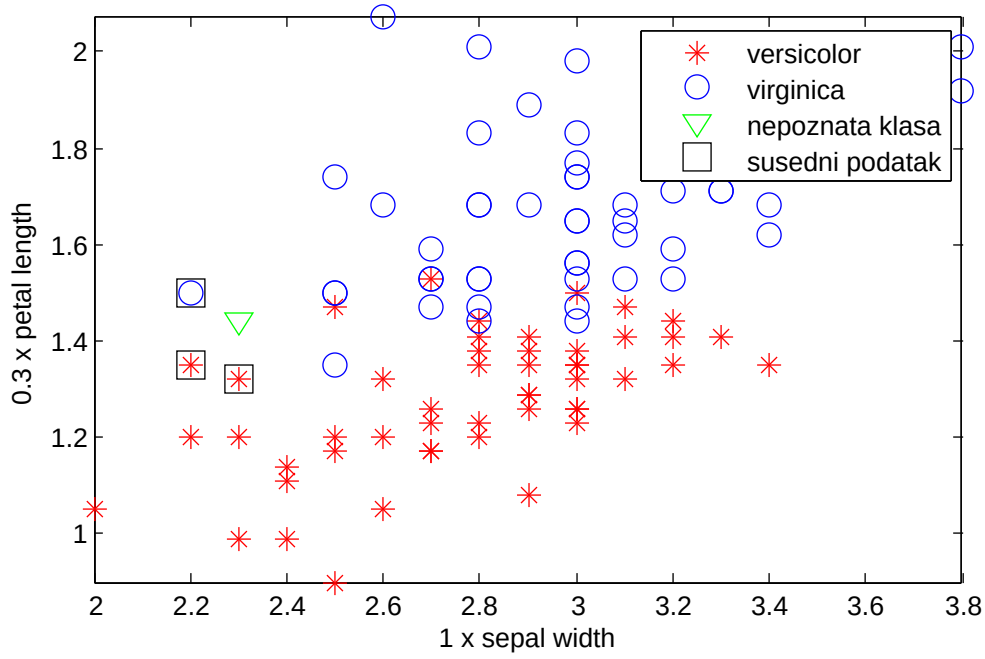
проблем додељивања тежина атрибута, а не додељивања тежина изразима $(x_i - y_i)^2,\ i = 1, ..., N$, практичнија је следећа форма:

$$teuklidska\_udaljenost(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{N} (w_i x_i - w_i y_i)^2} = \sqrt{\sum_{i=1}^{N} w_i^2 (x_i - y_i)^2} \quad (4.5)$$

Вредност w_i је тежина додељена i-том атрибуту, а из (4.5) је јасна веза са тежинским фактором h_i: $h_i = w_i^2$. У применама се неретко постављају и додатни услови на опсег вредности тежина, попут услова $0 \leq h_i \leq 1$ и $\sum_{i=1}^{N} h_i = 1$. Додатни услови, наравно, зависе од својства класификационог модела у којем ће мера удаљености бити употребљена, тј. одговора на питање: да ли су само релативни односи између тежина атрибута битни?

Илуструјмо утицај тежинских фактора на функцију одређивања класе у оквиру методе најближих суседа. На Слици 1.5 из уводног поглавља је био приказан случај класификације вектора атрибута овом методом. Као што се може видети на тој слици, податак је класификован као *iris virginica*, јер су два од три најближа суседа припадала тој класи. Ако се примене тежине $w_1 = 1$ и $w_2 = 0.3$ респективно на атрибуте *sepal width* и *petal length*, полазни распоред података ће се значајно променити. На Слици 4.1 је приказано ново стање након додељивања тежина. Може се видети да се променила и функција одлучивања, јер су након примене тежина, два од три најближа суседа цветови типа *iris versicolor*.

Поред употребе методе најближих суседа у решавању класификационих проблема, размотрена је њена примена и у закључивању на основу претходних случајева, општој методологији за решавање проблема која користи претходне случајеве (проблеме) када решава нове. У даљем тексту је закључивање на основу претходних случајева означено скраћено са CBR. Разрешавање новог случаја се у CBR састоји из две фазе: 1) проналажењу сличних претходно разрешених случајева и 2) прилагођавању (адаптацији) скупа тих случајева условима решавања новог случаја. CBR представља индуктивну методологију, и комплементарна је системима заснованим на правилима (енг. rule-based system), где се закључивање врши дедукцијом, односно примени општих правила у специфичним условима. Већина CBR система се састоји из неколико



Слика 4.1: 3NN на Ирис скупу података након примене тежина

подсистема који се баве потпроблемима CBR: прикупљању случајева, њиховој репрезентацији, индексирању, претраживању и прилагођавању. За детаљан преглед CBR методологија, читалац се упућује на [AP94] и [WM94]. CBR је применљив у различитим научним дисциплинама, индустрији, пољопривреди, медицини, итд. Од времена првих CBR система, десила су се многобројна унапређења, као и комбиновања CBR система са техникама машинског учења и истраживања података. У овом раду је представљено унапређење CBR система које се тиче прве фазе у решавању новог случаја, а то је проналажење сличних случајева. Ова фаза у многоме зависи од мере сличности (удаљености) која се користи при утврђивању сличности два случаја. Оптималну меру сличности је тешко формулисати коришћењем доменског знања за неку област. Разлог томе је велика динамичност података у погледу међусобних релација, а са друге стране, људи нису у стању да открију све такве релације. Зато се у дефинисању мера сличности између случајева CBR система све чешће примењују аутоматизоване процедуре вођене подацима (енг. data driven). Код одређивања функције сличности могуће је бирати њену функционалну форму и/или њену параметарску структуру. Већина радова на ову тему подразумева фиксирану функционалну форму, а затим врши подешавање параметара који

омогућавају највиши степен квалитета одабира случајева. Детаљан преглед мера сличности које се користе у CBR је направљен у [Cun09]. У овом поглављу се за одабир сличних случајева користи претходно поменута метода најближих суседа. Раније описана мера удаљености, тежинска Еуклидска удаљеност ће бити функционална форма, док ће се предложени метод за подешавање тежина атрибута користити за њено подешавање.

4.2 Претходни резултати

Примене у класификацији. Подешавање тежина атрибута је у стању да унапреди квалитет класификације. У литератури су предложене различите методе за ту намену.

У [WAM97], аутори праве преглед методологија за подешавање тежина атрибута у оквиру техника заснованих на лењом учењу (енг. lazy learning algorithms), којима, између осталих, припада и метода најближих суседа. Један од главних закључака овог прегледа је да методе које користе повратну информацију о квалитету функције циља (нпр. тачности класификације у случају да се посматра класификациони проблем), имају 3 предности у односу на друге методе: 1) захтевају мање препроцесирања, 2) боље се понашају када су присутни зависни (корелисани) атрибути, и 3) захтевају мање тренинг података за учење. Такође је показано да подешавање тежина атрибута може квалитативно да надмаши одабир атрибута када постоје атрибути који су корисни, али и мање значајни од других.

У скоријим студијама, проблем одабира атрибута и подешавања њихових тежина је често био разматран истовремено, нпр. то је случај у [PCN07]. Аутори користе класификациони модел заснован на линеарној дискриминантној функцији. Експериментално је утврђено да је тачност класификације предложене методе статистички значајно боља него код класичних дискриминантних линеарних класификатора (класична дискриминантна анализа и логистичка регресија). Сличан сценарио, у ком се истовремено разматрају оба проблема, предложен је у [TBK07]. Аутори користе k-NN методу за класификацију, док је за решавање проблема подешавања тежина и одабира атрибута употребљена табу претрага. Резултати тестирања су показали значајно унапређење у погледу тачности класификације у односу

на друге методе из литературе. Три хибридна модела за подешавање тежина и класификацију су предложена у [CLL12]. Аутори овог рада су користили комбиноване приступе засноване на оптимизацијиројевима, генетском алгоритму и кластер методи k-средина. Квалитет предложених метода је потврђен кроз тестирања над бинарним класификационим тест проблемима који имају примене у медицинској дијагностици. У [Alf+11] се испитује примена хибридног генетског алгоритма за подешавање тежина и одабир атрибута (под скраћеницом GEFeWS) за више-биометријско препознавање (енг. multi-biometric recognition). Резултати су показали да GEFeWS има већу моћ препознавања од GEFeS методе, која користи генетски алгоритам само за одабир атрибута. Поред тога, предложени хибрид користи значајно мањи број атрибута од GEFeW методе која користи генетски алгоритам само за подешавање тежина атрибута.

Аутори студије, описане у [Pol12], су предложили методу која користи расплинути метод c-средина (енг. fuzzy c-means - FCM) за подешавање тежина атрибута (FCMFW). Метода се користи за детекцију Паркинсонове болести. Тестирање је извршено над јавно доступним скупом података о Паркинсоновој болести ([BL13]). За класификацију је коришћен k-NN метод. У [Lug11] се примењује инкрементална техника ажурирања тежина атрибута у оквиру расплинутог класификатора (енг. fuzzy classifier). Када се достигне тежина блиска нули, врши се елиминација атрибута и тиме се смањује димензија улазног проблема. Резултати су показали да ова техника у стању да смањи димензију улазног скупа података, а да при том задржи или додатно побољша моћ класификације.

У [MFM11] се предлаже директан приступ одређивању тежина (без повратних информација о функцији циља). Као хеуристику, аутори користе информациону добит (енг. information gain) у утврђивању тежина атрибута. Ефикасност методе је велика, с обзиром да је потребан само један пролаз кроз скуп података. Тестирање спроведено над 128 бинарних и више-класних класификационих проблема показало је да је метода компетитивна, а неретко и боља од неколико успешних метода из литературе, укључујући и методу подржавајућих вектора. Посебно је јасан допринос у примени методе на великим скуповима података. Аутори наводе да је на скупу података који се

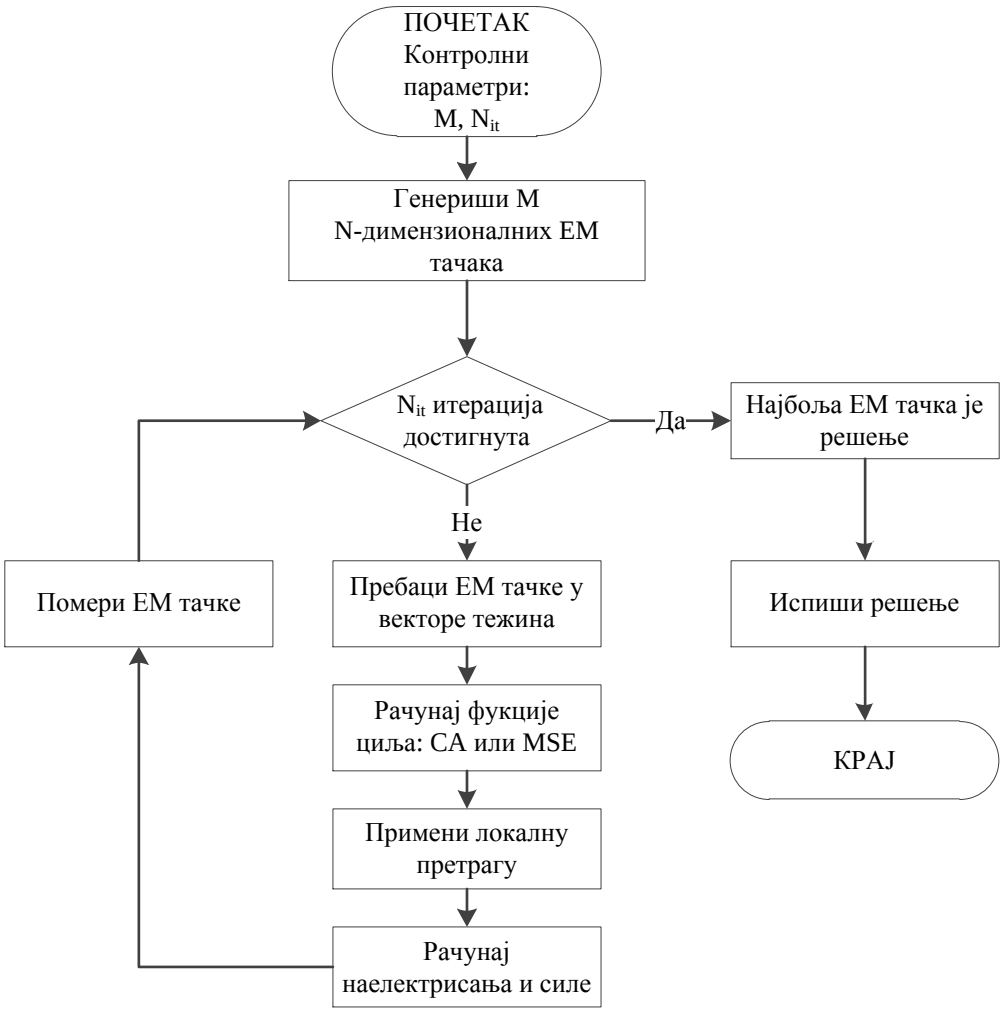
састојао из 12294 класе и 135973 тренинг података, предложени метод изградио класификациони модел за само 13 секунди, и при том произвео упоредив квалитет класификације као и метода подржавајућих вектора, којој је требало 15 минута да изгради модел.

Примене у закључивању на основу претходних случајева. У раду [BP06], CBR се користи за моделовање распореда смена медицинског особља (енг. nurse roostering). Проблем разматра процес генерисања распореда смена за одређени временски период, а чини га тешким то што је потребно усагласити велики број болничких правила са преференцијама медицинског особља. CBR је унапређен употребом генетског алгоритма у циљу одабира и подешавања тежина атрибута из улазног скупа података. Аутори су показали да предложена техника смањује број потребних атрибута и повећава тачност CBR система. Слично, у [AK09], аутори предлажу побољшани CBR модел за предвиђање банкрота. Побољшање је добијено истовременом оптимизацијом тежина атрибута, и одабира података (хоризонтална рестрикција). За ту намену је коришћен генетски алгоритам. Експериментални резултати су показали повећану тачност класификације у поређењу са конвенционалним CBR системима. У [IP07] је предложен CBR систем за прилагођавање кориснику (енг. personalization system), у којем је за подешавање тежина атрибута употребљена вештачка неуронска мрежа. Разијена је и методологија за рад са симболичким атрибутима, заснована на мери разлике вредности (енг. value difference metric). У раду [LC11], аутори предлажу CBR комбинован са алгоритмом вештачког имуног система. Показано је да употреба ове хибридне технике за подешавање параметара, одабир и подешавање тежина атрибута, повећава тачност класификације CBR система.

Даљи ток излагања је организован на следећи начин: у наредној секцији је описан EM метод за подешавање тежина атрибута; потом следи преглед експерименталних резултата и поређење са другим методама из литературе; последња секција је резервисана за завршна разматрања и правце даљег развоја када је у питању посматрани проблем.

4.3 Предложени ЕМ метод

Слика 4.2 приказује општу схему предложене ЕМ методе за подешавање тежина k-NN класификатора (ЕМ k-NN скраћено).



Слика 4.2: Шема предложеног ЕМ алгоритма за подешавање тежина

Шема предложеног ЕМ метода је дата на Слици 3.1. Алгоритам захтева 2 контролна параметра из основног ЕМ алгоритма: N_{it} - број итерација, и M - величину популације. Први корак је генерисање иницијалних ЕМ тачака. Након тога, алгоритам улази у главни циклус у којем се врше сви битни аспекти оптимизације. Најпре се врши рачунање функција циља за све ЕМ тачке. Након тога се врши локална претрага, док се на крају рачунају налектрисања и силе које дејствују на тачке, и у складу са тим врши померање ЕМ тачака. Главни

циклус се завршава када се достигне критеријум завршетка, у овом случају је то максимални број итерација. У наредним секцијама је сваки од аспеката ЕМ алгоритма за подешавање тежина k-NN бити детаљно описан.

4.3.1 Репрезентација решења и иницијализација

Свака ЕМ тачка $\mathbf{p_i}$, $i = 1, ..., M$ је N-димензиони вектор реалних координата које узимају вредности из интервала $[0, 1]$, где N одговара броју атрибута у скупу података. Генерисање иницијалних вредности ЕМ тачака се обавља тако што свака координата p_i^j, $j = 1, ..., N$ сваке ЕМ тачке $i = 1, ..., M$ узима вредност равномерне случајне променљиве из интервала [0,1]. Интерпретрација решења, на основу ЕМ тачака, је следећа: за посматрану ЕМ тачку $\mathbf{p_i}$, решење проблема представљено том тачком $\mathbf{w}$ се добија доделом: $\mathbf{w} = \mathbf{p_i}$, односно $(w_1, ..., w_N) = (p_i^1, ..., p_i^N)$. На тај начин ће тежине атрибута увек имати неку од вредности из опсега $[0, 1]$. У уводном делу овог поглавља је речено да се у неким применама могу постављати додатна ограничења на вредности тежина. Овде то није случај, тако да је допустиви скуп за тежине $\mathbf{w} \in [0, 1]^N$.

4.3.2 Функција циља

Функција циља представља меру квалитета решења, односно скупа тежина које ЕМ тачка представља. Одабир адекватне функције циља има критичан значај за процес претраге простора могућих решења. Као што је наглашено у уводном поглављу, када су у питању проблеми класификације, циљ је направити такву функцију циља која је у стању да понуди добру оцену квалитета класификације над тест скупом података. Тест скуп података се добија издвајањем подскупа података из оригиналног скупа, док се остатак података користи за тренирање класификационог модела. Постоје многе варијације у погледу величина тренинг и тест скупова, оцена квалитета и других аспеката који могу утицати на коначни квалитет класификационог модела. Преглед оцена квалитета направљених у Секцији 1.3.2 је довољан за разумевање функција циља које су коришћене за проблем подешавања тежина атрибута.

Функција циља у Експерименту 1. У првом експерименту се разматра проблем класификације над медицинско-дијагностичком базом података. За ту намену је коришћена 10 x 4 унакрсна провера. Ово значи да је 4-

унакрсна провера примењена 10 пута за различите поделе тренинг скупа на 4 компоненте. На крају се функција циља рачуна као просек добијених 10 вредности. Код k-NN методе бира се скуп најближих суседа за посматрани вектор атрибута. Коначна одлука о класи која ће бити додељена новом вектору атрибута се заснива на гласачкој шеми (енг. voting scheme). Она подразумева да се за класу бира она која се највише појављује у скупу најближих суседа. Будући да се разматрају само бинарни класификациони проблеми, пожељно је вршити k-NN метод за непарну вредност k, јер се на тај начин класа увек одређује једнозначно. Уместо тачности класификације приказане су грешке класификације, што је комплементарна мера. Грешка класификације појединачног тренинг податка $\mathbf{x}$ узима вредност 0 ако је класа коју је предложио k-NN, означена са $\hat{y} = c(\mathbf{x})$ (c је функција одређивања класе за методу најближих суседа дата Формулом 1.9), једнака стварној класи тренинг податка, означеној са y, иначе је грешка једнака 1. Грешка класификације за цео тренинг скуп се добија дељењем збира свих појединачних грешака са укупним бројем тренинг података. Након што се овај поступак изврши 10 пута, као што је већ речено, узима се просечна вредност добијених 10 вредности, и поставља за коначну вредност функције циља.

Функција циља у Експерименту 2. У другом експерименту, у којем се разматра одабир случајева у оквиру CBR система, функција циља је заснована на LOO. Проблем који се овде разматра има другачију структуру од класификационог проблема. Разлика је у димензији излазне променљиве. Код класификације је то једна вредност, класа која је придружена вектору атрибута. Код CBR је излазна променљива вектор вредности димензије R (Слика 4.3). Основна идеја у тако дефинисаном проблему одабира случајева је дати "добру" процену вредности излазних вредности. Слично као у случају класификације, ово се своди на формирање такве оцене, која је у стању да на тест скупу података произведе висок квалитет. Нека је са $\hat{\mathbf{y}} = (\hat{y}_1, ..., \hat{y}_R) = c_{CBR}(\mathbf{x})$ означен предвиђени вектор који производи k-NN модел. Његова вредност се добија упросечавањем излазних вредности најближих суседа по свакој од координата:

$$c_{CBR}(\mathbf{x}) = \frac{\sum_{j=1}^{k} \mathbf{y}^{(\mathbf{i_j})}}{k} \tag{4.6}$$

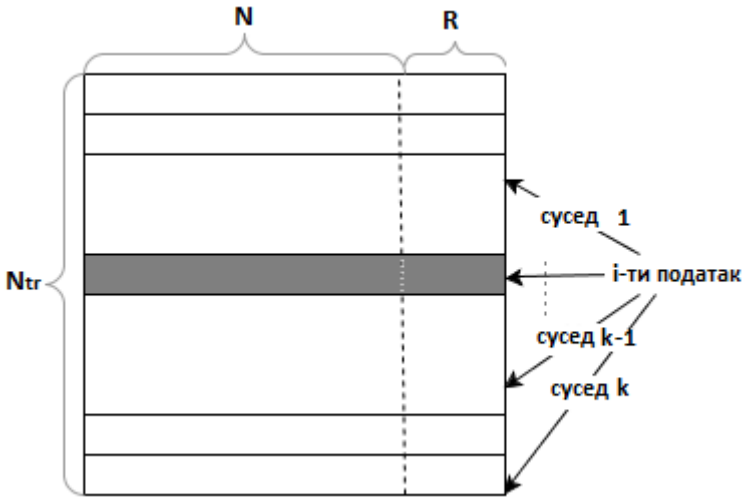
Вредностима $\{\mathbf{y}^{(\mathbf{i_1})}, ..., \mathbf{y}^{(\mathbf{i_k})}\}$ је представљен скуп излазних вредности k

најближих суседа. Грешка у рачунању излазних вредности једног тренинг податка $\mathbf{x}^{(i)}$ се рачуна као средња квадратна грешка вектора предвиђених излазних вредности у односу на праве (реализоване) вредности излазних променљивих означених са $\mathbf{y}^{(i)} = (y_1^{(i)}, ..., y_R^{(i)})$:

$$MSE_i = \frac{\sum_{j=1}^{R} (\hat{y}_j^{(i)} - y_j^{(i)})^2}{R}. \tag{4.7}$$

Укупна функција циља се потом рачуна као просечна грешка на целом тренинг скупу:

$$MSE = \frac{\sum_{i=1}^{N_{tr}} MSE_i}{N_{tr}}. \tag{4.8}$$



Слика 4.3: LOO функција циља за k-NN

4.3.3 Локална претрага

Локална претрага (LS) се извршава једном по итерацији главног циклуса. Као што је то био случај код ЕМ методе за подешавање параметара, LS је подељена у две фазе. У првој фази се бира највише једна ЕМ тачка над којом ће бити примењена LS, док се у другој врши сама примена. На Слици 4.4 је приказан псеудокод локалне претраге.

Фаза одабира је слична оној описаној у [FKM13] као и локалној претрази описаној у методи за подешавање параметара SVM. LS се може применити над једном од две најбоље јединке. Популација решења се прво уреди у растућем

```
   улаз: p, D_tr
 1 сортирајПоГрешциКласификацијеРастуће(p);
 2 ind = -1;
 3 if применљиваЛокалнаПретрага(p_1) then
 4 |  ind = 1;
 5 else if применљиваЛокалнаПретрага(p_2) then
 6 |  ind = 2;
 7 end
 8 if ind== -1 then
 9 |  return;
10 end
11 for k ← 1 to N do
12 |  for знак ← 0 to 1 do
13 |  |  корак = (знак-p_ind^k)/20;
14 |  |  побољшање = true;
15 |  |  while побољшање = true do
16 |  |  |  побољшање=false;
17 |  |  |  стараФункцијаЦиља=p_ind^obj;
18 |  |  |  стараКоордината=p_ind^k;
19 |  |  |  p_ind^k = p_ind^k + корак;
20 |  |  |  новаФункцијаЦиља=функцијаЦиља(p_ind, D_tr);
21 |  |  |  if новаФункцијаЦиља < стараФункцијаЦиља then
22 |  |  |  |  побољшање = true;
23 |  |  |  else
24 |  |  |  |  p_ind^k = стараКоордината;
25 |  |  |  |  p_ind^obj = стараФункцијаЦиља;
26 |  |  |  end
27 |  |  end
28 |  end
29 end
```

Слика 4.4: Локална претрага

поретку према вредности функције циља. Пошто је функција циља грешка класификације у првом експерименту, а MSE у другом, вредности које се у тако формираном уређењу налазе на почетку, имају бољу функцију циља. Након тога се анализирају функције циља прве две тачке у уређењу. Прво се провери најбоља ЕМ тачка помоћу процедуре *применљиваЛокалнаПретрага*. Ако процедура врати логичко да, локална претрага се примењује над том тачком. У супротном се на исти начин проверава друга најбоља тачка. Ако ни за њу није испуњен услов применљивости, LS се не врши уопште у тој итерацији

главног циклуса. Да би локална претрага била применљива, потребно је да су испуњени следећи услови: 1) на посматрану тачку никад раније није био примењен поступак LS, или је LS био примењен раније, али се вредност функције циља у међувремену променила, 2) од последње промене најбољег решења у популацији је прошло мање од 10 итерација циклуса. Први услов спречава беспотребну примену LS над тачкама над којима је LS био раније примењен, а након тога није дошло до промене тачке. Без ове рестрикције, LS би се примењивао и на тачкама на којима LS гарантовано не би дала никаква побољшања. Други услов изазива спречавање LS уколико је последња промена најбољег решења била релативно скора. На тај начин се даје времена популацији решења да се "боље" распореди по простору претраге и успорава конвергенција ка локалним оптимумима.

Након одабира тачке за LS, врши се њена примена. Проблем подешавања тежина атрибута има сличну репрезентацију решења као проблем подешавања параметара SVM. Из тог разлога је и механизам локалне претраге врло сличан. Врши се систематско мењање вредности координата EM тачке најпре према улево, односно минималној вредности 0. Ако то не произведе побољшање, мења се смер, и креће се ка десној граници интервала [0,1]. Величина корака, који се прави, је једнака двадесетини преосталог интервала до леве односно десне границе. Померање се врши док год то производи побољшање. Након што се на овај начин обраде све координате EM тачке, LS се зауставља.

4.4 Експериментални резултати

У овој секцији су изложени експериментални резултати и упоредна анализа предложене методе са другим методама из литературе. EM k-NN је написан у програмском језику C и преведен Visual Studio 2010 преводиоцем. Сва тестирања су извршена на рачунару Intel i5 2430M @2.4GHz са 4GB RAM под Windows 7 оперативним системом.

Експеримент 1. За потребе првог експеримента употребљена су два тест проблема из домена медицинске дијагностике: први са подацима о болести јетре (енг. liver disorder), а други о туморима дојке (енг. breast cancer). Први скуп података се састоји до 345 података, од којих сваки има 7 информација. Првих 5 су атрибути који описују резултате различитих крвних анализа за које се верује

да су осетљиви на претерану употребу алкохола. Шести податак представља број дневно попијених пића, где је свако урачунато као еквивалент пића од пола пинте (1 пинта ≈ 0.47 литре). На основу овога је израчунато седмо својство које представља информацију да ли је број пића већи од 5, што уједно представља и класу. На тај начин је проблем прилагођен да одговара бинарној класификацији (6. податак се не користи у класификацији). Други скуп података има 569 података, а сваки по 32 колоне. Прва колона је идентификациони број, па је он у старту елиминисан из разматрања. Следећа је информација о класи податка. Она категорише тип тумора на бенигни или малигни. Преосталих 30 атрибута су израчунати на основу дигитализоване слике FNA (енг. fine needle aspirate).

У овом експерименту, предложени EM k-NN је упоређен са три хибридна модела за класификацију и подешавање тежина атрибута, предложена у [CLL12]. Први од три модела, означен са PSO, представља хибридну технику која користи оптимизацију роjевима и кластер алгоритам k-средина. Други, означен са CBRPSO, је унапређење првог модела које у себи садржи и CBR технику за додељивање тежина атрибута. Коначно, трећи модел, GA-CBRPSO, користи тежине које је CBR одредио као иницијалне вредности. Потом се генетским алгоритмом те тежине додатно оптимизују, све у циљу смањења грешке класификације. Поред ова 3 модела, EM k-NN је упоређен и са другим методима описаним у [CLL12]: SVM, k-NN, Наивни Бајесом, и дрветом одлучивања које користи расплинуте логике (енг. fuzzy decision tree - FDT).

У циљу фер поређења, експериментално окружење је подешено на исти начин као и у [CLL12]: величина популације је 20, док је максимални број итерација 100. За тренинг је узето 75% почетног скупа, док је остатак употребљен за тестирање модела. Извршено је 500 покретања са различитим иницијалним вредностима генератора случајних бројева. На овај начин, модел је тестиран над 500 различитих подела на тренинг и тест скуп, чиме је онемогућено да неки од метода случајно буде бољи од других. У упоредним табелама које следе, приказане су просечне вредности добијене током тих 500 извршавања. Поређење по питању времена извршавања није направљено будући да у [CLL12] времена извршавања нису приказана.

Табела 4.1 приказује упоредну анализу свих поменутих метода. Све колоне, осим последње, приказују резултате описане у [CLL12]. Последња

колона, означена са EM 5-NN, приказује резултате предложене методе. Емпиријски је показано да вредност параметра k=5 производи најмање грешке класификације. Тестиране су следеће могућности: $k = 1$, $k = 3$,..., $k = 11$. Може се закључити на основу Табеле 4.1 да је EM 5-NN бољи у погледу просечне грешке класификације од 6 упоредних метода (од укупно 7), укључујући и 2 од 3 методе предложене у [CLL12]: PSO и CBRPSO. Само је GA-CBRPSO бољи од EM 5-NN када је у питању просечна грешка класификације. Међутим, EM 5-NN је бољи од GA-CBRPSO ако се узме у обзир најбоље достигнуто решење. EM 5-NN, дакле, надмашује квалитативно 6 од 7 метода, док производи нешто лошије резултате у поређењу са GA-CBRPSO.

Табела 4.1: Поређење EM 5-NN методе са неколико метода из литературе

проблем	SVM	KNN	NB	FDT	PSO	CBR-PSO	GA-CBRPSO	EM
liver disorder								
најбоље	22.4	26.3	29.8	31.7	28.5	26.1	21.8	19.5
просечно	30.7	38.9	40.8	39.9	37.5	31.6	23.2	31.4
најгоре	36.8	45.2	41.9	41.3	49.6	47.9	45.8	48.3
breast cancer								
најбоље	1.9	3.1	8.6	8.8	7.6	6.2	2.1	1.4
просечно	6.8	10.2	12.4	13.8	9.7	7.4	2.6	5.3
најгоре	18.7	19.9	14.7	21.1	11.6	9.2	3.7	10.5

Експеримент 2. У другом експерименту су коришћени подаци из студије о вођењу грађевинских пројеката [Sur]. Сваки податак је сачињен од улазних вредности, тзв. вредносних парамаметара (енг. value parameters - VP), и излазних вредности, тзв. критичних фактора успеха (енг. critical success factor - CSF). База ових података је оформљена на следећи начин. Контактирано је 263 руководилаца грађевинских пројеката. Одговори 142 руководиоца су прибављени и укључени у базу. Вредносни параметри су добијени рангирањем од стране руководилаца по питању следећих 6 својстава пословног процеса руковођења: 1) максимизација пословне ефикасности; 2) ефективност руковођења и испоручивања; 3) уклапање у финансијске оквире; 4) минимизација трошкова изградње, одржавања и утицај на околину; 5) позитиван утицај локације и 6) одговарање на специфичне захтеве. Излазни параметри, које је потребно предвидети кроз CBR систем су: 1) поље примене;

2) време; 3) трошкови; 4) квалитет изградње; 5) уговори и администрација; 6) људски ресурси; 7) ризици; 8) здравље и сигурност. Све укупно, база се састоји од 142 податка, а сваки од њих од 6 улазних и 8 излазних информација. Формирано је 10 различитих подела ове базе података на тренинг и тест скуп у односу 2:1. На тај начин се смањује могућност да нека од метода које се пореде буде случајно боља од неке друге. Поређење је извршено између 3 методе: основне методе најближих суседа (k-NN), у којој је сваком својству додељена иста тежина; предложене EM k-NN методе и вештачке неуронске мреже - ANN (Matlab имплементација). У циљу фер поређења, параметри основног k-NN и ANN су тако подешени да производе најбоље резултате по питању MSE вредности. Ово је урађено варирањем вредности k у случају k-NN, и варирањем вредности h, која представља број неурона у средишњем слоју мреже, у случају ANN. У Табели 4.2 је приказана зависност вредности параметра k и квалитета предвиђања, односно MSE.

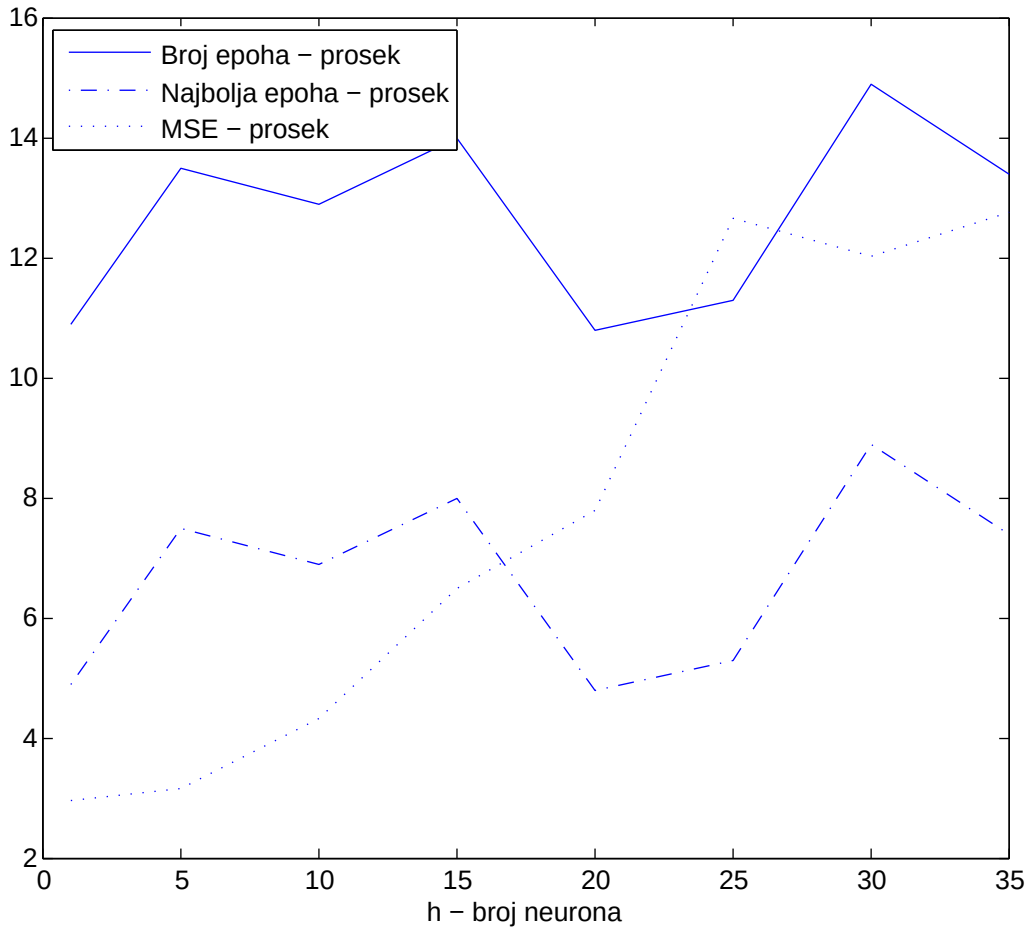
Табела 4.2: MSE за различите вредности параметра k у k-NN моделу

проблем	k=1	k=3	k=5	k=7	k=9	k=11	k=13	k=15
cpm1	2.753	1.779	1.745	1.833	1.800	1.813	1.896	1.852
cpm2	2.805	2.279	2.233	2.363	2.308	2.332	2.383	2.453
cpm3	2.268	1.733	1.620	1.555	1.607	1.562	1.507	1.502
cpm4	3.497	2.670	2.509	2.403	2.527	2.484	2.526	2.499
cpm5	2.667	2.374	2.330	2.264	2.381	2.376	2.508	2.503
cpm6	2.417	2.116	1.916	1.888	2.041	1.996	2.054	2.091
cpm7	2.808	2.158	2.141	2.156	2.062	2.173	2.033	2.161
cpm8	2.389	1.984	1.830	1.664	1.859	1.767	1.711	1.779
cpm9	2.866	2.585	2.411	2.415	2.477	2.397	2.429	2.481
cpm10	3.118	1.978	2.043	2.132	2.091	2.217	2.142	2.181
средњи ранг	8.000	4.800	2.800	2.800	4.200	3.900	4.400	5.100

Последњи ред приказује средњи ранг за различите вредности k. Средњи ранг се добија тако што се за сваки тест проблем рангирају различите методе, у овом случају k-NN методе за различите вредности параметра k. Ако посматрамо један скуп података, на пример cpm1, најбоља метода (5-NN) добија ранг 1, следећи по MSE је 3-NN, па он добија ранг 2, итд. Када се израчунају рангови за све тест проблеме и све вредности k, рачуна се просечна вредност

ранга за свако k. Може се видети да су 5-NN и 7-NN добили најмање (најбоље) просечне рангове. За даље поређење се бира 5-NN јер је увек пожељно имати једноставнији модел.

Слична процедура је урађена и за ANN. Извршена су тестирања за различите бројеве неурона у скривеном слоју мреже. На Слици 4.5 су приказане зависности броја епоха (епоха је вид итерације у процесу учења код ANN), просечне најбоље епохе и просечне MSE од броја скривених неурона. Најбоља епоха представља епоху у којој је достигнута најмања MSE. Може се приметити да је просечна најбоља епоха у свим случајевима довољно мања од просечног броја извршених епоха, што упућује на закључак да је неуронским мрежама остављено довољно времена за учење. Из Табеле 4.3 се може видети да се ANN понаша најбоље када је $h = 1$. Тада је просечни ранг од 1.8 најмањи.



Слика 4.5: Зависност MSE, броја епоха и најбоље епохе од броја скривених неурона

Табела 4.3: MSE за различите вредности параметра h у ANN

проблем	h=1	h=5	h=10	h=15	h=20	h=25	h=30	h=35
cmp1	2.814	2.275	4.810	10.487	8.298	7.512	16.448	19.997
cmp2	2.752	2.660	3.023	4.941	5.757	16.075	9.155	24.528
cmp3	2.490	3.496	2.612	3.829	8.662	9.484	6.846	6.789
cmp4	4.096	3.374	3.114	3.293	4.237	7.838	8.062	9.681
cmp5	3.547	3.115	3.771	3.290	2.337	4.767	8.062	3.931
cmp6	2.401	3.040	9.116	5.739	3.335	48.883	24.666	16.781
cmp7	2.820	2.924	5.605	9.854	21.552	8.487	17.892	7.658
cmp8	2.794	3.646	3.535	7.594	6.310	5.261	7.692	7.605
cmp9	3.230	3.550	3.972	10.947	12.847	12.397	9.276	13.713
cmp10	2.709	3.379	3.714	4.762	4.549	5.640	12.009	16.681
просечни ранг	1.800	2.100	3.000	4.500	5.000	6.100	6.700	6.800

Поређење је направљено између метода: 5-NN, EM 5-NN, и ANN која користи 1 неурон у скривеном слоју. Резултати, приказани у Табели 4.4 показују да је EM k-NN супериоран у односу на друге два метода, јер производи најбоље решење у 9 од 10 случајева. Евидентна разлика у квалитету је потврђена кроз статистички тест. Најпре је коришћен Фридманов (Friedman) тест како би се утврдило да ли постоји разлика у квалитету сва 3 метода, а након тога су извршена и даља тестирања над паровима метода [Dem06]. Фридманов тест је показао да постоји статистички значајна разлика када је у питању MSE $\chi^2(2) = 18.2, p = 0.000 (< 0.05)$. Фридманов тест може само да потврди или негира постојање статистичке различитости, али не и да укаже где се разлика, ако постоји, манифестује. Стога је извршен Вилкоксонов тест (Wilcoxon) за сваки од 3 пара метода. Показано је да EM 5-NN квалитативно надмашује 5-NN и ANN, и да је основни 5-NN модел бољи од ANN.

4.5 Завршна разматрања

У овој секцији је представљена EM метода за решавање проблема подешавања тежина атрибута у оквиру k-NN. Репрезентација EM тачака, заснована на реалним низовима се показала као погодна у решавању овог проблема јер је пресликавање из простора тежина у простор EM тачака дефинисано директном

Табела 4.4: Поређење 5-NN, EM 5-NN и ANN методе према вредности MSE

проблем	5-NN	EM 5-NN	ANN (h=1)
cmp1	1.749	1.745	2.814
cmp2	2.381	2.233	2.752
cmp3	1.662	1.620	2.490
cmp4	2.555	2.509	4.096
cmp5	2.297	2.330	3.547
cmp6	1.962	1.916	2.401
cmp7	2.182	2.141	2.820
cmp8	1.915	1.830	2.794
cmp9	2.473	2.411	3.230
cmp10	2.158	2.043	2.709
средњи ранг	1.900	1.100	3.000

доделом. Ова уска повезаност између домена посматраног проблема и домена EM методе је омогућила *глатко* пролажење кроз простор могућих решења, што је допринело налажењу добрих комбинација тежинских фактора. Спроведена су два експериментална тестирања. У првом, предложена метода је била примењена у домену класификације. Показало се да је EM k-NN бољи од 6 поређених метода (од укупно 7). У другом експерименту, радни оквир је било проналажење сличних случајева у процесу закључивања на основу случајева (case-based reasoning). Метода је примењена над тест проблемима добијеним из студије о руковођењу грађевинских пројеката. Показало се да је EM k-NN боља од основне k-NN методе и од вештачке неуронске мреже. EM k-NN је надмашила друге две методе у 9 од 10 случајева када је у питању MSE. Статистичко тестирање је показало да је квалитет EM k-NN значајно бољи од квалитета k-NN и ANN. У будућим истраживањима, могуће је побољшати EM k-NN методу тако да се може користити и за симболичке и сложене типове атрибута. Други правац развоја води у изградњу целокупног система за закључивање на основу случајева, у којем би EM k-NN имала централну улогу.

Поглавље 5

Закључак

У овом раду су истражена три проблема чијим се решавањем може унапредити квалитет класификације. То су проблем одабира атрибута, проблем подешавања тежина атрибута, и проблем подешавања параметара класификатора. За сваки од та три проблема, предложена је популациона метахеуристика заснована на електромагнетизму.

Код проблема одабира атрибута, решење је представљено бинарним низовима, где вредност 0 значи да атрибут није укључен, а вредност 1 значи да је укључен у скуп одабраних атрибута. EM тачке, које представљају носаче процеса претраге решења у методи заснованој на електромагнетизму, представљене су низовима реалних вредности из интервала [0,1]. Из тог разлога, реална репрезентација EM тачака је прилагођена проблему одабира атрибута коришћењем "праг" вредности од 0.5. Функција циља је заснована на тачности класификације и броју одабраних атрибута. То је утицало на повећање ефикасности предложене методе, јер је у поређењу са другим техникама из литературе, EM достизао боље или подједнако добре резултате уз мању употребу рачунарских ресурса. При решавању овог проблема, предложено је још неколико побољшања. Локална претрага је сачињена из две процедуре, једне која је врло ефикасна и заснива се на једној замени атрибута, и друге која је мање ефикасна, али има већи потенцијал проналажења добрих решења. Друго побољшање је скалирање EM решења које је омогућило бољу контролу над процесом претраге. Правилним одабиром скалирајућег фактора обезбеђен је добар компромис између интензификације и диверсификације претраге. Последње побољшање у овом раду је кеширање решења које је драстично

убрзало методу, што се директно види из експерименталних резултата(на неким тест проблемима и до 200 пута). Предложени ЕМ метод за одабир атрибута је допринео да се квалитет класификације, код методе најближих суседа и методе подржавајућих вектора, значајно побољша, и буде бољи од већине упоредних метода из литературе, а уз то је својом ефикасношћу допринео да процес одабира траје краће и до неколико редова величине.

Код проблема одређивања тежина атрибута, као и код проблема подешавања параметара класификатора, решења су представљена низовима реалних вредности. То је погодно својство код примене ЕМ методе, јер се врши пресликавање из реалног домена вредности ЕМ тачке у такође реални простор допустивих решења. На тај начин је омогућен *гладак* прелаз у простор могућих решења.

Адекватне комбинације тежина атрибута су омогућиле значајна побољшања тачности класификације код методе најближих суседа (k-NN). Допустиви скуп тежина атрибута је било могуће ефективно претражити из два разлога. Прво, претходно описана репрезентација решења је погодовала процесу ЕМ претраге. Друго, уведена је ефикасна локална претрага која може динамички да ажурира смер и интензитет промене вредности тежина. Квалитет методе је тестиран над два разнородна скупа тест проблема. У првом је разматран проблем класификације. У другом, ЕМ метод за подешавање тежина атрибута k-NN искоришћен је за побољшавање процеса налажења сличних случајева у оквиру система за закључивање на основу случајева. Предложена ЕМ метода је била значајно боља од других упоредних метода, што је потврђено и статистичким тестовима.

Применљивост ЕМ методе за подешавање параметара класификатора демонстрирана је на методи подржавајућих вектора (SVM). Примена над овом класификационом методом је мотивисана интерном параметарском структуром SVM, која може да буде врло сложена, односно сачињена из великог броја параметара са широким опсегом реалних вредности. Битан аспекат ЕМ методе за овај проблем је чинила специфична процедура за иницијализацију ЕМ тачака у којој се користе региони *добрих* асимптотских комбинација SVM параметара. Ово је омогућило значајно смањивање потребних временских ресурса за достизање простора квалитетних решења. Као и код проблема

подешавања тежина атрибута, у локалној претрази је динамички одређиван смер и интензитет померања по вредностима параметара. Извршена су темељна експериментална поређења над скуповима малих, средњих и великих димензија. EM метод је на већини тест проблема надмашио друге методе или био упоредив са њима. Спроведени статистички тестови су учврстили значај ових тврдњи.

Предложене EM технике су примењене над два класификациона метода: методи најближих суседа у случају проблема одабира атрибута и подешавања тежина, и на методи подржавајућих вектора код проблема подешавања параметара и проблема одабира атрибута. Све предложене EM методе се могу релативно лако проширити и прилагодити употреби над другим класификационим моделима. Код проблема одабира атрибута и подешавања тежина је то евидентно јер се та два проблема реализују ван класификационог процеса, тј. немају никакве везе са типом класификатора. Додатним разматрањем се може показати општост EM приступа у проблему подешавања параметара било ког класификатора. Наиме, заједничко за већину класификационих метода које користе параметре, на пример, дрвета одлучивања или вештачке неуронске мреже, је да им је параметарска структура заснована на векторима вредности из непрекидног или дискретног домена. Предложена метода је у стању да подеси произвољну параметарску структуру сачињену од вектора реалних вредности, а једина комуникација са класификатором је заснована на прихватању вредности функције циља, на пример, тачности класификације. Све три EM методе се, дакле, са класификационим моделом могу повезати по принципу "црне кутије" (енг. black box). Систем размене информација је након тога једноставан, EM метод прослеђује класификатору решење у виду комбинација тежина, подскупа одабраних атрибута или параметара, а као повратну информацију прихвата меру квалитета по којој се врши процес оптимизације.

5.1 Научни допринос рада

Најважнији резултати који представљају научни допринос ове дисертације су:

- Побољшан је процес одређивања раздвајајуће хиперравни при решавању произвољног класификационог проблема развојем ефективне

оптимизационе методе која користи идеју електромагнетизма за подешавање параметара SVM;

- Интерна репрезентација решења у предложеној методи, заснована на вектору реалних вредности, омогућава *глатко* пресликавање из простора EM тачака у простор параметара;

- Један од есенцијалних проблема класификације, проблем одабира атрибута, је изузетно успешно и ефикасно решен помоћу метахеуристике засноване на електромагнетизму. Добро осмишљена репрезентација решења је допринела значајном смањењу броја атрибута и грешке класификације у широком опсегу проблема преузетих из праксе;

- Развијена је EM метода за подешавање тежина атрибута која се, поред примене у класификацији, може применити и у решавању проблема закључивања на основу претходних случајева.

- Све предложене EM методе се могу применити у општем класификационом оквиру, без обзира на то који се класификациони модел користи.

Развијене методе су од изузетног значаја, јер врло успешно и ефикасно решавају широк спектар класификационих проблема великих димензија који су добијени директно из праксе.

Истраживање приказано у овом раду представља допринос у областима класификације, машинског учења, комбинаторне и глобалне оптимизације. Неки од резултата ове дисертације су објављени у међународним и домаћим часописима, док је значајан део резултата у фази припреме за објављивање.

Литература

[AP94] A. Aamodt and E. Plaza. "Case-based reasoning: Foundational issues, methodological variations, and system approaches". *AI communications* 7(1) (1994), pp. 39–59.

[AK09] H. Ahn and K.-j. Kim. "Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach". *Applied Soft Computing* 9(2) (2009), pp. 599–607.

[AMBM13] S. Al-Muhaideb and M. Bachir Menai. "Hybrid Metaheuristics for Medical Data Classification" in *Hybrid Metaheuristics*. Ed. by E.-G. Talbi. Vol. 434. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2013, pp. 187–217.

[Alf+11] A. Alford, K. Popplewell, G. V. Dozier, K. S. Bryant, J. Kelly, J. Adams, T. Abegaz, and J. Shelton. "GEFeWS: A Hybrid Genetic-Based Feature Weighting and Selection Algorithm for Multi-Biometric Recognition." in *MAICS*. Citeseer. 2011, pp. 86–90.

[AG10] M. Ali and M. Golalikhani. "An electromagnetism-like method for nonlinearly constrained global optimization". *Computers and Mathematics with Applications* 60(8) (2010), pp. 2279 –2285.

[ASS01] E. L. Allwein, R. E. Schapire, and Y. Singer. "Reducing multiclass to binary: a unifying approach for margin classifiers". *Journal of Machine Learning Research* 1 (Sept. 2001), pp. 113–141.

[AK98] E. Amaldi and V. Kann. "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems". *Theoretical Computer Science* 209(1) (1998), pp. 237–260.

[AMM12] M. Aneesh, A. A. Masand, and K. Manikantan. "Optimal Feature Selection based on Image Pre-processing using Accelerated Binary Particle Swarm Optimization for Enhanced Face Recognition". *Procedia Engineering* 30(0) (2012), pp. 750 –758.

[AKA11] I. Aydin, M. Karakose, and E. Akin. "A multi-objective artificial immune algorithm for parameter optimization in support vector machine". *Applied Soft Computing* 11(1) (2011), pp. 120–129.

[BLJ04] F. R. Bach, G. R. Lanckriet, and M. I. Jordan. "Multiple kernel learning, conic duality, and the SMO algorithm" in *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 6.

[BL13] K. Bache and M. Lichman. *UCI Machine Learning Repository*. 2013.

[BJLLD09] A. Barbero Jiménez, J. López Lázaro, and J. R. Dorronsoro. "Finding optimal model parameters by deterministic and annealed focused grid search". *Neurocomputin* 72(13-15) (Aug. 2009), pp. 2824–2832.

[BP06] G. R. Beddoe and S. Petrovic. "Selecting and weighting features using a genetic algorithm in a case-based reasoning approach to personnel rostering". *European Journal of Operational Research* 175(2) (2006), pp. 649–671.

[BF03] S. I. Birbil and S. C. Fang. "An electromagnetism-like mechanism for global optimization". *Journal of Global Optimization* 25 (2003), pp. 263–282.

[BFS04] S. I. Birbil, S. C. Fang, and R. L. Sheu. "On the Convergence of a Population-Based Global Optimization Algorithm". *Journal of Global Optimization* 30 (2004), pp. 301–318.

[BN06] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 1. springer New York, 2006.

[BS10] J. Blondin and A. Saad. "Metaheuristic techniques for Support Vector Machine model selection" in *Hybrid Intelligent Systems (HIS), 2010 10th International Conference on*. 2010, pp. 197–200.

[BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik. "A training algorithm for optimal margin classifiers" in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM. 1992, pp. 144–152.

[BM02] H. Brighton and C. Mellish. "Advances in instance selection for instance-based learning algorithms". *Data mining and knowledge discovery* 6(2) (2002), pp. 153–172.

[BL07] M. Bursa and L. Lhotska. "Automated Classification Tree Evolution Through Hybrid Metaheuristics". English in *Innovations in Hybrid Intelligent Systems*. Ed. by E. Corchado, J. Corchado, and A. Abraham. Vol. 44. Advances in Soft Computing. Springer Berlin Heidelberg, 2007, pp. 191–198.

[CY11] C. Campbell and Y. Ying. "Learning with support vector machines". *Synthesis Lectures on Artificial Intelligence and Machine Learning* 5(1) (2011), pp. 1–95.

[Can11] A. Candelieri. "A hyper-solution framework for classification problems via metaheuristic approaches". English. *4OR* 9(4) (2011), pp. 425–428.

[CMBRM12] E. Carrizosa, B Martín-Barragán, and D Romero Morales. *Variable neighborhood search for parameter tuning in support vector machines*. Tech. rep. 2012.

[CL11] C.-C. Chang and C.-J. Lin. "LIBSVM: A library for support vector machines". *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011), 27:1–27:27.

[CLL12] P.-C. Chang, J.-J. Lin, and C.-H. Liu. "An attribute weight assignment and particle swarm optimization algorithm for medical database classifications". *Computer methods and programs in biomedicine* 107(3) (2012), pp. 382–392.

[Cha+02] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. "Choosing Multiple Parameters for Support Vector Machines". English. *Machine Learning* 46 (1-3 2002), pp. 131–159.

[CSC12] L.-F. Chen, C.-T. Su, and K.-H. Chen. "An improved particle swarm optimization for feature selection". *Intelligent Data Analysis* 16 (2 2012), pp. 167–182.

[CG10] D. Conforti and R. Guido. "Kernel based support vector machine via semidefinite programming: Application to medical diagnosis". *Computers and Operations Research* 37(8) (2010), pp. 1389 –1394.

[CV95] C. Cortes and V. Vapnik. "Support-vector networks". *Machine Learning* 20 (3 1995), pp. 273–297.

[ČLM12] M. Črepinšek, S.-H. Liu, and L. Mernik. "A note on teaching–learning-based optimization algorithm". *Information Sciences* 212 (2012), pp. 79–93.

[Cue+12] E. Cuevas, D. Oliva, D. Zaldivar, M. Pérez-Cisneros, and H. Sossa. "Circle detection using electro-magnetism optimization". *Information Sciences* 182(1) (2012), pp. 40 –55.

[Cun09] P. Cunningham. "A taxonomy of similarity mechanisms for case-based reasoning". *Knowledge and Data Engineering, IEEE Transactions on* 21(11) (2009), pp. 1532–1543.

[Dem06] J. Demšar. "Statistical comparisons of classifiers over multiple data sets". *Journal of Machine Learning Research* 7 (2006), pp. 1–30.

[DM02] F. X. Diebold and R. S. Mariano. "Comparing predictive accuracy". *Journal of Business & economic statistics* 20(1) (2002).

[Dor92] M. Dorigo. "Optimization, learning and natural algorithms". *Ph. D. Thesis, Politecnico di Milano, Italy* (1992).

[DKP03] K. Duan, S. Keerthi, and A. N. Poo. "Evaluation of simple performance measures for tuning SVM hyperparameters". *Neurocomputing* 51(0) (2003), pp. 41 –59.

[DHS12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[DH10] B. Duval and J.-K. Hao. "Advances in metaheuristics for gene selection and classification of microarray data". *Briefings in Bioinformatics* 11(1) (2010), pp. 127–141. eprint: `http://bib.oxfordjournals.org/content/11/1/127.full.pdf+html`.

[Fil11] V. Filipović. "An Electromagnetism Metaheuristic for the Uncapacitated Multiple Allocation Hub Location Problem". *Serdica Journal of Computing* 5(3) (2011), pp. 261–272.

[FKM13] V. Filipović, A. Kartelj, and D. Matić. "An electromagnetism metaheuristic for solving the Maximum Betweenness Problem". *Applied Soft Computing* 13(2) (2013), pp. 1303 –1313.

[FML02] D. Fragoudis, D. Meretakis, and S. Likothanassis. "Integrating feature and instance selection for text classification" in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2002, pp. 501–506.

[FH03] V. Franc and V. Hlaváč. "Greedy algorithm for a training set reduction in the kernel methods" in *Computer Analysis of Images and Patterns*. Springer. 2003, pp. 426–433.

[FA10] A. Frank and A. Asuncion. *UCI Machine Learning Repository*. 2010.

[FI05] F. Friedrichs and C. Igel. "Evolutionary tuning of multiple SVM parameters". *Neurocomputing* 64 (2005), pp. 107–117.

[GT+04] M. Garcıa-Torres, F. Garcıa-López, B. Melián-Batista, J. A. Moreno-Pérez, and J. M. Moreno-Vega. "Solving feature subset selection problem by a hybrid metaheuristic" in *First International Workshop in Hybrid Metaheuristics at ECAI*. 2004, pp. 59–69.

[GM+11] J Gascón-Moreno, E. G. Ortiz-García, S. Salcedo-Sanz, A Paniagua-Tineo, B Saavedra-Moreno, and J. A. Portilla-Figueras. "Multi-parametric gaussian kernel function optimization for ε-SVMr using a genetic algorithm" in *Advances in Computational Intelligence*. Springer, 2011, pp. 113–120.

[GM+13] J Gascón-Moreno, E. Ortiz-García, S. Salcedo-Sanz, L. Carro-Calvo, B. Saavedra-Moreno, and A. Portilla-Figueras. "Evolutionary optimization of multi-parametric kernel ε-SVMr for forecasting problems". English. *Soft Computing* 17(2) (2013), pp. 213–221.

[Glo89] F. Glover. "Tabu search-part I". *ORSA Journal on computing* 1(3) (1989), pp. 190–206.

[GA11] M. Gönen and E. Alpaydın. "Multiple kernel learning algorithms". *The Journal of Machine Learning Research* 12 (2011), pp. 2211–2268.

[GE03] I. Guyon and A. Elisseeff. "An introduction to variable and feature selection". *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182.

[Hor+12] M.-H. Horng, Y.-X. Lee, M.-C. Lee, and R.-J. Liou. "Firefly metaheuristic algorithm for training the radial basis function network for data classification and disease diagnosis". *Theory and new applications of swarm intelligence* (2012), pp. 115–132.

[Hua+12] H. Huang, H.-B. Xie, J.-Y. Guo, and H.-J. Chen. "Ant colony optimization-based feature selection method for surface electromyography signals classification". *Computers in Biology and Medicine* 42(1) (2012), pp. 30 –38.

[HH09] W.-M. Hung and W.-C. Hong. "Application of SVR with improved ant colony optimization algorithms in exchange rate forecasting". *Control and Cybernetics* 38(3) (2009), pp. 863–891.

[IP07] K. H. Im and S. C. Park. "Case-based reasoning and neural network based expert system for personalization". *Expert Systems with Applications* 32(1) (2007), pp. 77–85.

[IL04] F Imbault and K Lebart. "A stochastic optimization approach for parameter tuning of support vector machines" in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 4. IEEE. 2004, pp. 597–600.

[Joa00] T. Joachims. "Estimating the generalization performance of a SVM efficiently" (2000).

[KI+10] M. Kabir, M. Islam, et al. "A new wrapper feature selection approach using neural network". *Neurocomputing* 73(16) (2010), pp. 3273–3283.

[Kar12] A. Kartelj. "Electromagnetism metaheuristic algorithm for solving the strong minimum energy topology problem". *Yugoslav Journal of Operations Research* 22(2) (2012).

[Kar] A. Kartelj. "An Improved Electromagnetism-like Method for Feature Selection". submitted.

[KŠC14] A. Kartelj, N. Šurlan, and Z. Cekic. "Case-based reasoning and electromagnetism-like method in construction management". *Kybernetes* 43(2) (2014), pp. 265–280.

[Kar+13] A. Kartelj, N. Mitić, V. Filipović, and D. Tošić. "Electromagnetism-like algorithm for support vector machine parameter tuning". *Soft Computing* (2013), pp. 1–14.

[Kec01] V. Kecman. *Learning and soft computing: support vector machines, neural networks, and fuzzy logic models*. MIT press, 2001.

[Kee02] S. S. Keerthi. "Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms". *IEEE Transactions on Neural Networks* 13(5) (2002), pp. 1225–1229.

[KL03] S. S. Keerthi and C.-J. Lin. "Asymptotic behaviors of support vector machines with Gaussian kernel". *Neural Computation* 15(7) (July 2003), pp. 1667–1689.

[KV+83] S. Kirkpatrick, M. Vecchi, et al. "Optimization by simmulated annealing". *science* 220(4598) (1983), pp. 671–680.

[KCB04] D. Korycinski, M. M. Crawford, and J. W. Barnes. "Adaptive feature selection for hyperspectral data analysis" in *Remote Sensing*. International Society for Optics and Photonics. 2004, pp. 213–225.

[Kra12] J. Kratica. “An electromagnetism-like method for the maximum set splitting problem”. *Yugoslav Journal of Operations Research* 23(1) (2012).

[KJ99] L. I. Kuncheva and L. C. Jain. “Nearest neighbor classifier: Simultaneous editing and feature selection”. *Pattern Recognition Letters* 20(11–13) (1999), pp. 1149 –1156.

[LD06] N. Lavesson and P. Davidsson. “Quantifying the impact of learning algorithm parameter tuning” in *AAAI*. Vol. 6. 2006, pp. 395–400.

[Leb+05] G. Lebrun, C. Charrier, O. Lezoray, C. Meurie, and H. Cardot. “Fast Pixel Classification by SVM Using Vector Quantization, Tabu Search and Hybrid Color Space” in *Computer Analysis of Images and Patterns*. Ed. by A. Gagalowicz and W. Philips. Vol. 3691. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 685–692.

[LV91] J.-H. Lin and J. S. Vitter. “Complexity results on learning by neural nets”. *Machine Learning* 6(3) (1991), pp. 211–230.

[LC11] S.-W. Lin and S.-C. Chen. “Parameter tuning, feature selection and weight assignment of features for case-based reasoning by artificial immune system”. *Applied Soft Computing* 11(8) (2011), pp. 5042–5052.

[Lin+08] S.-W. Lin, Z.-J. Lee, S.-C. Chen, and T.-Y. Tseng. “Parameter determination of support vector machine and feature selection using simulated annealing approach”. *Applied Soft Computing* 8(4) (2008), pp. 1505 –1512.

[Lug11] E. Lughofer. “On-line incremental feature weighting in evolving fuzzy classifiers”. *Fuzzy Sets and Systems* 163(1) (2011). Theme: Classification and Modelling, pp. 1 –23.

[MFM11] H. Malik, D. Fradkin, and F. Moerchen. “Single pass text classification by direct feature weighting”. English. *Knowledge and Information Systems* 28(1) (2011), pp. 79–98.

[MMZ10] M. Marinaki, Y. Marinakis, and C. Zopounidis. “Honey Bees Mating Optimization algorithm for financial classification problems”. *Applied Soft Computing* 10(3) (2010), pp. 806 –812.

[MDJ09] Y. Marinakis, G. Dounias, and J. Jantzen. "Pap smear diagnosis using a hybrid intelligent scheme focusing on genetic algorithm based feature selection and nearest neighbor classification". *Computers in Biology and Medicine* 39(1) (2009), pp. 69 –78.

[Mar+08] Y. Marinakis, M. Marinaki, M. Doumpos, N. Matsatsinis, and C. Zopounidis. "Optimization of nearest neighbor classifiers via metaheuristic algorithms for credit risk assessment". English. *Journal of Global Optimization* 42(2) (2008), pp. 279–293.

[Mar+09] Y. Marinakis, M. Marinaki, M. Doumpos, and C. Zopounidis. "Ant colony and particle swarm optimization for financial classification problems". *Expert Systems with Applications* 36(7) (2009), pp. 10604 –10611.

[MAT10] E. Martinez, M. M. Alvarez, and V. Trevino. "Compact cancer biomarkers discovery using a swarm intelligence feature selection algorithm". *Computational Biology and Chemistry* 34(4) (2010), pp. 244 –250.

[MLY11] J. Meng, H. Lin, and Y. Yu. "A two-stage feature selection method for text categorization". *Computers and Mathematics with Applications* 62(7) (2011). Computers & Mathematics in Natural Computation and Knowledge Discovery, pp. 2793 –2800.

[MST94] D. Michie, D. J. Spiegelhalter, and C. Taylor. *Machine Learning, Neural and Statistical Classification*. 1994.

[MH97] N. Mladenović and P. Hansen. "Variable neighborhood search". *Computers & Operations Research* 24(11) (1997), pp. 1097–1100.

[MS07] V. G. J. S. R. Mollineda and R. A. J. Sotoca. "The class imbalance problem in pattern classification and learning" (2007).

[Mul+01] K Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. "An introduction to kernel-based learning algorithms". *Neural Networks, IEEE Transactions on* 12(2) (2001), pp. 181–201.

[NATG10] Z. Naji-Azimi, P. Toth, and L. Galli. "An electromagnetism metaheuristic for the unicost set covering problem". *European Journal of Operational Research* 205(2) (2010), pp. 290–300.

[NF77] P. M. Narendra and K. Fukunaga. "A branch and bound algorithm for feature subset selection". *Computers, IEEE Transactions on* 100(9) (1977), pp. 917–922.

[PVP11] P. K. P, P. Vadakkepat, and L. A. Poh. "Fuzzy-rough discriminative feature selection and classification algorithm, with application to microarray and image datasets". *Applied Soft Computing* 11(4) (2011), pp. 3429 –3440.

[PCN07] J. Pacheco, S. Casado, and L. Nuñez. "Use of VNS and TS in classification: variable selection and determination of the linear discrimination function coefficients". *IMA Journal of Management Mathematics* 18(2) (2007), pp. 191–206. eprint: `http://imaman.oxfordjournals.org/content/18/2/191.full.pdf+html`.

[PK10] T. Phienthrakul and B. Kijsirikul. "Evolutionary strategies for hyperparameters of support vector machines based on multi-scale radial basis function kernels". *Soft Computing* 14(7) (2010), pp. 681–699.

[Pol12] K. Polat. "Classification of Parkinson's disease using feature weighting method on the basis of fuzzy C-means clustering". *International Journal of Systems Science* 43(4) (2012), pp. 597–609. eprint: `http://www.tandfonline.com/doi/pdf/10.1080/00207721.2011.581395`.

[ROM01] G. Rätsch, T. Onoda, and K.-R. Müller. "Soft Margins for AdaBoost". English. *Machine Learning* 42 (3 2001), pp. 287–320.

[SSA10] F Samadzadegan, A Soleymani, and R. A. Abbaspour. "Evaluation of Genetic Algorithms for tuning SVM parameters in multi-class problems" in *Computational Intelligence and Informatics (CINTI), 2010 11th International Symposium on*. IEEE. 2010, pp. 323–328.

[STC04] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[SS89] W. Siedlecki and J. Sklansky. "A note on genetic algorithms for large-scale feature selection". *Pattern Recognition Letters* 10(5) (1989), pp. 335 –347.

[SR07] R. K. Sivagaminathan and S. Ramakrishnan. "A hybrid approach for feature subset selection using neural networks and ant colony optimization". *Expert systems with applications* 33(1) (2007), pp. 49–60.

[Son+06] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. "Large Scale Multiple Kernel Learning". *Journal of Machine Learning Research* 7 (Dec. 2006), pp. 1531–1565.

[SL11] C.-T. Su and H.-C. Lin. "Applying electromagnetism-like mechanism for feature selection". *Information Sciences* 181(5) (2011), pp. 972 –986.

[Sur] N. s. ˇ Surlan. "Construction project knowledge database as a decision support system through CBR application". submitted.

[TBK07] M. A. Tahir, A. Bouridane, and F. Kurugollu. "Simultaneous feature selection and feature weighting using Hybrid Tabu Search k-nearest neighbor classifier". *Pattern Recognition Letters* 28(4) (2007), pp. 438–446.

[Tal+08] E. Talbi, L. Jourdan, J. Garcia-Nieto, and E. Alba. "Comparison of population based metaheuristics for feature selection: Application to microarray data classification" in *Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on*. 2008, pp. 45–52.

[Tal09] E.-G. Talbi. *Metaheuristics: from design to implementation*. Vol. 74. John Wiley & Sons, 2009.

[TMKN09] R Tavakkoli-Moghaddam, M Khalili, and B Naderi. "A hybridization of simulated annealing and electromagnetic-like mechanism for job shop problems with machine availability and sequence-dependent setup times to minimize total weighted tardiness". *Soft Computing* 13(10) (2009), pp. 995–1006.

[UM10] A. Unler and A. Murat. "A discrete particle swarm optimization method for feature selection in binary classification problems". *European Journal of Operational Research* 206(3) (2010), pp. 528 –539.

[UMC11] A. Unler, A. Murat, and R. B. Chinnam. "mr^2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification". *Information Sciences* 181(20) (2011), pp. 4625–4641.

[VHM94] K. S. Van Horn and T. R. Martinez. "The minimum feature set problem". *Neural Networks* 7(3) (1994), pp. 491–494.

[Vap95] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[Vap99] V. N. Vapnik. "An overview of statistical learning theory". *Neural Networks, IEEE Transactions on* 10(5) (1999), pp. 988–999.

[VSK12] S. M. Vieira, J. M. Sousa, and U. Kaymak. “Fuzzy criteria for feature selection”. *Fuzzy Sets and Systems* 189(1) (2012), pp. 1–18.

[Wan+12] J. Wang, A.-R. Hedar, S. Wang, and J. Ma. “Rough set and scatter search metaheuristic based feature selection for credit scoring”. *Expert Systems with Applications* 39(6) (2012), pp. 6123 –6128.

[WM94] I. Watson and F. Marir. “Case-based reasoning: A review”. *Knowledge Engineering Review* 9(4) (1994), pp. 327–354.

[WAM97] D. Wettschereck, D. W. Aha, and T. Mohri. “A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms”. *Artificial Intelligence Review* 11(1-5) (1997), pp. 273–314.

[Yan+11] J. Yang, Y. Liu, Z. Liu, X. Zhu, and X. Zhang. “A new feature selection algorithm based on binomial hypothesis testing for spam filtering”. *Knowledge-Based Systems* 24(6) (2011), pp. 904 –914.

[YH98] J. Yang and V. G. Honavar. “Feature Subset Selection Using a Genetic Algorithm”. *IEEE Intelligent Systems* 13(2) (Mar. 1998), pp. 44–49.

[YWD05] C.-Y. Yeh, C.-H. Wu, and S.-H. Doong. “Effective spam classification based on meta-heuristics” in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*. Vol. 4. 2005, 3872–3877 Vol. 4.

[YE10] A. Yurtkuran and E. Emel. “A new Hybrid Electromagnetism-like Algorithm for capacitated vehicle routing problems”. *Expert Systems with Applications* 37(4) (2010), pp. 3427 –3433.

[Yus09] S. C. Yusta. “Different metaheuristic strategies to solve the feature selection problem”. *Pattern Recognition Letters* 30(5) (2009), pp. 525 –534.

[ZS02] H. Zhang and G. Sun. “Feature selection using tabu search method”. *Pattern Recognition* 35(3) (2002), pp. 701 –711.

[ZCH10] X. Zhang, X. Chen, and Z. He. “An ACO-based algorithm for parameter optimization of support vector machines”. *Expert Systems with Applications* 37(9) (2010), pp. 6618 –6628.

[ZHQ13] G. Zhiqiang, W. Huaiqing, and L. Quan. “Financial time series forecasting using LPP and SVM optimized by PSO”. English. *Soft Computing* 17(5) (2013), pp. 805–818.

[Zuo+08] W. Zuo, W. Lu, K. Wang, and H Zhang. “Diagnosis of cardiac arrhythmia using kernel difference weighted KNN classifier” in *Computers in Cardiology, 2008*. IEEE. 2008, pp. 253–256.

Биографија

Основни подаци. Александар Картељ је рођен 10. новембра 1986. године у Новом Кнежевцу. У Кикинди је завршио основну школу као добитник Вукове дипломе и ђак генерације, а потом и гимназију "Душан Васиљев" као добитник Вукове дипломе. Основне академске студије на Математичком факултету, Универзитета у Београду, уписао је 2005. године, а завршио их 2008. године на смеру Информатика са просечном оценом 9,94. Мастер академске студије уписао је 2008. године, а завршио их 2010. године са просечном оценом 9,92 и одбрањеном мастер тезом под називом "Решавање проблема минималне енергетске повезаности у тежинском графу коришћењем генетског алгоритма", под менторством др Владимира Филиповића. По завршетку мастер студија уписао је докторске студије на истом факултету, модул Информатика. Положио је све испите на докторским студијама са просечном оценом 10,00.

Искуство у настави. Од 2009. до 2011. године, Александар Картељ је био запослен као сарадник у настави, а потом, 2011. године је изабран у звање асистента за научну област Рачунарство и информатика на Математичком Факултету у Београду. Држао је вежбе из следећих предмета: Програмирање 1, Програмирање 2, Дизајн програмских језика, Програмске парадигме, Образовни софтвер и Паралелни алгоритми.

Учешће на пројектима. Од 2011. до данас учесник је на научном пројекту бр. 174010 под називом “Математички модели и методе оптимизације великих система”, под руководством др Ненада Младеновића, на Математичком институту САНУ, у оквиру текућег Програма истраживања научног и технолошког развоја, финансираног од стране Министарства просвете, науке и технолошког развоја Републике Србије.

Остали подаци. Године 2009. је освојио треће место на националном такмичењу „Imagine Cup“ које је организовао Microsoft Serbia, а 2010. године је добио награду за најбољег асистента на Математичком факултету. Учествовао је у пројектовању и имплементацији 9 софтверских пакета који се користе или су били коришћени у државним институцијама и приватном сектору.

Прилог 1.

Изјава о ауторству

Потписани-а Александар Картељ

број уписа 2025/2010

Изјављујем

да је докторска дисертација под насловом

“Примене метахеуристике засноване на електромагнетизму у решавању проблема класификације”

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, _____

_____

Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора: Александар Картељ

Број уписа: 2025/2010

Студијски програм: Информатика

Наслов рада: “Примене метахеуристике засноване на електромагнетизму у решавању проблема класификације”

Ментор: др Владимир Филиповић, ванредни професор

Потписани: Александар Картељ

изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, _____

_____

Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

“Примене метахеуристике засноване на електромагнетизму у решавању проблема класификације”

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство

2. Ауторство - некомерцијално

3. Ауторство – некомерцијално – без прераде

4. Ауторство – некомерцијално – делити под истим условима

5. Ауторство – без прераде

6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

Потпис докторанда

У Београду, _____

_____

1. Ауторство - Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. Ауторство – некомерцијално. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. Ауторство - некомерцијално – без прераде. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. Ауторство - некомерцијално – делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. Ауторство – без прераде. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. Ауторство - делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.