## ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ

**ФАКУЛТЕТ ПО ТЕЛЕКОМУНИКАЦИИ**

ДЕКЛАРАЦИЯ

Долуподписаният ....................................................................................... фак. № .......................... декларирам, чепредставенатаотмендипломнатаработанатема "................................................................................................. ..........................................................................................................................." е разработенасамостоятелно. В работатасъмсесъобразил с авторскитеправанадругиизточнициилиресурси, коитосъмизползвал и съмцитирал в текста. Несъмизползвалдругиматериали, обектнаавторскоправо, освенпосочените в раздела "Използваналитература".

Декларирам, чедипломнатаработане е представянадосеганапредишнисесиизадипломназащита.

Дата: Декларатор:

гр. София

**Съдържание**

Глава1 Литературен обзор

1.1 Извличание на признаци

1.2 Обучение

1.3 Принципи на предаване на информацията от акселерометър

Глава 2 Теоретична част

2.1 Информация за базата от данни

2.2 Теоретично описание на избрания алгоритъм

Глава 3 Инженерно решение на поставената задача

Глава 4 Анализ на получените резултати, приложимост и изводи

Използвана литература

Списък на използваните означения и съкращения

**Увод**

Осигуряването на точна и подходяща информация за дейноста и държанието на хората е една от най-трудните задачи в широко разпространената изчислителна наука.Безброй са приложенията в медицината,сигурноста,забавления и др. Развитието на микроелектрониката и компютърните системи през последното десетилетие позволи създаването на сензори и мобилни устройства с достатъчно добри характеристики.Тяхната висока изчислителна способност,малък размер и ниска цена позволява на хората да си взаимодействат с такива устройства всеки ден. По-специално Разпознаване на човека дейност(Human activity recognition) се превърна в задача с доста голям интерес,особено за медицински,военни и цели за безопасност. Например за пациенти болни от диабет,затлъстяване и сърдечни болист се изисква да правят специални упражнения всеки ден като част от тяхната терапия. Затова разпознаване на дейност като ходене,тичане,каране на колело и др. става доста полезно,за да може да предостави информация на лекуващия за държанието и състоянието на пациента. По същият начин пациенти с Деменция и други проблеми с паметта могат да бъдат наблюдавани за странно поведение и следователно да бъдат предотвратени нежелани последствия. За военни цели точната информация за дейноста на войниците,тяхното местоположение и жизнени показатели е много полезно за тяхното представяне и безопасност. Такава информация също е полезна при вземане на решения в бойна ситуация.

**Глава 1**

**Литературен обзор**

Преди да може да разпознаваме човекшка дейност трябва да обработим суровите данни и да извлечем признаци от тях. След това се построява модел за разпознаване от набор от вече извлечените признаци със способите на машинното обучение. След като веднъж моделът е трениран разделеният на прозорци сигнал може да бъде оценен от модела за разпознаване и така да бъде предсказано какво действие извършваме. В тази глава ще разгледаме как теоретично се извличат признаци и как се класифицират данни.

***1.1.***Извличание на признаци

Човек извършва някаква дейност в сравнително дълъг интервал от време(секунди,минути) сравнено с честотата на дискретизация на сензора. Освен това отделен отчет във времето(ускорение от 2.5g по У компонента или 130 удара пулс) не предоставят достатъчно информация,за да се опише извършената дейност. Различните дейности трябва да се разпознават в някакъв отрязък от време, а не само по един отчет. Въпросът,който възниква е как да сравним два прозореца във времева област? Почти невъзможно е сигналите да са напълно идентични,дори ако идват от един и същ обект,извършващ една и съща дейност. Поради тази причина се извличат признаци от всеки един отрязък(прозорец) от време,с други думи извличане на полезна за нас информация и получаване на количсвени измервания,които ще позволят сравняването на сигналите.Основно има два подхода за извличане на признаци от данни във времева област: статистически и структурни. Статистическите методи като преобразувание на Фурие и Wavеlet трансформация използват количествени характеристики на данните,за да извличат признаци,където структурните подходи използват корелацията между данните. Критерият,по който се избира един от методите строго зависи от вида на сигнала и е индивидуален за всеки отделен случай. Според на фиг.1.1 е показана обработката на суров сигнал във времева област (който може да е ускорение,жизнени показатели или параметри на околната среда) в набор от признакови вектори. Всяка отделна извадка от вече обработените данни съответсва на признаков вектор извлечен от определен прозорец от време. Ще покрием най-основните техники за извличане на признаци за всеки от отделните атрибути(ускорение,параметри на околната среда и жизнени показатели).GPS данните няма да бъдат взимани в предвид ,защото те най-често се използват,за да се изчисли скоростта или за да разберем къде се е извършило действието.

1.1.Ускорение

Сигналът ,който се формира от ускорението (виж фиг.2.1)са много колебателни и с големи флуктоации ,което прави трудно разпознаването на скритите модели ,използвайки техните сурови стойности,съществуващите системи за РЧД базирани на данни от акселерометър използват извличане на статистически признаци и в повечето случаи във времева или честотна

област.Дискретно косиносово преобразуване и основен анализ на компонентите (PCA) са също прилагани с обещаващи резултати.Всички тези техники са замислени да работят със силни променящи се стойности присъщи на сигнала от ускорението .В табл.1.2 са резюмирани методите за извличане на признаци от сигнала на ускорението.Ще изброим някои от най-използваните признаци за даден сигнал:

*Y* = *{y*1*, ...,yn}*.

средна стойност у и корен квадратен от средна квадратична стойност (уравнение 1 и 2)

= (1)

RMS(Y)= (2)

стандартно отклонение, квадратично отклонение и абсолютно средно отклонение (уравнения 3,4 и 5) преминаване от една област в друга като например в енергийна,където fiei-тиякомпонент на преобразуванието на Фурие от Y (уравнение 6)

σy= (3)

= (4)

MAD(Y)=(5)

Energy(Y) = (6)

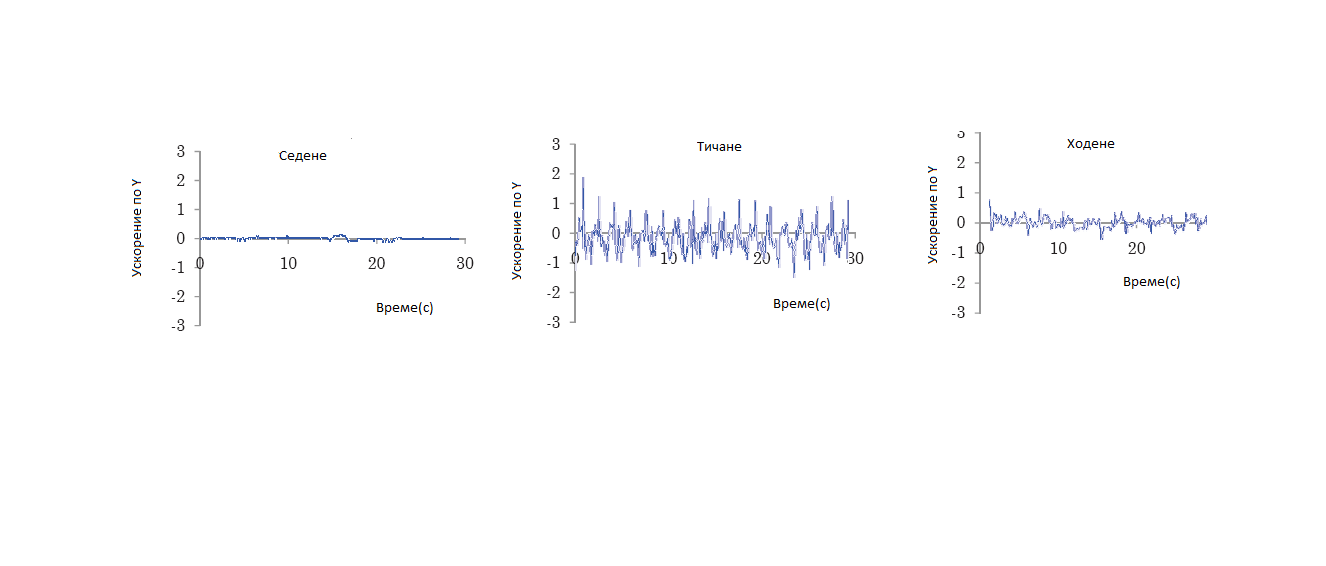
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| w | време | ax | ay | az | дейност |
| j  j  j  ...  j | 0  1/s1  1/s2  ...  tmax | 1.3  1.4  1.1  ...  1.8 | -2.1  -2.3  -2.6  ...  2.2. | 0  0.1  0  ...  -0.4 | тичане  тичане  тичане  ...  тичане |
| w | време | удара в мин. | ... | бр.вдишвания в мин | дейност |
| j  j  j  ...  j | 0  1/s1  1/s2  ...  tmax | 120  120  120  ...  120 | ...  ...  ...  ...  ... | 15  16  15  ...  18 | тичане  тичане  тичане  ...  тичане |
| w | време | температура | ... | влажност % | дейност |
| j  j  j  ...  j | 0  1/s1  1/s2  ...  tmax | 25  25  25  ...  25 | ...  ...  ...  ...  ... | 50  50  50  ...  50 | тичане  тичане  тичане  тичане |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| w | f0 | ... | fn | Дейност |
| 0  ... | 231  ... | ...  ... | -6.2  ... | Неопределена  ... |
| j | 543 | ... | 8 | тичане |
| ...  k-1 | ...  339 | ...  ... | ...  7.1 | ...  Качване на стълби |

Фиг.1.1Обработката на суров сигнал във времева област

Табл.1.2 Методи за извличане на признаци

|  |  |
| --- | --- |
| Област | Методи |
| Времева | Средна стойност,стандартно отклонени,дисперсия,ентропия,абсолютна средна стойност |
| Честотна | Преобразувване на Фурие,Дискретно косинусово преобразуване |
| Други | Анализ на основните компоненти,Авто-регресивен модел,Линеен Дискриминантен анализ,Филтри на Хар |



Фиг.1.1 Ускорение по Y оста съответно за седене,тичане и ходене

1.1.2.Променливи на средата

Променливите на средата заедно със сигнала на ускорението, се използват на обогатят косвено разпознаването.Например стойностите от налягането на въздуха и интензитета на светлината спомагат ,за да се определи дали индивида е вътре или на открито.Също аудио сигналите са полезни да се заключи дали потребителя говори с някого или слуша музика.

1.1.3.Жизнени показатели

Първоначално първите хора работили по извличане на признаци от жизнени показатели за разпознаване на човешка дейност са използвали статистически признаци например са изчислили броя на стойностите на пулса ,които са над нормалния такъв .Или са изчислили признаците във времева област на пулса дихателните усилия ,ЕКГ, артериална кислородна сатурация и температура на кожата.Обърнете внимание на ситуацията ,която е показана на фиг. 2.2 Сигнал от пулса S (t) за индивид ,който ходи е показан с удебелена линия същия сигнал само че обърнат S’(t) е показан с тънка линия.Забележете ,че повечето признаци(средна стойност,дисперсия и енергия) извлечени във времева и честотна област са идентични за двата сигнала като те могат да представляват различна дейност.Това е основната обосновка за прилагане на структурни методи за извличане на признаци .

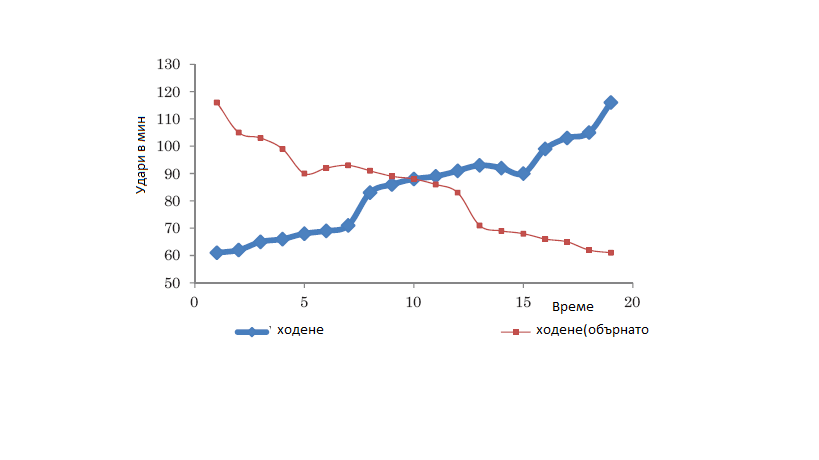
Във времева област структурните детектори имат за цел да опишат морфологичните връзки между данните.Нека имаме сигнал Y(t) структурният детектор имплементирафункция f(t)= Y’(t) такова ,че Y’(t) има структурата на Y(t) но е апроксимирана,за да може да се измери дали формата на Y’(t) достатъчно се доближава до тази на Y(t) сумата от квадратичните грешки се изчислявва както следва:

SSE=

Извлечените признаци всъщност са Y’(t) ,който естествено зависи от функцията. Табл.1.3 съдържа някои типични функции,които са имплементиране от структурни детектори. Полиномните функции са най-добрите функции за обработка на физиологични сигнали като пулс,честота на дишане,амплитуда на дишане,температура на кожата.

Табл.1.1.3 Типични функции за структурни детектори

|  |  |  |
| --- | --- | --- |
| Функция | Уравнение | Параметри |
| Линейна | *F*(*t*) = *mt*+ *b* | *{m, b}* |
| Полиномна | *F*(*t*) = *a*0 + *a*1*t* + *...* + *an−*1*tn−*1 | *{a*0*, ..., an−*1*}* |
| Експоненциална | *F*(*t*) = *a|b|t*+ *c* | *{a, b, c}* |
| Синусоидална |  | *{a, b, c}* |



Фиг.1.2 Ходене,ходене(обърнато)

1.1.4. Избор на големина на времевия прозорец

Сегментирането на сигнала на отделни отрязъци от време(времеви прозорци) е много удобен метод за улесняване работата при разпознаване на дейност. Следователно ключов фактор е избирането на големината на времевия прозорец,защото сложността при изчисление,при който и да е метод за извличане на признаци зависи от броя на отчетитe. Ако дължината на един прозорец е малка това може да подобри разпознаването,но това ще увеличи времето за изчисление,защото методът за извличане на конкретните признак/ци ще бъде повтарян повече пъти. Освен това по-кратки прозорци може да не предоставят достатъчно информация,за да се опише напълно изследваното действие. Обратно,ако прозорците са твърде дълги може да има повече от една дейност,която се извършва за този време-интервал. Разбира се решението за това колко да е продължителен прозорецът зависи от конкретната задача. Например при измерване на пулса са нужни прозорци с продължителност 30 сек. За дейност като поглъщане,пък е необходим прозорец с продължителнос 1.5 сек. Прозорците могат да бъдат припокриващи се или не.Припокриващите се прозорци са пригодени по-лесно да разграничават преминаването от едно действие в друго,ако използвате кратки не-припокриващи се прозорци,погрешната класицикация заради преход ще е незначителна.

1.1.5.Избор на признаци

Някои признаци може да носят излишна или неподходяща информация,която може да въздейства негативно на крайния резултат. Тогава се използват техники са избиране на правилните признаци за извличане. Бейсов критерии за информация и минимална дължина за описване са широко разпространени при решаването на основни проблеми на машинното обучение. При разпознаване на дейност често срещан метод е Минимален Излишък Максимално Значение. Това означава,че минималната обща информация между признаците е критерии за минимален излишък и максималната обща информация между класове и признаци е критерии за максимално значение. Обратно на това методът базиранна Корелация между признаците взима предвид,че признаците трябва да са силно взаимосвързани с даден клас,но не помежду си.Итеративни(повтарящи се) подходи също са оценявани,когато става въпрос за избор на признаци. Тъй като броят на подмножествата от признаци е О(2n) за n признаци оценяването на всички възможни подмножества е неизчислимо.

***1.2.*Обучение**

В последните години бележитият напредъкът в устройствата за възприятие (камери,акселерометри GPS) улесни процеса на събиране на информация за отделните индивиди и тяхната среда. Обаче повечето приложения изискват много повече от просто измерване на някаква променлива. В същност има допълнителни предизвикателства,които трябва да се преодолеят,за да се постигне достатъчна осведоменост трябва да бъде придобита достатъчна и полезна информация,тъй като само суровият сигнал придобит от сензора е безполезен. За тази цел системите за РЧД разчитат на методите на Машинно обучение,които са полезни за изгражане на образец,които описва,анализа и предсказва. В контекста на машинното обучение образците се откриват от набор от дадени примери или наблюдение на точно определени примери. Такъв входен набор се нарича набор за тренировка. В нашия конкретен случай всеки пример представлява признаков вектор извлечен от суровия сигнал от определен времеви сегмент(прозорец). Примерите в набора за тренировка може да имат,а може и да нямат етикет(label) или още приобщени(асициирани) към определен клас от данни. В някои случаи слагането на етик може да не е осъществимо,защото може да се наложи експерт ръчно да анализира данните и да постави етикет на база опита си. Този процес обикновенно е досаден,скъп и отнемащ време в много от приложенията,събиращи данни.

Съществуват два начина за обучение.Контролирано и не контролирано,които съответно боравят с данни с етикет и такива без. Тъй като след класификацията на човещка дейност трябва да се сложи етикет ходене,тичане и т.н. повечето системи за разпознаване работят с данн с етикет. Всъщност ще е доста трудно да различим дейностите напълно,ако системата ни работи не контролирано . Има такива системи,които работят полу-контролирано,което позволява част от данните да останат без етикет.

1.2.1.Контролирано обучение

Слагането на етик на данни при извършване на дейност е сравнително лесна задача. Някои запазват взетите данни на отделен носител,докато отделен човек се занимава с това да контролира и слага етикет на данните. Други използват приложение,при което дейноста се избира от списък предварително. По този начин се знае всеки отчет за каква дейност се отнася.

Контролираното обучение е област,която е доста добре изследвана и има доста разработени алгоритми в тази насока. В табл.1.4 са обобщение основните класификатори,използвани при разпознаване на дейност.Обяснението за всеки един е както следва:

1.2.2Дърво на решенията-построява йерархичен модел,при който параметрите са разпределени като възли и краищата представляват възможните стойности на параматъра. Всеки клон,излизащ от основата,водещ до възел представлява отделно правило за класификация. С4.5 е може би най-широко използвания класификатор от този тип. Използва ентропията на информацията,за да избере параметрите,които се слагат накрая на възлите. Всяко дърво може да бъде оцененто чрез *O*(log *n*)за n параметрии обикновенно генерират модели,които са лесно разбираеми.На фиг.1.3 е показан пример.

Отговори

Извънредно обаждане

Да

Маркирай като задача и сложи приоритет

Ще отнеме по-малко от 2 мин за отговор

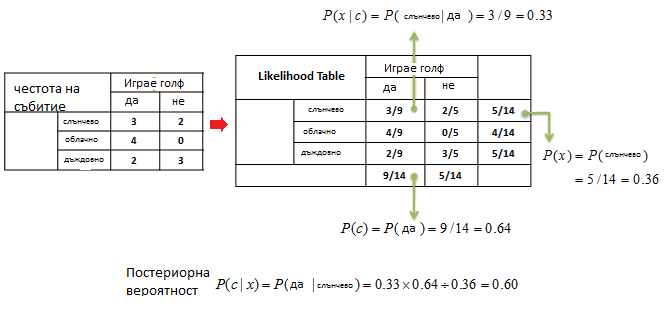
Ще отнеме ли по-малко от 2 мин за отговор

Отговор

Не

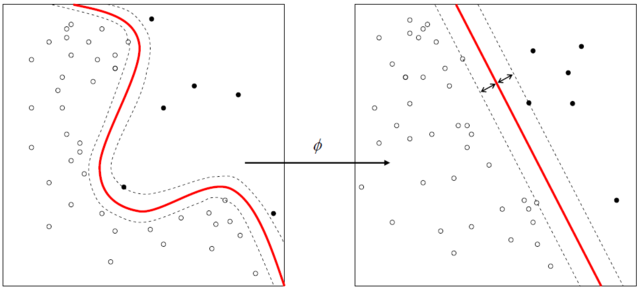
Фиг.1.3 Дърво на решенията

1.2.3Методи на Бейс-изчисляват постериорна вероятност за всеки клас,използвайки очакваната условна вероятност от набора за обучение. Мрежа на Бейс и класификатор на Наийв-Бейс са примери за такива методи. Основен проблем при метода Мрежана Бейс е построяването на топологията,защото е необходимо да се предположи независимоста между признаците. Например класификатор на Наийв-Бейс предполага,че всички признаци са условно независими,но това предположение не издържа в много от случаите. В интересна на истината сигналите от ускорението са с много голяма корелация,както и много от физиологичните сигнали пулс,ЕКГ и др.На фиг.1.4 е даден пример



Фиг.1.4 Методи на Бейс

1.2.4.Инстанция базирани- Това са методи,които класифицират информацията на база подобие на даден пример с набора за обучение. За тази цел се дефинира функция,която отчита разстоянието в подобието между всяка двойка примери. Това прави този вид методи доста скъпи относно време и изчисление,защото класификатора трябва да сравни конкретния пример с целия набор за обучение. Поради тази причина тези методи не се прилагат в мобилни устройства.

 1.2.5.Поддържащи векторни машини и изкуствени невронни мрежи са широко използвани методи при РЧД. ПВМ разчитат на една основна функция,която проектира данните в пространство с по-голяма размерност,което е с цел да се намери линейна граница между класовете(хиперлиния). Невронните мрежи имитират поведението на невроните в човешкия мозък. Невронните мрежи са доказан универсален метод. Дългото време за изчисление и необходимоста от голям брой данни за трениране са два основни проблема на невронните мрежи. На фиг.1.5 е даден пример за Поддържащи векторни машини а)нелинейно разделяне и б)линейно разделяне

а)НелинейниФиг.1.5 Поддържащи векторни машини б)Линейни

1.2.6. Комбинирани методи-комбинацията от различни методи спомага,за да се увеличи точноста при класификация. Очевидно времето за изчисление е проблем,защото е необходимо да се тренират няколко различни модела. В табл.4 са по казани алгоритми за класификация,който се използват в системите за РЧД

Табл.1.4 Алгоритми за класификация използвани в системи за РЧД

|  |  |
| --- | --- |
| **Тип** | **Класификатори** |
| Дърво на решенията | С4.5 ID3 |
| Бейсови | Наийв-Бейс,Мрежи на Бейс |
| Инстанция базирани | k най-близък съсед |
| Невронни мрежи | Перцептрон |
| Променяне на пространстовто | Поддържащи векторни машини |
| Фъзи логика | Фъзи базирани функции |
| Регресивни методи | Многократна линейна регресия |
| Модели на Марков | Скрити модели на Марков |

1.2.6а). Полу-контролирано обучение

Има сравнително малко методи,които са имплементирани за решаване на проблеми свързани с РЧД,които използват полу-контролирано обучение. Тъй като полу-контролирано обучение почти не се използва при РЧД няма стандартни алгоритми и методи,затова се използва различен подход при всеки отделен проблем.

1.2.7.Оценка на измерванията

Попринцип изборът на алгоритъм за класифициране за РЧД се базира на емирични доказателства. Огромното мнозинство от изследвания изпозлват крос-валидация със статистически тестове,за да сравнят класификатори. Резултатите от класификацията за отделен метод могат да бъдат организирани в матрица на грешките *Mn×n*,където n е броят на класовете. Това е матрица,такава че елемент *Mij*е елемент съответстваш на пример от клас i ,който е бил класифициран към клас j .Следните стойности могат да бъдат извадени от матрицата на грешките:

Истинско позитивни(ИП)=броят на положителните примери,които са класифицирани като такива

Истинско негативни(ИН)=броят на негативните примери,които са класифицирани като такива

Фалшиво позитивни(ЛП)=броят на негативните примери,които са класифицирани като позитивни

Фалшиво негативни(ЛН)=броят на позитивните примери,които са класифицирани като негативни

Точноста е стандартна оценка,която резюмира представянето на класификатора

Точност=(ИП+ИН)/(ИП+ИН+ФП+ФН)

Прецизност е отношението на правилно класифицираните примери,към всички примери класифицирани като позитивни.

Прецизност=ИП/(ИП+ФП)

Норма на истински позитивните е отношението между правилно класифицираните положителни примери към всички данни.

Норма=ИП/(ИП+ФН)

Ф измерено е комбинация между прецизност и нормата

Ф измерено=2\*(Прецизност\*Норма)/(Прецизност+Норма)

1.3 Принципи на предаване на информацията от акселерометър.

Приложеното ускорение на аскелерометъра води до разместване на вътрешната маса или огъване на вътрешна греда, причинени от резултатната вътрешна сила. Има няколко вида принципи, които се прилагат за засичане на разместване на вътрешната маса или огъване на вътрешна греда . Често срещаните продавани акселерометри засичат промяна в съпротивлението, капацитет, заряд, температура или опични характеристики. Още съществуват и други методи като електромагнитен принцип, но няма производители, които предлагат акселерометри използващи подобни методи и принципи. Основно акселерометрите се разделят на два типа.Т.нар. АС акселерометри могат да измерват само динамично ускорение, а вторият DC акселерометри, които измерват статично ускорение (земното ускорение) и динамично ускорение.

А. Пиезоелектричен акселерометър с изход заряд

Включва пиезоелктричен материал (естествен кварц), който е свързан с тестова субстанция с позната маса. Ускорението води до преместване на тестовата субстанция и указва натиски на кристака. В следствие, на което се появява пиезоелектричния ефект, който води до преминаване на заряд през кристала. Амплитудата на заряда може да бъде измерена, но със специален нискошумящ кабел. Работят само за АС измервания.

Б. Пиезоелектричен акселерометър с изход напрежение

Работят по абсолютно същия начин както и акселерометрите с изход заряд, само че имат вградена електроника, която усилва сигнала от заряда и го превръща в напрежение.

В. Резистивни акселерометри

Засичат промяна в съпротивлението на метална пластина с калибрирана деформация закачена за вътрешна( конзолна) греда. Ускорението води огъване на вътрешната греда, което води до промяна в съпротивлението на калибрираната пластина. До четири калибрирани пластини са закачени в мост на Вин. Те предоставят сигнал и работят с DC измервания.

Г.Пиезорезистивни акелерометри

Работят по същия начин както резистивните, но огъващатасепластина е изработена от полупроводникови материали. Едно силициево кристалче е 100 пъти по-чувствително на огъване, защото за полупроводниците пиезорезистивния ефект е по-силен, отколкото геометричния. Свързани в мостнаВин осигуряват изходно напрежение, което работи с DC измервания.

Д. Капацитивни акселерометри

Засичатпромени в капацитета на плосък кондензатор. Кондензаторът се състои от местеща се плоскост измежду външните плоскости на кондензатора. Централната плоскост сеотколконява в следствие на ускорението и води до промяна на капацитета. Свързани в мостнаВин осигуряват изходно напрежение, което работи с DC измервания.

Е. Оптични акселерометри

Обикновенно засичат промяната в оптичните характеристики на оптично влакно. Има различен вид техники за подобно измерване. Методанаоптичните решетки Браге доста широко използван метод. Решетките на Браг са интерференчни филтри в оптичното влакно. Решектите отразяват само тесен спектрален компонент на индуцираната светлина. Ускорение води до деформация на оптичното влакно закачено за спирачнагреда. Деформациятанаоптичнотовлакнопроменяхарактеристикитенаотражениенаинтерференчнитерешетки. Тазипромянаможедабъдепроверенакатосесъпоставятспектралнитесъставкинаотразената и индуциранатасветлина. Обикновеннонямавграденаелектрониказапреобразуваненасигнала, затованаизходасеполучаваоптичен сигнал. Работят с DC измервания

Ж. Топлинни акселерометри

Топлинни акселерометри с отместване на субстанцията трудно могат да бъдат намерени произведени масово. Обаче без отместване на субстанцията могат да бъдат открити макар и трудно. Този тип акселерометри от нагревател и термодвойки, които се намират около нагревателя в херметически затворено пространство. Без ускорение нагревателя поражда симетричен профил на температурата. Когато се появи ускорение то измества топлия въздух в затвореното пространство и топлинния профил става асиметричен. Асиметрията бива засичана от термодвойките около нагревателя. Има вградена електроника и изходния сигнал е по напрежение. Работят с DC измервания

**Глава 2**

**Теоретичначаст**

2.1 Информация за базата от данни

Запровеждане на експеримента бе използвана публична база от данни, която може да бъде намерена на адрес: <http://www.cis.fordham.edu/wisdm> . Процедурата за събиране на данните ще бъде описана накратко.

За да бъдат събрани данните за контролирано обучение е необходимо да са от различен брой потребители. В случая са 29 на брой различни потребители(доброволци). Телефонът е поставен в джоба на панталоните и всеки от доброволците изпълнява 6 вида дейности за точно определен период от време. За да бъде сигурно,че събирането на данни ще бъде изпълнено правилно по време на събирането е присъствал човек от екипа учени, които провеждат експеримента. Периода на дискретизация е 20 семпъла в секунда (50ms). Нафигура2.1 е показано къде точно е поставен телефона и съответните направления.



фиг.2.1

В табл.2.1 е дадена подробна информация относно използваната база от данни

|  |  |
| --- | --- |
| Брой на семпълите в базата от данни | 1 098 204 |
| Брой активности | 6 |
| Разпределение на различните дейности в % спрямао общия брой | Ходене -> 424,400 -> 38.6%,  Тичане -> 342,177 -> 31.2%,  Качване по стълбите -> 122,869 -> 11.2%,  Слизане по стълбите -> 100,427 -> 9.1%,  Седене -> 59,939 -> 5.5%,  Стоене -> 48,395 -> 4.4% |
| Формат на даниите | [user],[activity],[timestamp],[x-acceleration],[y-accel],[z-accel]; |
| Пример | 33,Jogging,49105962326000,-0.6946377,12.680544,0.50395286; |
| -Означения на полетата от формата | - Означения на полетата от формата:  [user] - уникален номер на доброволеца 1....36  [activity]- дейност {  Ходене  Тичане Качване по стълбите  Слизане по стълбите  Седебе  Стоене }  [timestamp] - точновременадискретизация |
| Период на дискретизация на сигнала | 50ms |

табл.2.1

2.2 Теоретично описание на избрания алгоритъм

За метод закласификация е избран k най-близъксъсед(kNN)като е малко модифициран.Вместо Евклидово разстояние като критерии за изчесление на средното разстояние е избрано ‘cosine’ . Причините за избиране на този класификатор са няколко.

1. Изключително простата програмна реализация

2. Времето за изпълнение на алгоритъма е доста малко, което е изключително важнo предвид предназначение. Именно мобилен телефон, където ресурсите са ограничени

2.2.1АлгоритъмзакНН

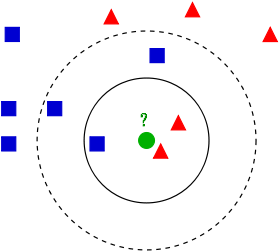
Наборът от данни за тренировка са вектори в многомерното пространство всеки със съответен етикет. Фазата по обучение на алгоритъма се състои от запазване на признаковите вектори и етикетите на класа на тренировъчния набор от данни.

Във фазата по класификация, k константа зададена от потребителя и вектор, който няма етикет (заявен за класифициране вектор или тестова точка) е класифициран по съответния етикет, който е най-често срещан сред набора за обучение, които са най-близо до не-класифицираната точка. Най-често използвана метрика за изчисление на разстоянието между точките е Евклидово разстояние:

За дискретни стойности като класификация на текст е по-подходящо да се използва разстояние по Хеминг

Друга вариация е, че кНН е използвам с коефициенти на корелация като Пиарсън и Спиърман. Често точността на класификация може да бъде значително подобрена, ако метриката за разстояние е 'научена' със специални алгоритми 'анализ на съседните компоненти' и 'голяма разлика за най-близкия съсед'

Малка крачка назад е основната класификация по 'мажоритарно гласуване', която се получава, когато разпределението на класа е изкривено ( с много неправилна форма). Това се проявява,когато на клас, който е по-често срещан е по-вероятно да му се класифицира нов вектор,защото броя на векторите в класа е по-голям. Един от начините да се превъзмогне такъв тип грешка е като се създате теглови коефициент за метриката. Класът ( или стойността за регресивните методи) за всеки най-близък съсед (точка) се умножава по теглови коефициент прорционален на обратното на дистанцията от тази точка до тренировъчната точка. Друг начин да се превъзмогне е като се направи абстракция на представянето на данните. Например в самоорганизиращата се карта всеки възел е представител (център) на клъстър от подобни точки, независимо от тяхната плътност независимо от оригиналния обучителен набор. След това кНН може да бъде приложен към самоорганизиращата се карта. На фиг.2.2 е показано как става решението при кНН класификатора.

.

фиг.2.2

2.2.2Избор на параметри

Най-добре е да изберем параметър k според данните, които имаме в общи линии по-големи стойности на k намаляват шума при класификацията, но правят граници между класовете по-малко разграничими. За да намерим оптималната стойност за параметъра може да се използват и евристични методи методи. Специалният случай, където точка е предсказано да принадлежи към клас он най-близкия обучителен набор ( k=1) се нарича най-близък съсед.

Точността на кНН алгоритъм може да бъде понижена значително, ако присъстват шумящи или незначещи признаци или ако значимостта на признаците не се повишава постоянно със тяхната важност. Много проучвания са направени в посока на това дали да се избират селективни или скалиращи признаци, за да се подобри класификацията. Особено популярен подход е използването на еволюционни алгоритми за оптимизация на скалиране на признаци. Друг популярен подход е да скалираме признаци по общата информация за тренировъчните данни с тренировъчните класове.

Най-интуитивният тип кНН класификатор е този единствен най-близк класификатор, който назначава точка 'х' към най-близкия клас в признаковото пространство, който е

Докато размерът на обучителните данни приближава безкрайност кНН гарантира размерност на грешката не по-зле от два пъти минимално постигнатия размер на грашката при разпределението на данните.

Утежнен класификатор най-близък съсед

К-най-близък съсед класификатор може да бъде разглеждан като се назначи на най-близкия съсед теглови коефициент 1/к , а на всички останали 0. Това може да бъде генерализирано като утежненен най-близък съсед класификатор. Това е, когато на и-тия най-близък съсед има теглови коефициент със . Нека намали тежеста на най-близкия класификатор с коефициент . Като обект на регулярните условия на разпределението на класа излишният риск има следното асимптотично разширение

, за константи B1 и B2 , където и оптималната схема за утежняване това балансира двете условия по-горе и е дадено като

за и за

С оптимални теглови коефициенти доминантното условие в асимптотичното разширение на излишния риск е

2.2.3Свойства

Най-простата версия на алгоритъма е лесна за имплементация като изчислява разстоянието от тренировъчния набор до всички подадени на входа данни. Може да заеме много процесорна мощ при голям брой данни. Използвайки подходящ кНН алгоритъм за търсене може много да намали времето за изчисление. Много кНН алгоритми за търсене са били предлагани през годините. Най-често се използват за намаляване на точките, на които се изчислява разстоянието.

За многокласовия кНН, се доказва, че горната граница на размерността на грешката е , където е големината на грешката на Бейс(минимално възможната грешка) е вероятността за грешка на кНН и М е броя на класовете. За М=2 и грешка на Бейс да се приближава до 0 тази граница се намалява до не повече от два пъти от грешката на Бейс.

2.2.4Големина на грешката

Направени са много изследвания за големината на грешката на кНН класификатора. Класификатора е много постоянен( това е са всяко разпределениие (X,Y)), при условието се разминава, а при условието клони към 0.

2.2.5Извличане на признаци

Когато входните данни на алгоритъм са твърде големи да се обработят, тогава входните данни се трансформират и намали до намален набор от пизнакови вектори. Трансформирането на данни в набор от признаци се нарича извличане на признаци. Ако извлечените признаци са внимателно избрани се очаква, че признаковия набор ще извлече необходимата информация от входните данни, за да извърши желаната задача, използвайки по-малко данни от набора. Извличането на признаци се извърша на суровите данни преди да се приложи кНН алгоритъма.

Пример за типично изчисление в областа на компютърното зрени за разпознаване на лице, използвайки кНН включително извличане на признаци и стпки за пред-процесорно намаляване на размерноста на признаковото пространство(обикновено са имплементирани в OpenCV)

1. Хаар разпознаване на лице
2. Следене на средното изместване.
3. PCA или Fisher LDA проектиране в признаковото пространство последнвано от кНН класификация

Намаляне на размерноста на пространството

За данни с по-голяма размерност на пространството (по-голяма от 10) обикновено то се намалява преди да се приложи кНН алгоритъма, за да се избегнат нежеланите ефекти, които се появяват при големи размерности на пространството. При кНН това означава, че Евклидово разстояние не помага, защото всички вектори са на „евклидово разстояние“ от подадения вектор за класификация (представете си множество точки, които лежат повече или по-малко на един кръг с точка за тестване в центъра на кръга. Разстоянието от тази точка до всички останали ще е приблизително равно)

Извличането на признаци и намаляването на прострранството могат да бъдат компинирани в една стъпка като се използва PCA и LDA или CCA техники като пред-прецосорна обработка, която е последвана от кНН клъстеризатор на признаковите вектори във вече намаленото признаково пространство. В машинното обучение процесът се нарича вграждане на малка размерност.

За много големи размерности ( когато се прилага търсене по сховдство във видео на живо, DNA данни) данни, които използват приблизителен кНН алгоритъм за търсене се използва Locality sensitive hashing(локално разделяне) или някои други подобни алгоритми.

2.2.6 Разширание на кНН за класификация

За разлика ит класическите кНН методи, в които само най-близкия съссед на някой обект се използват, за да се изчисли принадлежноста към дадена група. Разширения кНН метод наречен (ENN) използва дву-посочна комуникация за класификация. Взима под внимание не само най-близките точки, ами и техните най-близки точки. Идеята на този алгоритъм е да причисли група от членове към обект като максимално увеличи вътре-класовата кохерентност, която се измерва статистически измежду всички класове. Емпирични проучвания показват, че новия алгоритъм значително подобрява точноста на класификация сравнен с обикновения метод.

2.2.7 Намаляване на данните

Намаляването на данните е един от най-важните проблеми при работа с голям набор от данни. Обикновено само за част от данните е необходима точна класификация. Тези данни се наричат прототипи и мога да бъдат намерени по следните стъки

1. Намиране на най-отдалечените точки на класа, които се използват за обучителни данни и са класифицирани неправилно
2. Разделяне на тези точки от данните в два набора от данни.Първият прототипите, които са използвани за взимане на решение при класификация. Вторият са точките, които са абсорбирани и могат да бъдат правилно класифицирани,използвайки прототипите. Тези абсорбирани точки могат да бъдат премахнати от обучителния набор.

Избиране на отдалечените точки от класа

Примерни обучителни данни от един, които са заобиколени от примерни обучителни данни от друг клас се наричат външни точки за класа. Причината за присъствието на такива точки включват.

* Произволна грешка
* Недостатъчно обучителни данни за този клас
* Липса на важни признаци
* Твърде голяма размерност на тренировъчния набор за други класове, които пречат на по-малкия клас

Външните точки за класовете са причина за повишаване на шума. Те могат да бъдат намерени и разделени за по-нататъшен анализ.

2.2.8 Модифициран кНН с Харт алгоритъм

На английски (Condensed nearest neighbor, CNN the Hart algorithm) накратко цНН. Е алгоритъм, който е проектиран, за да намали набора от данни за кНН класификация. Избира набор от прототипи У от обучителните данни, такъв че 1НН и У могат да класифицират данните почти толкова точно, колкото 1НН да класифицира целия набор от данни.

При наличен набор от данни Х,цНН работи итеративно

1. Сканира всички елементи на Х, търсейки за елемент х, който е най-близък прототип от У и има различен етикет от х.
2. Премахва х от Х и го прибавя към У
3. Повтаря сканирането, докато няма повече прототипи да се добавят до У

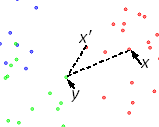
Използва се У, вместо Х за класификация. Примерите, които не са прототипи се наричат „абсорбирани“ точки( фиг.2.3)

C:\Users\pdinev\Desktop\PointsTypes.png

Фиг.2.3

Ефективно е да се сканира обучителния набор от данни по ред на намаляващото отношение на границата. Отношението на границата на обучителните данни х се дефинира по следния начин:

(3)и е показано на фиг.2.4



Фиг.2.4

Където е разстоянието до най-близкия пример от у, който има различен цвят от х, и е дистанцията от у до най-близкия пример , който има същия етикет кат х .

Отношението на границата е в интервала [0;1], защото никога не надвишава . Тази подредба дава предимство на границите на класовете за включване в набора от прототипи на У. Точка с различен етикет от х е външна за х. Етикетите на данните се различават по цвят както е показано на фиг2.4. Първоначалния цвят на точка х е червен. Външните точки са сини и зелени. Най-близката до х външна точка е у . Най-близката до у червена точка е . Отношението на границата е е атрибут на точката х.

**Глава 3**

**Инженерно решение на поставената задача**

На фиг.3.1 е показана блокова диаграма на на точния алгоритъм изплзван за реализиране на инженерната задача

C:\Users\PD\Desktop\Untitled Diagram.png

Фиг.3.1 Алгоритъм на кНН

Следва точни код за реализиране на програмния продукт. Кодът е написан на Matlab v.2013. Подробно обяснение следва след кода.

clc

clear all

close all

tic\_big=tic;

a=dlmread('C:\Users\PD\Desktop\Diplomna magistar\data.csv',',');

train=dlmread('C:\Users\PD\Desktop\Diplomna magistar\Sample.csv',',');

group=importdata('C:\Users\PD\Desktop\Diplomna magistar\class.txt');

labels=importdata('C:\Users\PD\Desktop\Diplomna magistar\labels.txt');

x=a(:,1);

x=2\*(x-min(x))/(max(x)-min(x))-1;

y=a(:,2);

y=2\*(y-min(y))/(max(y)-min(y))-1;

z=a(:,3);

z=2\*(z-min(z))/(max(z)-min(z))-1;

x\_t=train(:,1);

x\_t=2\*(x\_t-min(x\_t))/(max(x\_t)-min(x\_t))-1;

y\_t=train(:,2);

y\_t=2\*(y\_t-min(y\_t))/(max(y\_t)-min(y\_t))-1;

z\_t=train(:,3);

z\_t=2\*(z\_t-min(z\_t))/(max(z\_t)-min(z\_t))-1;

a=[x y z];

train=[x\_t y\_t z\_t];

a=a(1:100:(length(a)),:);

labels=labels(1:100:(length(labels)),:);

x=x(1:100:(length(x)),:);

y=y(1:100:(length(y)),:);

z=z(1:100:(length(z)),:);

error\_x=0;

error\_xy=0;

count\_x=0;

count\_xy=0;

count\_xyz=0;

count\_zy=0;

count\_a=0;

ticid4=tic;

Class\_a = knnclassify(a, train, group,100,'cosine','random');

time4=toc(ticid4)

parfor i= 1:length(x)

error\_a(i)= isequal(labels(i),Class\_a(i));

if error\_a(i)==false

count\_a=count\_a+1;

idx\_a(i,:)=i;

else

idx\_ta(i,:)=i;

end

end

idx\_ta=idx\_ta(idx\_ta~=0);

parfor h=1:length(idx\_ta)

class\_ta(h,:)=labels(idx\_ta(h));

end

check\_1a=class\_ta==1;

check\_1a=check\_1a(check\_1a~=0);

numb\_class1a=length(check\_1a);

check\_2a=class\_ta==2;

check\_2a=check\_2a(check\_2a~=0);

numb\_class2a=length(check\_2a);

check\_3a=class\_ta==3;

check\_3a=check\_3a(check\_3a~=0);

numb\_class3a=length(check\_3a);

check\_4a=class\_ta==4;

check\_4a=check\_4a(check\_4a~=0);

numb\_class4a=length(check\_4a);

check\_5a=class\_ta==5;

check\_5a=check\_5a(check\_5a~=0);

numb\_class5a=length(check\_5a);

check\_6a=class\_ta==6;

check\_6a=check\_6a(check\_6a~=0);

numb\_class6a=length(check\_6a);

ticidx=tic;

Class\_x = knnclassify(x, x\_t, group,100,'cosine','random');

timex=toc(ticidx)

parfor i= 1:length(x)

error\_x(i)= isequal(labels(i),Class\_x(i));

if error\_x(i)==false

count\_x=count\_x+1;

xy(i,:)=[x(i) y(i)];

idx\_x(i,:)=i;

else

idx\_tx(i,:)=i;

end

end

idx\_x=idx\_x(idx\_x~=0);

idx\_tx=idx\_tx(idx\_tx~=0);

parfor i=1:length(idx\_tx)

class\_tx(i,:)=labels(idx\_tx(i));

end

check\_1x=class\_tx==1;

check\_1x=check\_1x(check\_1x~=0);

numb\_class1x=length(check\_1x);

check\_2x=class\_tx==2;

check\_2x=check\_2x(check\_2x~=0);

numb\_class2x=length(check\_2x);

check\_3x=class\_tx==3;

check\_3x=check\_3x(check\_3x~=0);

numb\_class3x=length(check\_3x);

check\_4x=class\_tx==4;

check\_4x=check\_4x(check\_4x~=0);

numb\_class4x=length(check\_4x);

check\_5x=class\_tx==5;

check\_5x=check\_5x(check\_5x~=0);

numb\_class5x=length(check\_5x);

check\_6x=class\_tx==6;

check\_6x=check\_6x(check\_6x~=0);

numb\_class6x=length(check\_6x);

xy\_t=[x\_t y\_t];

parfor i=1:length(idx\_x)

yx(i,:)=xy(idx\_x(i),:);

labels\_xy(i,:)=labels(idx\_x(i));

end

xy=yx;

ticidxy=tic;

Class\_xy=knnclassify(xy, xy\_t, group,100,'cosine','random');

timexy=toc(ticidxy)

parfor k= 1:length(xy)

error\_xy(k)= isequal(labels\_xy(k),Class\_xy(k));

if error\_xy(k)==false

count\_xy=count\_xy+1;

idx\_xy(k,:)=k;

else

idx\_txy(k,:)=k;

end

end

idx\_xy=idx\_xy(idx\_xy~=0);

idx\_txy=idx\_txy(idx\_txy~=0);

parfor k=1:length(idx\_txy)

class\_txy(k,:)=labels(idx\_tx(k));

end

check\_1xy=class\_txy==1;

check\_1xy=check\_1xy(check\_1xy~=0);

numb\_class1xy=length(check\_1xy);

check\_2xy=class\_txy==2;

check\_2xy=check\_2xy(check\_2xy~=0);

numb\_class2xy=length(check\_2xy);

check\_3xy=class\_txy==3;

check\_3xy=check\_3xy(check\_3xy~=0);

numb\_class3xy=length(check\_3xy);

check\_4xy=class\_tx==4;

check\_4xy=check\_4xy(check\_4xy~=0);

numb\_class4xy=length(check\_4xy);

check\_5xy=class\_tx==5;

check\_5xy=check\_5xy(check\_5xy~=0);

numb\_class5xy=length(check\_5xy);

check\_6xy=class\_txy==6;

check\_6xy=check\_6xy(check\_6xy~=0);

numb\_class6xy=length(check\_6xy);

zy\_t=[y\_t z\_t];

parfor q=1:length(idx\_x)

zy(q,:)=[x(idx\_x(q),:) y(idx\_x(q),:)];

labels\_zy(q,:)=labels(idx\_x(q));

end

tic\_z=tic;

Class\_zy=knnclassify(zy, zy\_t, group,100,'cosine','random');

time\_z=toc(tic\_z)

parfor q= 1:length(zy)

error\_zy(q)= isequal(labels\_zy(q),Class\_zy(q));

if error\_zy(q)==false

count\_zy=count\_zy+1;

idx\_zy(q,:)=q;

else

idx\_tzy(q,:)=q;

end

end

idx\_zy=idx\_zy(idx\_zy~=0);

idx\_tzy=idx\_tzy(idx\_tzy~=0);

parfor q=1:length(idx\_tzy)

class\_tzy(q,:)=labels(idx\_tzy(q));

end

check\_1zy=class\_tzy==1;

check\_1zy=check\_1zy(check\_1zy~=0);

numb\_class1zy=length(check\_1zy);

check\_2zy=class\_tzy==2;

check\_2zy=check\_2zy(check\_2zy~=0);

numb\_class2zy=length(check\_2zy);

check\_3zy=class\_tzy==3;

check\_3zy=check\_3zy(check\_3zy~=0);

numb\_class3zy=length(check\_3zy);

check\_4zy=class\_tzy==4;

check\_4zy=check\_4zy(check\_4zy~=0);

numb\_class4zy=length(check\_4zy);

check\_5zy=class\_tzy==5;

check\_5zy=check\_5zy(check\_5zy~=0);

numb\_class5zy=length(check\_5zy);

check\_6zy=class\_tzy==6;

check\_6zy=check\_6zy(check\_6xy~=0);

numb\_class6zy=length(check\_6zy);

parfor l=1:length(idx\_xy)

err\_z(l,:)=z(idx\_xy(l));

err\_x(l,:)=x(idx\_xy(l));

err\_y(l,:)=y(idx\_xy(l));

labels\_xyz(l,:)=labels(idx\_xy(l));

end

err\_xyz=[err\_x err\_y err\_z];

ticid3=tic;

Class\_xyz=knnclassify(err\_xyz, train, group,100,'cosine','random');

time3=toc(ticid3)

parfor p= 1:length(err\_xyz)

error\_xyz(p)= isequal(labels\_xyz(p),Class\_xyz(p));

if error\_xyz(p)==false

count\_xyz=count\_xyz+1;

idx\_xyz(p,:)=p;

else

idx\_txyz(p,:)=p;

end

end

idx\_txyz=idx\_txyz(idx\_txyz~=0);

idx\_xyz=idx\_xyz(idx\_xyz~=0);

parfor l=1:length(idx\_txy)

class\_txyz(l,:)=labels(idx\_txy(l));

end

check\_1xyz=class\_txyz==1;

check\_1xyz=check\_1xyz(check\_1xyz~=0);

numb\_class1xyz=length(check\_1xyz);

check\_2xyz=class\_txyz==2;

check\_2xyz=check\_2xyz(check\_2xyz~=0);

numb\_class2xyz=length(check\_2xyz);

check\_3xyz=class\_txyz==3;

check\_3xyz=check\_3xyz(check\_3xyz~=0);

numb\_class3xyz=length(check\_3xyz);

check\_4xyz=class\_txyz==4;

check\_4xyz=check\_4xyz(check\_4xyz~=0);

numb\_class4xyz=length(check\_4xyz);

check\_5xyz=class\_txyz==5;

check\_5xyz=check\_5xyz(check\_5xyz~=0);

numb\_class5xyz=length(check\_5xyz);

check\_6xyz=class\_txyz==6;

check\_6xyz=check\_6xyz(check\_6xyz~=0);

numb\_class6xyz=length(check\_6xyz);

parfor j=1:length(idx\_txyz)

temp(j,:)=idx\_xy(idx\_txyz(j));

end

temp\_tyz=cat(1,temp,idx\_txy);

parfor k=1:length(temp\_tyz)

temp\_txyz(k,:)=idx\_x(temp\_tyz(k));

end

idx\_ta=cat(1,temp\_txyz,idx\_tx);

parfor l=1:length(idx\_ta)

ta(l,:)=a(idx\_ta(l),:);

labels\_ta(l,:)=labels(idx\_ta(l,:));

end

time\_big=toc(tic\_big)

Обяснение на алгоритъма и кода както следва.

1. Зареждат се файлове с пълната база данни, с пълната база от етикети, тренировъчни данни и етикети за тях.
2. След това трите оси x, y и z се разделят на отделни вектори
3. Нормализиране на всички вектори в интервала [-1;1]
4. Подаване на входа на класификатора матрица, която съдържа едновременно вектори и на трите оси. Защото след това те разделени ще бъдат подадени по отделно на класификатора за класификация, за да може да се сравнят времената и точноста на класификация.
5. След класификацията се намират сгрешените и правилно класифицирани вектори.
6. Т.4 и т.5 се повтарят за всяка от осите със следните промени. Първо се подава пълният вектор на оста „х“ след класификация грешно класифицираните стойности се подават отново на класификатора заедно със съответните им от „у“. Но освен това след класификация на „х“ спрямо „у“ се класифицира допълнително класификация на „у“ спрямо „z“. При „у“ спрямо „z“ стойностите на „у“ са тези съответни на грешните от „х“, а тези на „z“ са спрямо „у“ .
7. Най-накрая се подава последния вектор „xyz“, който представлява грешните вектори след класификация на „xy“ и съответните от „z“

Крайни резултати за време и точност.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вектор/клас | a | x | xy | zy | xyz |
| Тичане | 1102 | 1822 | 530 | 373 | 377 |
| Ходене | 927 | 2407 | 681 | 465 | 404 |
| Нагоре по стълбите | 381 | 0 | 0 | 198 | 149 |
| Надолу по стълбите | 119 | 0 | 0 | 154 | 115 |
| Седене | 133 | 0 | 0 | 126 | 167 |
| Ставане | 0 | 0 | 0 | 0 | 0 |
| Време | 0,5963 с. | 0,2805 | 0,2656 | 0,382 | 0,2306 |
| Процент на разпознаване\* | 24,21% | 38,44% | 17,93% | 19,46% | 21,84% |

\*Забележка процентът на разпознаване е за отношение към съответните входни данни, а не към общите.

Общият процент на разпознаване е – 60,51%

Алгоритъмът на kNNкласификатора ще бъде обяснен както следва

Най-напред се избират параметри за функцията на kNNкласификатора

Class = knnclassify(sample, train,group,100,'cosine','random')

В табл.2.2 са записани параметрети на класификатора и кратко пояснение

|  |  |
| --- | --- |
| Class | Параметър, който представлява класа, към който е класифициран съответния семпъл |
| sample | Матрица, чийто редове ще бъдат класифицирани в групи. Трябва да има същия брой колони като train |
| train | Матрица, която се използва за трениране на алгоритъма. Трябва да има същия брой колони както sample |
| group | Лейбъл( етикет) на тренировъчните вектори |
| 100 | Брой на векторите, които се използват за обучителни |
| 'cosine' | Параметър, който указва вида на разстоянието, което се използва |
| 'random' | Правилото, по което се избира как да се класифицира семпъла |

табл.2.2

Описание на алгоритъма kNN

1. Начално инициализиране спрямо тренировъчния набор от вектори.

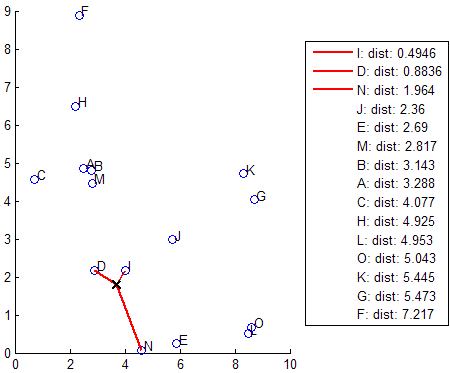
2. Векторите, които се подават за класификация биват класифицирани според правилото за разстояние. В случая не е Евклидово, а 'cosine', което се характеризира със следната формула

(1)

Къдетоса векторите между, които се съпоставят за изчисление на разстоянието.

3. Прилагане на 'правило', от което зависи 'поведението' на класификатора. Избран е параметър 'random' - Основно правило с произволна точка на прекъсване.

На фиг.3.2 е показано нагледно как се пресмята разстоянието до определената точка



Фиг.3.2

**Глава 4**

**Анализ на получените резултати, приложимост и изводи**

В конкретния експеримент бе разгледан алгоритъм кНН( най-близък съсед), който се използва за разпознаване на човешка дейност. Данните бяха класифицирани по два начина с цел да се установи бързодействие и процент на разпознаване. Първият начин е да се класифицират всички данни наведнъж. Вторият е първо да се класифицира едната компонента „х“, след това грешно класифицираните вектори от „х“ се подават наново на класификатора заедно със съответните по „у“ и накрая сгрешените от класификаията на „ху“ се подават наново заедно със „z“ компонентата. Освен това е направена отделна класификация на съответните „уz“ от сгрешените по „х“ компонентата. Т.е. вместо да се класифицира „ху“, класифицираме „zy“. Това е с цел да се провери дали, когато се класифицира по някоя от осите отделно би имало изявена дейност, която се класифицира по-лесно. И резултатът е на лице. По „х“ оста са много ясно изразено разпознаването на дейности тичане и ходене. А по „уz” и „хуz“ ясно се изразява как се разпознават дейност качване по стълбите, слизане по стълбите и седене. Относно времете спрямо процент на разпознаване, когато данните са подадени наведнъж алгоритъмът е по-бърз, отколко на три пъти по отделните оси, но за сметка на това крайно неточен. Но времето,с което се различават е твърде малко що се отнася до това потребителя да е засегнат, защото този брой данни ще се равнява на данните събрани за 549,15с. при период на дискретизация 50мс. т.е. имам 20 семпъла за секунда. Тези данни бяха обработени за общо време от 9,2491с.

За конкретната задача, а именно съставяне на алгоритъм за обработка на данни от акселерометър с цел разпознаване на човекшка дейност. Получените резултати са