Реферат

на теми

„Методи за регистриране на графични елементи в изображенията. Трансформация на Хаф за линейни отсечки. Връзка с трансформация на Радон. Приложения при разпознаване на текст и графика“

и

„Методи за обучение. Обучение с учител - параметрични и непараметрични подходи. Линеен дискриминант на Фишер. Обучение без учител – minimax, kMeans, Izodata“

за курса „Разпознаване на образи“

при доц. д-р Димо Димов

Изготвил:
Теодора Петкова
специалност
„Компютърна графика“
ФН: 23630

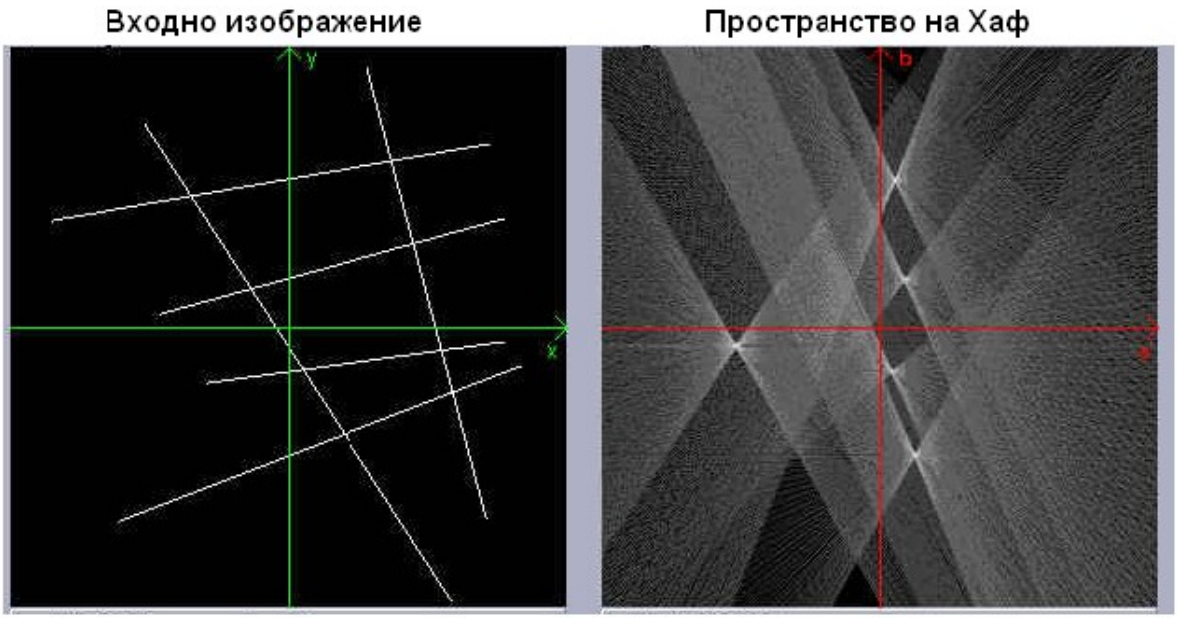
6. Методи за регистриране на графични елементи в изображенията. Трансформация на Хаф за линейни отсечки. Връзка с трансформация на Радон. Приложения при разпознаване на текст и графика

Методи за регистриране на графични елементи в изображенията.

Регистрирането на изображение е процес на препокриване на две или повече изображения от една и съща сцена, направени по различно време, от различни гледни точки или от различни сензори. Геометрично двете изображения се подравняват – референтното(шаблона) и разглежданото. Намират се разликите между двете изображения спрямо различните условия, при които са създадени.
Като цяло приложенията могат да се разделят в четири групи спрямо начина, по който са създадени изображенията:

1. Различни гледни точки (multiview analysis)
   Изображенията от една и съща сцена са заснети от различни гледни точки. Целта е да се получи по-голям 2D или 3D модел на разглежданата сцена.
   Примери за приложения: дистанционни изследвания – разделяне на изображението на изследваната област на подблокове
2. Различно време (multitemporal analysis)
   Изображенията от една и съща сцена са заснети по различно време, често през определено време и при различни условия. Целта е да се намерят и оценят промените в сцената, които се появяват между последователните заснемания на изображението.
   Примери за приложния: дистационни изследвания – мониторинг на използване на обработваемата земя, наблюдение на звездното небе и откриване на нови космически обекти; компютърно зрение – проследяване на движение; медицински изображения – наблюдения на резултатите от терапия на лечение
3. Различни сензори (multimodal analysis)
   Изображенията от една и съща сцена са заснети от различни сензори. Целта е информацията, получена от източници да се интегрира за да се получи по-ясно и детаилно представяне на сцената.
   Примери за приложения: медицински изображения – комбинация от сензори, записващи структурата на тялото като магнитен резонанс или метаболитните процеси в тялото чрез позитронно-емисионната томография (Positron emission tomography) .
4. Регистрация модел към сцена
   Моделът може да бъде компютърно представяне на сцената, например карти или цифрови модели на релефа (DEM - digital elevation models) в ГИС или друга сцена с подобно съдържание. Целта е да се намери заснетото изображение в сцената или да се сравнят.
   Примери за приложения: дистанционни изследвания – регистрация на сателитни данни върху карти или други ГИС слоеве

Трансформация на Хаф (Hough Transform for line-cuts)



Фигура. 1 Трансформация на Хаф – обектно и параметричното пространство

Трансформацията на Хаф е техника, чрез която се извличат характеристиките на форми в дадено изображение. Класическата Хаф трансформация се използва за откриване на правилни криви като линии, кръгове, елипси, т.е. форми, чиито характеристики могат да бъдат описани в параметричен вид. Не е много ефективен при шум и липсващи данни в изображението.

Трансформация на Хаф за отсечка

Примерно трансформацията на Хаф се използва за разпознаване на отсечки. Права се дефинира чрез две точки A = (x_1, y_1) и B = (x_2, y_2). Всички прави, минаващи през точка A, се изразяват с уравнението $y_1 = m*x_1 + b$, където m е наклонът, а b е y – отрязъкът (y – intercept). Аналогично, всички прави през B се изразяват чрез: $y_2 = m*x_2 + b$. Единствената обща точка за двете прави в (b,m) – параметричното пространство е точката, която в (x,y) – пространството на изображението представлява правата, свързваща A и B. Това означава, че правата в изображението е представена чрез единствена точка в (b,m) – параметричното пространство и всяка точка от правата се трансформира в същата точка. Основната идея за откриване на правите е да се определят всички възможни точки от правата в изображението, да се трансформират всички прави, които могат да минат през тези точки като съответстващите им точки в параметричното пространство и да се определят точките (x, y) в параметричното пространство, получени от приложената трансформация на Хаф на линиите от вида $y = m*x + b$ от изображението. Откриването на всички възможни пиксели от правата в изображението може да се постигне чрез прилагането на детектор за ръбове (edge detector) върху изображението. Примерно всички пиксели от ръба с интензитет по-голям от даден праг могат да се считат за възможни пиксели от правата.

В най-общия случай, нищо не се знае за правите от изображението и следователно прави с всякаква посока могат да минават през пикселите на ръбовете. Реално броят на правите е безкраен, но заради практически цели се разглежда само ограничен брой от посоки на

правите. Възможните посоки на правите се определят от дискретизацията на параметъра m. По подобен начин, параметърът b може да приема само ограничен брой от стойности. Така параметричното пространство вече не е непрекъснато и може да бъде представено като правоъгълна структура от клетки. Този масив от клетки се нарича акумулатор A, чийто елементи са акумулаторните клетки A(m, b).

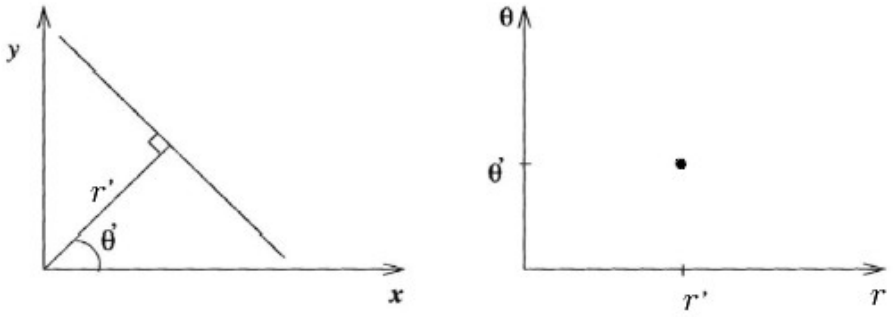
За всеки пиксел от ръб на изображението се определят параметрите m и b, които представляват правите от всички позволени посоки, минаващи през този пискел. За всяка такава права, стойностите на параметрите на правата m и b се използват за да увеличат стойностите на акумулаторната клетка A(m, b). Например, ако правата е представена чрез уравнението y = s*x + t, стойността на акумулаторната клетка A(s, t) ще бъде увеличена толкова пъти колкото правата y = s*x + t е открита като права, която е възможно да преминава през пиксел от ръб на изображението.

За всеки пиксел P, правите, преминаващи през него могат да имат всякаква посока m (от множеството от позволени посоки), но вторият параметър b е ограничен от координатите на пиксела P и посоката m. Следователно съответстващите клетки на правите в изображението ще имат много големи стойности, докато други прави, минаващи през пиксели на ръбове, които не отговарят на съществуващи прави от изображението, имат различни m и b стойности за всеки пиксел от ръб и отговарящите им акумулаторни клетки ще се увеличават рядко. В резултат на това, откриването на права в изображението се превръща в откриването на локални максимуми за пространството от акумулаторните клетки.

Трябва да се отбележи, че трансформацията на Хаф е чувствителна към липсващи части на правите, към шум или други нелинейни структури в изображението. Всъщност параметричното уравнение y = m*x + b е подходящо само за обяснение на принципите на трансформацията на Хаф. То причинява трудности при откриването на вертикални линии, при $m \to \infty$ или нелинейна дискретизация на на параметъра m. По-удобно е представянето чрез уравнението на права с полярни координати:

$$y = -\frac{\cos(\theta)}{\sin(\theta)} * x + \frac{r}{\sin(\theta)} \Rightarrow r = \cos(\theta) + y * \sin(\theta)$$

На изображението по-долу са представени права линия във входното изображение и точката, получена от натрупването на правите (образите на точките от правата на изображението) в (r, θ) - пространството.



Фигура. 2 Права в обектното и параметричното пространство (Хаф)

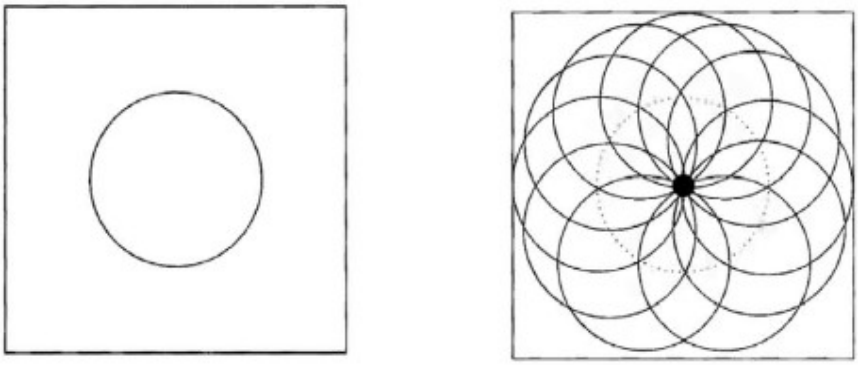
Откриване на крива, използвайки трансформацията на Хаф

Разглеждаме случайна крива, представена чрез уравнението $f(x, a) = 0$, където a е вектор от параметрите на кривата.

1. Ограничаваме на параметричното пространство спрямо параметъра a. Размерността n на пространството се определя от броя параметри на вектора a.
2. Образуваме n – размерен масив за акумулиращите клетки, чиято структура да съответства на размера на параметричното пространство и занулуваме всички елементи.
3. За всяка точка от изображението (x_1, y_1), която е от разглежданите точки в изображението (примерно отговоря на подходящ праг т.е. от обекта, който ни интересува), увеличава акумулиращите клетки A(a): $A(a) = A + \Delta A$
4. Намираме локалните максимуми в акумулирания масив A(a), които отговарят на кривите f(x, a), които са представени в оригиналното изображение.

Трансформация на Хаф за кръг

Ако разглеждаме кръг, аналитичното представяне на кривата ще е:
$(x-a)^2+(y-b)^2=r^2$, където (a, b) е центърът на кръга и r е радиусът. Следователно акумулиращата структура от данни ще бъде триизмерна. За всеки пиксел от ръб на изображението, чийто интензитет надвишава дадения праг, всички акумулаторни клетки, отговарящи на потенциални центрове на кръга, се увеличават в стъпка 3 според алгоритъма. Акумулиращата клетка A(a, b, r) се увеличава, ако точката (a, b) е на разстояние r от разглеждания пиксел.
Ако някой потенциален кръг с център (a, b) и радиус r се среща често в параметричното пространство, е много вероятно кръг с този радиус и център наистина да съществува в обработваното изображение.
Разглеждаме всеки пиксел от окръжността на входното изображение като потенциален център на кръг с радиус r.



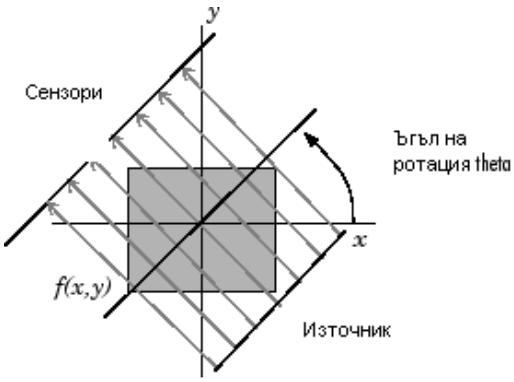
Фигура. 3 Хаф трансформация на кръг

Обобщена трансформация на Хаф
Обобщената трансформация на Хаф се използва, когато формата на характеристиката, която искаме да изолираме не може да се представи с просто аналитично уравнение, представящо границата й. В този случай вместо да използваме параметрично уравнение на кривата, използваме таблица, която дефинира връзките между позицията и ориентацията на границата и параметрите на Хаф. (Създаването на тази таблица се извършва предварително като се използва прототип на желаната форма за идентификация.).

Например нека предположим, че знаем формата и ориентацията на желаната характеристика, което търсим. Можем да определим случайна референтна точка в тази характеристика (x_{Ref}, y_{Ref}) , спрямо която се дефинира и формата на характеристиката (т.е. разстоянието r и ъгълът на нормалните прави, прекарани през границата към тази референтна точка ω). Таблицата (R-table) ще се състои от тези разстояния и двойки, определящи посоката.

Трансформация на Радон (Radon Transform)

Трансформацията на Радон е интегрална трансформация, състояща се от интеграл на функция, дефинирана над множество от прави линии. Трансформацията на Радон е широко приложима в томографията, създаването на изображение от множество разпръснати данни, получено при сканиране на даден обект.
Трансформацията на Радон на множество малки обекти графично се представя като множество синусоидни вълни с различни амплитуди и фази (подобно на трансформацията на Хаф).
Трансформацията на Радон може да се разглежда и като множество проекции на паралелни лъчи върху изображението при ротиране на различни ъгли спрямо центъра.



Фигура. 5 Трансформация на Радон
(представена чрез проекция на паралелни лъчи при ротация)

Нека f е непрекъсната функция. Трансформацията на Радон Rf е функция, дефинирана върху пространство от прави L в $\mathbf{R}^2$ чрез интеграла:
$Rf(L)=\int_L f(x)\,d\sigma(x)$,където $d\sigma$ е дължината на права спрямо L.
Всяка права L може да бъде параметризирана чрез
$(x(t), y(t))=t(\sin\alpha, -\cos\alpha)+s(\cos\alpha, \sin\alpha)$, където s е разстоянието на L от центъра на координатната система и α е ъгълът, който сключва с оста x.
Следователно (α,s) могат да се разглеждат като координати на всички прави в $\mathbf{R}^2$ и трансформацията на Радон може да се изрази с тези координати:

$$Rf(a,s)=\int_{-\infty}^{\infty} f(x(t), y(t))\,dt=\int_{-\infty}^{\infty} f(t(\sin\alpha, -\cos\alpha)+s(\cos\alpha, \sin\alpha))\,dt$$

Може да се разглежда и за n-размерно евклидово пространство $\mathbf{R}^n$.

Приложения при разпознаване на текст и графика

Буквите не могат да бъдат сегментирани чрез просто отделяне с дадена стойност за праг. Цветът, размерът, шрифтът и ориентацията на текста не са известни. Основният проблем е видът на текста, т.е. дали е част от сцената на изображението или имаме текст от документ. Текстът може да се появява и като част от сцена като пътни знаци, табели на улицата и т.н. За разлика от текста от документ, в сцената текстът е част от 3D пространството и следователно може да бъде изкривен от наклона или от формата на обектите, върху които е принтиран. Извличането на текст от естествена сцена е необходимо примерно в проекти като определяне на регистрационния номер на превозно средство.

Сегментация на текстура

Откриването на текст се базира върху подхода за сегментация на текстури. Текстът като част от документ може да се разглежда като област с текстура, която трябва да се отдели. Останалите части от изображението, които не съдържат текст като празни полета, картинки, графики, могат да се считат за области с различни текстури. Човешкото зрение може бързо да разграничи области с текст без да е необходимо да разпознава отделните букви, защото текстът има характеристики, които го различават от останалата част от сцената:

1. Буквите контрастират с техния фон.
2. Текстът има характерна ориентация.
3. Текстът е пространствено свързан: буквите образуват клъстъри на определени разстояния и са подравнени във вертикална линия.

11. Методи за обучение. Обучение с учител - параметрични и непараметрични подходи. Линеен дискриминант на Фишер. Обучение без учител – minimax, kMeans, Izodata

Методи за машинно обучение (Machine learning methods)

Машинното обучение е направление в изкуствения интелект, което се занимава с алгоритми и методи за автоматичното научаване на системи от правила, признаци и характеристики, чрез които могат да взимат решения за класификация на данните.

Всеки процес на обучение се базира на множество от данни, от които програмата се обучава. При това има два типа обучение — със специално обучаващи примери, където е указано какво решение да се вземе за всеки пример (Supervised Machine Learning) или без изричното задаване на решенията при обучаващите примери (Unsupervised Machine Learning).

Обучение с учител (Supervised Machine Learning)

Обучението с учител е техника за научаване чрез извличане на целева функция на базата на примерни данни (training data). Данните се състоят от двойки от входни обекти (обикновено вектори) и желани резултати. Изходът от функцията може да бъде непрекъсната стойност (регресия) или може да определя класа, към който принадлежи входния обект (класификация). Задачата на методите за обучение с учител е да предскажат стойността на функцията за всеки валиден входен обект, базирани на примерни опитни данни (двойки от вход и желан изход). За да се постигне това, е необходимо да се обобщят представените данни до ненаблюдавани ситуации индуктивно.

Параметрични подходи

Наивен класификатор на Бейс (Naïve Bayes)
Наивният класификатор на Бейс се прилага при задачи, където всеки елемент x се описва от набор от атрибути и целевата функция f(x) може да приема всяка стойност от крайно множество C. Имаме множество от пробни данни за целевата функция на базата, на които трябва да се определи целевата стойност или клас за новия елемент. Всеки елемент е представен чрез вектор от стойности на атрибути (x_1, x_2...x_n).

Теорема на Бейс
Нека X е екземпляр от данни. Нека H е хипотеза, че X принадлежи към клас C.
Класификационната задача изисква да се определи P(H|X), т.е. вероятността да се реализира хипотеза H, при условие, че се наблюдава екземпляра X
P(H) - (априорна вероятност), начална вероятност
P(X): вероятност да се наблюдава екземпляр X
P(X|H) (постериорна вероятност), вероятността да се наблюдава X , при условие, че е реализирана хипотезата H

Съгласно теоремата на Бейс е: $$P(H|X)=\frac{P(X|H)P(H)}{P(X)}$$

Идея на Бейсовата класификация

Наивният класификатор на Бейс е прост вероятностен класификатор, при който се прилага теоремата на Бейс със строги (наивни) предположения за независимостта на характеристиките на елементите т.е. наличието или отсъствието на дадена характеристика за даден клас не е свързана с наличието или отсъствието на друга характеристика.

При класифициране на нов елемент чрез подхода на Бейс определяме най-вероятната целева стойност чрез V_{MAP} (MAP = maximum a posteriori probability estimate) при дадени стойности на атрибутите (x_1, x_2 ...x_n), които описват елемента.

$$V_{MAP} = argmax\, P(C_j | x_1, x_2, ..., x_n)$$

Пренаписваме формулата използвайки теоремата на Бейс:

$$V_{MAP} = \underset{C_j \in C}{argmax}\, P(C_j | x_1, x_2, ..., x_n) = \underset{C_j \in C}{argmax} \frac{P(x_1, x_2, ..., x_n | C_j)}{P(x_1, x_2, ..., x_n)}$$

Тъй като P(X) е константа за всички класове и максимизираме, можем да използваме формулата:

$$V_{MAP} = \underset{C_j \in C}{argmax}\, P(x_1, x_2, ..., x_n | C_j)\, P(C_j)$$

Класификаторът на Бейс е основан на твърдението, че стойностите на атрибутите са условно независими. С други думи, при дадена целева стойност на елемент, вероятността за наблюдение на набора от атрибути (x_1, x_2 ...x_n) е просто произведението от вероятностите на отделните атрибути т.е. $P(x_1, x_2, ..., x_n | C_j) = \prod_i P(x_i | C_j)$.

Оттук получаваме формулата за наивния класификатор на Бейс:

$$V_{MAP} = \underset{C_j \in C}{argmax}\, P(C_j) \prod_i P(x_i | C_j)$$

Ако дадена стойност никога не се среща в пробното множество от данни, тогава вероятността, с която ще бъде оценена ще е нула, но така се губи информация. В този случай е добре да се добави още едно правило, никоя вероятност да не се задава като нулева.

Непараметрични подходи

K най-близки съседи (K nearest neighbours)

Алгоритъмът за к най-близки съседи (k-NN) е метод за класификация на елементи, базиран на най-близките пробни данни в пространството от елементи, което разглеждаме. Предполагаме, че всички елементи, съответстват на n-размерно пространство Rn. Най-близките съседи за даден елемент са дефинирани спрямо стандартното Евклидово разстояние. Нека елемент x се задава чрез вектор от неговите характеристики:

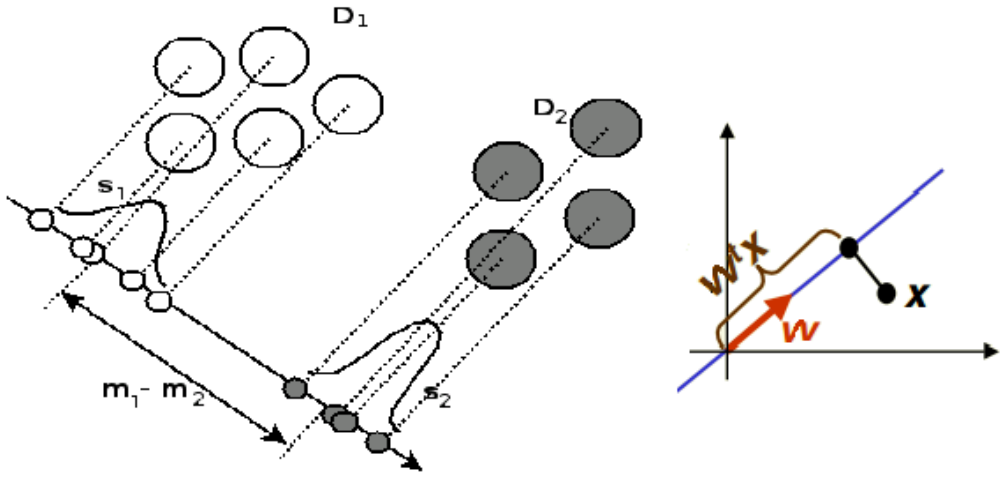
$a_1(x), a_2(x), ..., a_n(x)$

ar(x) е r-тият атрибут на елемента x. Следователно разстоянието между два елемента xi и xj се определя като d(xi, xj), където $d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2}$

След като изберем стойност за k, можем да правим предположения за примерите за kNN. Избираме класа за неизвестния елемент спрямо класовете, към които принадлежат най-много от съседите му (k на брой). Следователно за класификационни задачи с два класа, вземаме нечетни стойности за k, за да избегнем равенство. При регресия, където определяме зависимите променливи спрямо независимите (имаме непрекъсната функция) вземаме средната стойност от известните данни за k съседите: $f(x) = \frac{1}{k} \sum_{i=1}^{k} x_i$

Линеен дискриминант на Фишер (Fisher's linear discriminant)

Линеен дискриминант на Фишер е метод за класификация, който проектира данни от голяма размерност (примерно d-размерност) върху права и извършва класификацията в това едноразмерно пространство. Всъщност, дори ако елементите формират добре разпределени и компактни клъстъри в d-размерното пространство, при проекция върху случайна права обикновено се получава смес от елементи от всички класове и следователно това би довело до трудно разпознаване. Целта е да се намери подходящата ориентация на правата, така че проектираните елементи да са подходящо групирани т.е. добре разделени.
Нека имаме множество от n d-размерни елементи $x_1, ..., x_n$, n_1 е броят елементи на единия клас D_1, а n_2 е броят елементи на другия клас D_2.



Фигура. 5 Линеен дискриминат на Фишер
(проектирането на данните върху права)

Нека проектираме елемент x към скаларна стойност y, която отговаря на вектора w, лежащ върху проектираната права. Скаларът $y = w^t \cdot x$ е разстоянието от проекцията на x от центъра на координатната ос т.е. е проекцията на x в едноразмерното пространство.

Така получаваме съответстващо множество от n елемента $y_1, ..., y_n$, разделени в подмножества Y_1 и Y_2. Целта е да определим степента на разделение между различните класове. Мярка за разделението между проектираните точки е разликата между очакванията на проекциите. Следователно $\mu_i = \frac{1}{n} \sum_{x \in D_i} x$ е очакването за класа D_i, i=0,1 и $\tilde{\mu}_i = \frac{1}{n} \sum_{y \in Y_i} y$ е очакването за класа Y_i, i=0,1.
Разстоянието между очакванията на проекциите на класовете се изразява чрез:
$|\tilde{\mu}_1 - \tilde{\mu}_2| = |w^t(\mu_1 - \mu_2)|$

Можем да увеличаваме разликата колкото желаем само като скалираме вектора w.
Освен очакванията, като фактор при разделянето на класовете трябва да се отчете и отклонението т.е. разсейването на елементите след проекцията: $\tilde{s}_i^2 = \sum_{y \in Y_i} (y - \tilde{\mu}_i)^2$

Оттук получаваме линейния дискриминант на Фишер за проектиране върху права в посока на

$$J(w)=\frac{|\tilde{\mu}_1-\tilde{\mu}_2|^2}{\tilde{s}_1^{\,2}+\tilde{s}_2^{\,2}}$$

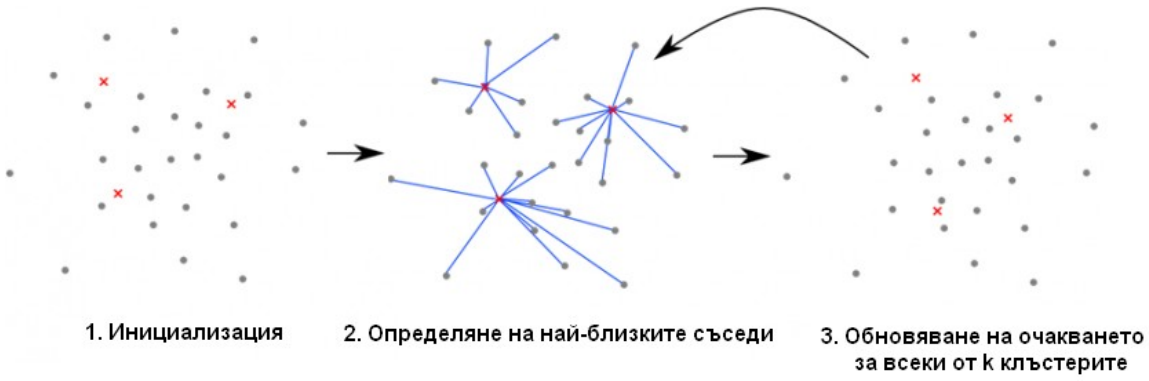
вектора w:

Когато максимизираме w, това води до по-добре разделение на двете проектирани множества, т.е. искаме разстоянията между очакванията на проектираните класове да е голямо и разсейванията (отклоненията) за проектираните класове да са малки.

Обучение без учител (Unsupervised learning (self-learning))

При обучение без учител се опитваме да определим как е организирана информацията. Основната цел е да се търси подобие между елементите за да се определи как могат да бъдат характеризирани, така че да се оформят в групи (клъстъри).

Методът на k средни(k-means)



Фигура. 6 Стъпките при метода на k-means

Това е метод от клъстърния анализ, който разделя n наблюдения в k клъстъра като всяко наблюдение принадлежи на клъстъра с най-близкото очакване (т.е. разделяне на пространството от данни на клетки на Вороной).

Целта е да класифицираме множеството от данни спрямо дадения брой клъстери k, фиксиран предварително като определим k центъра, по един за всеки клъстър $S=\{S_1, S_2, \ldots, S_к\}$. Всъщност, трябва да се минимизира целевата функция – вътреклъстърната сума на квадратите (within-cluster sum of squares (WCSS)):

$$J=\underset{S}{argmin}\sum_{i=1}^{k}\sum_{x_j\in S_i}\|x_j-\mu_i\|^2 \quad ,$$

където μ_i е очакването на S_i (или може да се разглежда разстоянията между точките, характеризиращи данни от техните съответстващи клъстърни центрове).

Алгоритъм:

1) Избираме k начални стойности произволно от множеството данни или на базата на някаква предварителна евристика
2) Създаваме k клъстъра спрямо най-близките стойности за очакването (диаграми на Вороной спрямо очакването)
   $S_i=x_t:\|x_t-\mu_i\|\leqslant\|x_t-\mu_j\|\ for\ all\ j=1,\ldots,k$
   като x_t отива точно в един клъстър S_i
3) Обновяване на очакването за всеки от k клъстърите:
   $$\mu_i=\frac{1}{|S_i|}\sum_{x_t\in S_i}x_t$$

4) Стъпки 2 и 3 се повтарят до достигане на сходимост т.е. докато престанат да се обновяват стойностите за очакването

Алгоритъмът не винаги намира най-доброто оптимално решение, съответстващо на глобалния минимум за целевата функция, тъй като зависи от първоначално избраните клъстърни центрове. Възможно решение е да се изпълни няколко пъти, за да се намали този ефект.

ISODATA (Iterative Self-Organizing Data Analysis Technique)

ISODATA е алгоритъм, който е разширение на метода за k средни чрез евристики за автоматично избиране на броя клъстъри.
Алгоритъмът отново е итеративен:

1) Прилагаме метода на k средните
2) Разделяме клъстърите, чиито елементи са достатъчно нееднакви
3) Обединяваме клъстъри, които са достатъчно близки
4) Продължаваме със стъпка 1) до сходимост т.е. класификацията на елементите не се е променила

Клъстърите се обединяват, ако броят на елементите в клъстърите е по-малък от дадения праг или центровете на двата клъстъра са по-близки от дадения праг и съответно се разделят, ако стандартното отклонение надвишава предопределена стойност и броят на елементите е два пъти по-малък от дадения праг.

Целевата функция на алгоритъма за k средни е сумата от квадратите на разстоянията между всеки елемент и центъра на неговия клъстър:

$$SS_{distances} = \sum_{\forall x} [x - C(x)]^2 \quad .$$

При ISODATA тази сума се минимизира Mean Squared Error (MSE):

$$MSE = \frac{\sum_{\forall x} [x - C(x)]^2}{(N-c)b} = \frac{SS_{distances}}{(N-c)b} \quad ,$$

където N е броят елементи, c – броят клъстъри

На практика ISODATA трябва да се пусне много пъти с различни стойности на параметрите, за да се определи разделянето на клъстъри с най-малка грешка на минималната сума на квадратите.