Journal of AL-Qadisiyah for computer science and mathematics Vol.10 No.2 Year 2018
ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

Linear programming measures for solving the problem of a linear division of convex multiplayers.

Sarmad H. Ali Osamah A. Ali Samir C. Ajmi

Electronic Computer Center , Al-Muthanna University

sarmad@mu.edu.iq osama@mu.edu.iq Samir@mu.edu.iq

Recived : 11\2\2018 Revised : 29\3\2018 Accepted : 2\4\2018

Available online : 10 /6/2018

DOI: 10.29304/jqcm.2018.10.2.395

Abstraction

In this research, we are trying to solve Simplex methods which are used for successively improving solution and finding the optimal solution, by using different types of methods Linear, the concept of linear separation is widely used in the study of machine learning, through this study we will find the optimal method to solve by comparing the time consumed by both Quadric and Fisher methods.

Keyword

simplex method, linear programming problem, Quadratic simplex method, separating hyperplane

Introduction

The creation of computers and the associated accelerated development of mathematical theories, including mathematical cybernetics and discrete mathematics, as well as information technologies, made it possible to set and solve on the computer new tasks that until recently were exclusively within the competence of man. One of such fundamental problems is the solution of multidimensional problems of linear separation of convex polyhedra.

The concept of linear separation is widely used in the study of machine learning. It is also used for the preparation of support vector methods, by means of which problems of complex systems are solved. In this study, some methods for testing the linear separation between two classes of data will be presented[10].

In general form, this problem can be formulated as follows: two subsets (two classes) X and Y of R ^ d are considered. They are linearly separable if there exists a hyperplane P of R ^ d and if the elements of X and Y lie on opposite sides of each other.

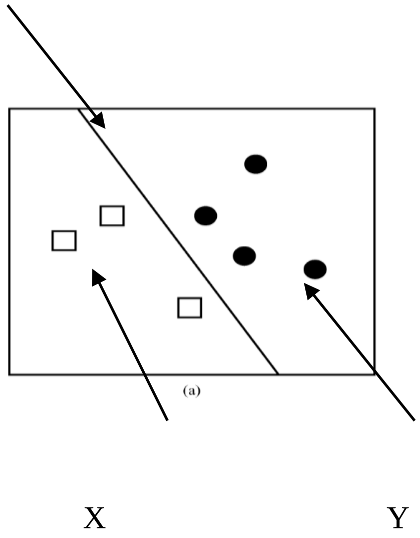
i.e.

$$P\ (w,\gamma)\ =\ \{z \in\ R^d : < W,z > =\ \gamma\}.$$

$$< W,x > \leq \gamma \quad \forall\, x \in X.$$

$$< W,y > \leq \gamma \quad \forall\, y \in Y.$$

P (w, γ)



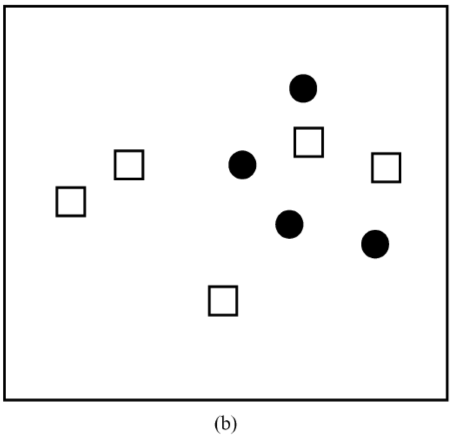


figure.1.

Squares and circles represent the two classes. (X and Y)

In Figure 1 (a) is represented by the linearly separable case, as in Figure 1 (b) is not separable case.

Purpose and objectives of the study

The purpose of this study is to develop a software package that numerically realizes and compare the solution of the problem of linear separation of convex polyhedral by various computational means:
1) By solving the linear programming problem by the simplex method;
2) By reducing to the problem of quadratic programming;
3) Through the application of the Fisher's method.
To achieve this goal, it was necessary to solve the following tasks:
-describe a general approach to solving the problem of linear separability of two classes on the basis of the listed methods;
- Design and implement a software package for solving the separation problem, using the developed methods and algorithms. Conduct computational experiments to analyze the effectiveness of the proposed approaches.

Theoretical and practical value

The theoretical value of the study is that it gives a formal description of methods for solving the separation problem.
The practical value of the study lies in the fact that the proposed methods are implemented in the form of a software complex for the computer, which allows to effectively solve the problems of linear separability, in particular, used for dimension equal to 2, for the implementation of visual graphical interpretation.

In conclusion, summarizes the main results of the study, brief conclusions have presented the work.

1.1. The solution of the linear programming problem, Simplex - method.

When solving problems of linear programming, one of the most popular methods is the simplex method. The simplex method is a method of successively improving the solution and finding the optimal solution[1]. at the beginning if there is any admissible basic solution corresponding to one of the corner points of the solution polyhedron, and then this solution is purposefully improved by moving to a new basic solution at the neighboring corner point at which the value of the objective function does not decrease to the maximum , until an optimal solution is obtained. This method is universal, with the help of which it is possible to solve any linear programming problem.[2]
The simplex method is intended to solve linear programming problems in canonical form. Depending on the nature of the limits, linear programming problems can be solved with a natural and artificial basis.
If the limits are given by inequalities of the type "≤", then the problem is solved with a natural basis if the limits are given by inequalities of the type "≥" or the equalities "=", then the solution is conducted with an artificial basis.
There are three ways to solve the task :

1) The model is solvable.

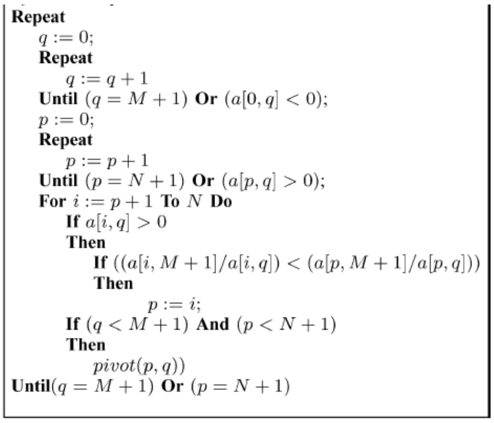
2) The model is inadmissible.

3) The model is unlimited.

In this study, we use the simplex method to solve the problems of linear separability between two classes. Algorithms for solving the objective function are presented in Tables I and II. The first algorithm makes it possible to find the values of **q and p**, shows the obtaining of the optimal solution, or vice versa, shows that the problem does not have a solution

Table I. Algorithm of the objective function

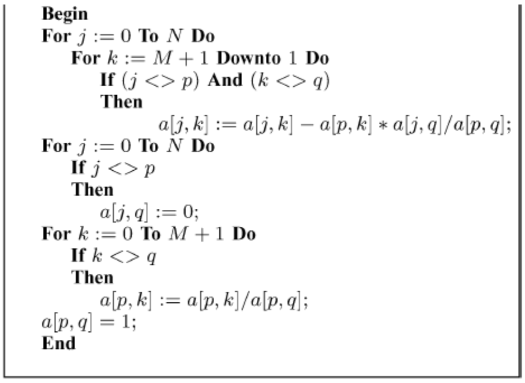
- Data: an array A of size M × N is given, containing the constraints of linear expressions, where the rotation column and rows are represented as **p and q**.
- Result: the solution is, if the limits of the linear expression are solvable, and otherwise there is no solution.



```
Repeat
    $q := 0;$
    Repeat
        $q := q + 1$
    Until $(q = M + 1)$ Or $(a[0, q] < 0);$
    $p := 0;$
    Repeat
        $p := p + 1$
    Until $(p = N + 1)$ Or $(a[p, q] > 0);$
    For $i := p + 1$ To $N$ Do
        If $a[i, q] > 0$
        Then
            If $((a[i, M + 1]/a[i, q]) < (a[p, M + 1]/a[p, q]))$
            Then
                $p := i;$
        If $(q < M + 1)$ And $(p < N + 1)$
        Then
            $pivot(p, q))$
Until$(q = M + 1)$ Or $(p = N + 1)$
```

Table II. Procedure of turning (p, q)

Data: the string p and the column q of the array M × N are given, containing the linear expression constraints that are used to perform the rotation



```
Begin
For $j := 0$ To $N$ Do
    For $k := M + 1$ Downto 1 Do
        If $(j <> p)$ And $(k <> q)$
        Then
            $a[j, k] := a[j, k] - a[p, k] * a[j, q]/a[p, q];$
For $j := 0$ To $N$ Do
    If $j <> p$
    Then
        $a[j, q] := 0;$
For $k := 0$ To $M + 1$ Do
    If $k <> q$
    Then
        $a[p, k] := a[p, k]/a[p, q];$
$a[p, q] = 1;$
End
```

These algorithms can be used to solve the jective function.
We illustrate the simplex method, showing Linear programming through some task Z[7].
Task Z: Let X = {(1,1)} and $Y = \{(0,0), (1,0), (0,1)\}$ be the initial data for two classes that define the problem Z. We check $X \mid\mid Y$?

Adj: is the operation of adding a value, in this case, we add *1* if the point belongs to class *1*, otherwise *-1* and the coordinates of points from *2* classes change the sign.$S =$ $Adj(X,1) \cup$ $-Adj(Y,1)\{1,1,1),(0,0,-1),(0,-1,-1),(-1,0,-1)\}$. So, to find out X || Y, we need to find a set of values for weights w_1 $+w_1$+t, and threshold *t*, so that it minimizes one of the indicated inequalities.

$$\text{On Condition} \begin{cases} w_1 + w_2 + \mathrm{t} > 0 \\ -t > 0 \\ -w_2 - t > 0 \\ -w_1 - t > 0 \end{cases} \qquad (1.1)$$

Since the Simplex method limits the values of variables that are ≥ 0, and the weight value can be either positive or negative, we transform each of our original variables as a difference of two variables. That is, we reduce it to the canonical form:

$$\begin{cases} w_1 = w_{11\_aux} - w_{12\_aux} \\ w_2 = w_{21\_aux} - w_{22\_aux} \\ t = t_{1\_aux} - t_{2\_aux} \\ \end{cases} \qquad (1.2)$$

Since the simplex method does not take a strict inequality, we change our sign > 0 by ≥ 1.

Using the above transformations, our new set of constraints becomes:

$$\begin{cases} \left(w_{11_{aux}} - w_{12_{aux}}\right) + \left(w_{21_{aux}} - w_{22_{aux}}\right) + \left(t_{1_{aux}} - t_{2_{aux}}\right) \geq 1 \\ \left(-t_{1_{aux}} - t_{2_{aux}}\right) \geq 1 \\ \left(- w_{11_{aux}} + w_{12_{aux}}\right) - (t_{1_{aux}} + t_{2\_aux}) \geq 1 \\ \left(- w_{21_{aux}} - w_{22_{aux}}\right) - (t_{1_{aux}} + t_{2\_aux}) \geq 1 \\ \end{cases}$$

Applying the simplex method next, we obtain an admissible solution which gives the following result:

$$\begin{aligned} w_{11_{aux}} &= 2 \quad w_{12_{aux}} = 0 \\ w_{21_{aux}} &= 2 \quad w_{22_{aux}} = 0 \\ t_{1_{aux}} &= 0 \quad t_{2\_aux} = 3 \end{aligned}$$

[8] Thus, we can conclude that task Z is solved. Now we can define the values for the variables: $w_1 = 2, w_2 = 2$ t= –3. These variables form a hyperplane that separates the two classes.

The disadvantage of the Simplex method is that the best solution is not guaranteed in advance, it is impossible to say how many steps it takes to achieve the optimal solution. The complexity of this method depends on the rotation rule used, which was indicated in Table II.

1.2. The solution of the quadratic programming problem

The process of programming this method consists of solving Quadratic programming. It occurs in the use of the support vector method (SVM). The support vector method is a set of algorithms used for classification problems and regression analysis. Belongs to the family of linear classifiers. A special property of the support vector method is a continuous decrease in the empirical classification error and an increase in the gap, so the method is also known as the classifier method with the maximum gap. The main idea of the method is the translation of the initial vectors into a space of higher dimension and the search for a separating hyperplane with the maximum gap in this space. Two parallel hyperplanes are constructed on both sides of the hyperplane that separates our classes. The separating hyperplane is a hyperplane that maximizes the distance to two parallel hyperplanes. The algorithm works under the assumption that the greater the difference or the distance between these parallel hyperplanes, the smaller will be the average classifier error.

The hyperplane separating the two classes is defined as follows:

$$\begin{cases} \overrightarrow{w^T}\overrightarrow{x_\iota} + \mathrm{b} \geq 1 \ \ c_i = 1 \\ \\ \overrightarrow{w^T}\overrightarrow{x_\iota} + \mathrm{b} \leq 1 \quad c_i = -1 \end{cases}$$

(1.4)

Initial data from us $\vec{x}_i \in R^n, i = 1, \ldots, l$ and $c_i, c \in$ $\{-1,1\}$,

We find the inverse matrix by the Gauss method, for the convenience of the definition we denote $S_w = A \; and \; S_w^{-1} = X$, then we use the equation A*X=E, where: X is the inverse matrix of the matrix A, and E is the identity matrix,

We reduce the matrix A to triangular matrix using the Gauss algorithm“Direct motion”.

In parallel, we perform similar operations with the unit matrix E.

We finally represent the obtained equation in its expanded form (fig 1.2)

The method of support vectors should find a solution to the minimization problem: $\frac{1}{2}\overrightarrow{||w||^2}$

On Condition $c_i(\overrightarrow{w^T}\overrightarrow{x_i} + \mathrm{b}) \geq 1$ Where C> 0 represents the penalty parameter.

We solve the following minimization problem: $\frac{1}{2}\overrightarrow{||w||^2}$ (1.5)

Formal description of the task:

We believe that the points have the form:

$$\{(x_1, c_1), (x_2, c_2), \dots, (x_n, c_n)\}$$

Where $c_i$takes the values 0 or 1, depending on which class the point $x_i$belongs to. Each $x_i$is a p-dimensional real vector, usually normalized by the values [0,1]. If the points are not normalized, the point with large deviations from the average coordinates of the points will affect the classifier too much. We can treat this as a learning collection, in which the class to which it belongs is already assigned to each element. We want the algorithm of the support vector method to classify them in the same way. To do this, we construct a separating hyperplane that looks like this:

w*x-b=0; (1.6)

Where w, b - are unknown, they need to be found.

The vector w is the perpendicular to the separating plane (hyperplane);

W=$(w_1, w_2, \dots, w_n)$;

$\frac{b}{||w||}$- the distance from the hyperplane to the origin;

If the parameter b is zero, the hyperplane passes

through the origin, which limits the solution.

$||w|| = \sqrt{\sum_{i=1}^n w_i^2}$ -Is the length of the vector (1.7)

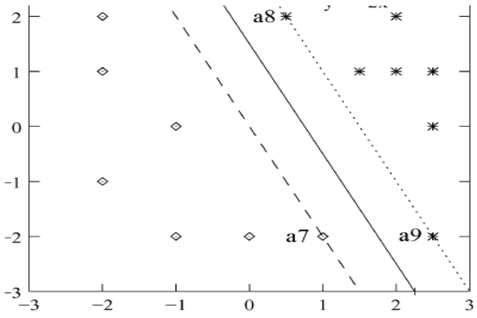
Since we are interested in the optimal separation, we are interested in support vectors and hyperplanes parallel to the optimal and closest to the supporting vectors of two classes. It can be shown that these parallel hyperplanes can be described by the following equations (up to normalization).

w*x-b=1, (1.8)

w*x-b=−1 . (1.9)

If the training sample is linearly separable, then we can choose hyperplanes in such a way that no points of the training sample lie between them and then maximize the distance

between the hyperplanes. The width of the strip between them is easy to find from considerations of geometry, it is $\frac{2}{||\mathrm{w}||}$, so our task is to minimize $||\mathbf{w}||$.



$$\frac{2}{||\mathbf{w}||}$$

To exclude all points from the strip, we must verify for all *i* that

$$\begin{cases} \mathrm{w} * \mathrm{x_i} - \mathrm{b} \leq -1 , c_i = -1 \\ \mathrm{w} * \mathrm{x_i} - \mathrm{b} \geq 1 , c_i = 1 \end{cases}$$

(1.10)

This can also be written in the form:

$c_i(w * x_i - b) \geq 1; \quad 1 \leq i \leq n$ (1.11)

The problem of constructing an optimal separating hyperplane reduces to minimizing || w || , Under the condition (1.11). This is the problem of quadratic optimization, which has the form:

$$\begin{cases} \boldsymbol{||w^2|| \to min} \\ c_i(w * x_i - b) \geq 1; \quad 1 \leq i \leq n \end{cases} \quad (1.12)$$

By the Kuhn-Tucker theorem (necessary conditions for solving the problem of non-linear programming), this problem is equivalent to the dual problem of finding the saddle point of the Lagrange function.

$$\begin{cases} L(w, b, h) = \frac{1}{2} \boldsymbol{||w^2||} \\ \boldsymbol{- \sum_{i=1}^n h_i(c_i((w, x_i) - b) - 1 \to min_{w,b} max_\lambda} \\ \boldsymbol{\lambda_i \geq 0, 1 \leq i \leq n} \end{cases}$$

(1.13)

$\lambda = (\lambda_{1,\dots,}\lambda_n)$ a vector of dual variables.

We reduce this problem to an equivalent quadratic programming problem, which contains only dual variables:

$$\begin{cases} -L(\boldsymbol{\lambda}) = -\sum_{i=1}^n \boldsymbol{\lambda_i} + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \boldsymbol{\lambda_i \lambda_j c_i c_{j_i}} (\boldsymbol{x_i x_j}) \to \boldsymbol{min\lambda_i} \\ \boldsymbol{\lambda_i \geq 0, 1 \leq i \leq n} \\ \sum_{i=1}^n \boldsymbol{\lambda_i c_i = 0} \end{cases}$$

(1.14)

All variables are λ
We single out the linear and quadratic parts as follows:

$$-L(\boldsymbol{\lambda}) = -\underbrace{\sum_{i=1}^{n} \boldsymbol{\lambda}_i}_{\text{CX}} + \frac{1}{2}\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{\lambda}_i\boldsymbol{\lambda}_j\, \boldsymbol{c}_i\boldsymbol{c}_j\, (\boldsymbol{x}_i\boldsymbol{x}_j) \to \boldsymbol{min\lambda}_i}_{\substack{i \\ \boldsymbol{X}^{\boldsymbol{T}}\text{DX}}}$$

(1.15)

Next, we make a system check (1.10). Namely, a quadratic function having a nonnegative definite quadratic form is minimized, hence, the objective function is convex. An admissible region is a polyhedron, which means that we are faced with the problem of quadratic programming.

Hence we can apply the *Kuhn-Tucker* optimality theorem.

We have a task: $\mathbf{CX} + \frac{1}{2}\boldsymbol{X}^{\boldsymbol{T}}\mathbf{DX} \to \boldsymbol{min}$(1.16)
Under the constraints: AX$\leq b, X \geq 0$,

Next, we apply the *Kuhn-Tucker* theorem. The convexity conditions of the objective function and the admissible domain are allowed to use this theorem.
We record the *Kuhn-Tucker* system:

$$\begin{cases} \mathrm{Ax} + \ Y = b \\ -Dx - A^{\{T\}U} + V = C \\ V^{\{T\}x} = 0,\ \ U^{\{T\}Y} = 0 \\ \mathrm{X, Y, U, V} \geq 0 \end{cases} \qquad (1.17)$$

Hence we have a system of equations.
$V^{\{T\}x}$*=0*, $U^{\{T\}Y}$*=0*− is called the complementary non-rigidity condition.
That is, if we introduce vectors y = (y_1...,y_n) и $\boldsymbol{v}$ = (v_1...,v_m), Then relation

(c+2xD+uA)x^T=0

xAT—b £0 (1.18)

(xAT—b)u^T =0

Will look like:

c + 2 ***x D*** + ***u A*** — ***в*** = ***0*** (1.19)

x AO — ***b*** + ***v*** = ***0*** (1.20)

в xT = ***0***, ***v uT*** = ***0*** (1.21)

The system (1.19) - (1.20) consists of n + m linear equations relatively $2(m+n)$ Unknown $xj,\ yj\ (j = 1\ldots,n), ui,\ vi\ (i = 1\ldots,m)$.
In addition, as follows from conditions (1.18), there should be:

If $x_j > 0$, then $y_j = 0$ (1.22)

If $y_j > 0$, then $x_j = 0$ (1.23)

If $u_i > 0$, then $v_i = 0$ (1.24)

If $v_i > 0$, then $u_i = 0$. (1.25)

Consequently, the desired solution of the system (1.19) - (1.20) can be an arbitrary inseparable basic solution of it, but such that the variables *xj* and *yj* (and also *ui* and *vi*) with the same indices cannot be basic at the same time. To find such a solution, one can apply any of the known methods of ***LP***, in particular, the artificial base method.
To this goal, we write the system (1.19) - (1.20) in the form
2 x D + u A — в = — c (1.26)
x AO + v = b. (1.27)

Without limiting universality, we will assume that the right-hand parts of this system are inalienable. According to the artificial basis method, in every equation of the system (1.26)—(1.27), which does not contain a basic variable, we introduce an artificial variable. Since the variables vi (i=1...,m) Can be considered basic, then the artificial variables z=(z1...,zn) We introduce only Eq. (1.26) and consider the auxiliary CLRF.
z iT min (1.28)

2 x D + u A — в + z = — c (1.29)
x AT+ v = b (1.30)

Where i = (1,1...,1) — n Dimensional unit vector.

Next, we solve this system, applying the rule, the transition from one basis to another, which is used in the simplex method.

We have, that $\boldsymbol{V_i x_i}$ = 0 = $\boldsymbol{U_j Y_j}$ For all i, j. The fulfillment of this condition means that if the variable $\boldsymbol{V_i}$ in the basic solution takes a positive value, then the variable $\boldsymbol{U_j}$can not be a basic variable and take a positive value. Similarly, the variables $\boldsymbol{x_i}$and $\boldsymbol{Y_j}$ cannot simultaneously take positive values. That is, this rule is also applied in the simplex method.

Quadratic simplex method

If the found admissible basic solution of this problem satisfies the conditions of complementary non-rigidity, then it determines the optimal solution of the original. Otherwise, we must pass to a new admissible basic solution. In this case, a new variable with a zero estimate is included in the basic variables. The simplex method with the conditions (1.22) - (1.25) for the solution of the auxiliary CLPP (1.28) - (1.30), constructed on the basis of the problem of convex quadratic programming (1.16), is called the quadratic simplex method (the algorithm of the ordinary simplex method)[9].

If in the optimal solution of the auxiliary CPRF (1.28) - (1.30) all the artificial variables zj (j=1...,n) take zero values, then discarding them, we obtain the DBR of the system (1.26) - (1.27). That part of it that corresponds to the variables of the initial problem of convex quadratic programming (1.16) is its optimal solution.

If the value of at least one of the artificial variables is different from zero in the optimal solution of the auxiliary CLRP (1.28) - (1.30), then the system (1.26) - (1.27) has no solutions, and consequently the set of saddle points of the Lagrange function of the initial problem of convex Quadratic programming is empty.

Let's say we solved the problem and found $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_{1,\ldots,}\boldsymbol{\lambda}_n)$ and now we can find a hyperplane that separates two classes:

w=$\sum_{i=1}^{n} \boldsymbol{\lambda_i c_i x_i}$, (1.31)

b=w$\boldsymbol{x_i}$-$\boldsymbol{c_i}$, $\boldsymbol{\lambda_i > 0}$ (1.32)

The summation does not go over the entire sample, but only over reference ones, for which $\boldsymbol{\lambda_i \neq 0}$.

The advantage of support vector methods is that the problem of convex quadratic programming has a unique solution. And also the support vector method is the best method of linear classification.

1.3. The solution of the problem of linear separation by the Fisher method

Fisher's linear discriminant in the original value is a method that determines the distance between the distributions of two different classes of objects or events. It can be used in problems of machine learning with a statistical (Bayesian) approach to solving classification problems[3].

Suppose that the training sample satisfies the following hypotheses in addition to the basic hypotheses of the Bayesian classifier:

Classes are distributed according to the normal law

Covariance matrices of classes are equal

Such a case corresponds to the best division of classes by the Fisher discriminator (in the original value). Then the statistical approach leads to a linear discriminant, and it is this classification algorithm that is now often understood as the linear discriminant of Fisher[4].

Under certain general assumptions, the Bayesian classifier reduces to the formula:

a(x)=arg $\max_{y\in Y} \lambda_y P_y p_y(x)$ (1.33)

Where Y is the set of answers (classes), x belongs to the set of objects X, P_y is the a priori probability of class y, p_y(x)is the likelihood function of class y, $\lambda_y$is the weighting factor (error price on object of class Y).

In the extension of all the above hypotheses, in addition to the hypothesis of the equality of covariance matrices, this formula has the form:

$$a(x)=\arg \max_{y\in Y}\left(ln(\lambda_y P_y) - \frac{1}{2}(x-\mu_y)^T \Sigma_y^{-1}(x - \mu_y) - \frac{1}{2}\, ln\,(|\Sigma_y^{-1}|) - \frac{n}{2}\ln(2\pi)\right), \quad (1.34)$$

Where $\mu_y = \frac{1}{l_y}\sum_{\substack{i=1\\ y_i=y}}^{l} x_i$ $\Sigma_y = \frac{1}{l_y}\sum_{\substack{i=1\\ y_i=y}}^{l} (x_i - \mu_y)\,(x_i - \mu_y)^T$ Approximations of the expectation vector and the covariance matrix of class Y, obtained as maximum likelihood estimates, l is the length of the training sample, $l_y$is the number of objects of class Y in the training sample x∈ R^n n.

This classification algorithm is

A quadratic discriminant. It has a number of drawbacks, one of the most important of which is the poor conditionality or degeneracy of the covariance matrix $\sum y$ with a small number of training elements of class Y, which can result in a strongly distorted result when the given matrix $\sum_y^{-1}$ in the past, And the entire classification algorithm is unstable, it will work poorly

(it is also possible that the inverse matrix $\sum_y^{-1}$ does not exist at all).

The linear discriminant of Fisher solves this problem

The main idea of the algorithm

When accepting the hypothesis of equality between covariance matrices, the classification algorithm takes the form:

$$\text{a(x)=arg max}_{y\in Y}(ln(\lambda_y P_y) - \tfrac{1}{2}\mu_y{}^T \textstyle\sum_y^{-1} \mu_y + x^T \textstyle\sum_y^{-1} \mu_y), \text{ OR a(x)=arg } \max_{y\in Y}\left(\beta_y + x^T \alpha_y\right) \quad (1.35)$$

Simplicity of classification by the linear *discriminant Fisher -*

One of the advantages of the algorithm: in the case of two classes in the two-dimensional characteristic space, the separating surface will be a straight line. If there are more than two classes, then the dividing surface will be piecewise-linear. But the main advantage of the algorithm in comparison with the quadratic discriminant is the decrease in the effect of the poor conditionality of the covariance matrix with insufficient data.

For small l_y approximations

$$\textstyle\sum_y = \frac{1}{l_y} \sum_{\substack{i=1 \\ y_i=y}}^{l} (x_i - \mu_y)\,(x_i - \mu_y)^T \quad (1.36)$$

Will give a bad result, therefore, even in those problems where it is known that classes have different forms, sometimes it is advantageous to use the heuristic of the Fisher discriminator and to consider the covariance matrices of all classes to be equal. This will allow us to calculate some "average" covariance matrix using the entire sample[6]:

$$\sum = \frac{1}{l_y} \sum_{i=1}^{l} (x_i - \mu_y)\,(x_i - \mu_y)^T \qquad (1.37)$$

The use of which in most cases will make the classification algorithm more stable.

Formulation of the problem:

Let $X_1$and X_2represent a certain problem of classifying points into two classes. And let $X_1 = \{x_1^1, x_2^1, \dots, x_{l_1}^1\}$Set of points belonging to class X

and $X_2 = \{x_1^2, x_2^2, \dots, x_{l_2}^2\}\ x_i^j \in R^d$, Belong to the class

Y.

By Fisher's method, we solve the problem posed and determine whether the points are separable or not. The linear Fisher discriminator is given by the vector w, which maximizes

$\text{J}\overrightarrow{(w)} = \overrightarrow{w^T}\, S_w \vec{w}\ /\ \overrightarrow{w^T}\ \text{SW}\ \overrightarrow{w'}$
Where S_w Selective covariance matrix[8]

The **covariance matrix** in probability theory is a matrix composed of the pairwise covariances of the elements of one or two random vectors. The covariance matrix of a random vector is a square symmetric non-negative definite matrix on the diagonal of which the variances of the vector components are located, and the off-diagonal elements are the covariance between the components. The covariance matrix of a random vector is a multidimensional analog of the variance of a random variable for random vectors. The covariance matrix of two random vectors is a multidimensional analog of the covariance between two random variables.
It is given by the following formulas:

$S_w = (m_1 - m_2)(m_1 - m_2)^T (1.38)$
$S_W = \sum_{i\epsilon\{1,2\}} \sum_{j=1}^{l_i} \left(x_j^i - m_i\right) \left(x_j^i - m_i\right)^T$ (1.39)
$m_i = l -\frac{1}{i} \sum_{j=1}^{l_i} x_j^i$;
$m_i = \sum_{j=0}^{l_i} x_j^i \ \frac{1}{l_i}$
l_i - Number of elements in a class l .

The parameter of the linear classifier is given by the vector$\overrightarrow{W}$:

q(x) = 0, for $\overrightarrow{W}' \times \vec{x}$ +b ≥ 0 (1.40)
q(x) = 1, for $\overrightarrow{W}' \times \vec{x}$ +b < 0 (1.41)

Example 1. Let the points belong to the classes X and Y.

X = {(−1; 3); (−1; −2); (−3; 2)}
Y = {(−1; 2); (2; 1)}
$m_1 - Mean\ point\ value x_i$;
$m_2 - Mean\ point\ value\ y_i$;
For each class, calculate the mean of the points:

$m_1 = (-1{,}6666;\ -1)$;
$m_2 = (0{,}5;\ 1{,}5)$;
$S_1\ Matrix\ of\ selective\ covariance\ of\ points$
$S_2 Matrix\ of\ selective\ covariance\ of\ points y_i$
$S_1[i,j]$ = Cov$(x_i\ , x_j)$; (1.42)
$S_2[i,j]$ = Cov$(y_i\ , y_j)$; (1.43)
S_W – Sum of two matrices
$S_W = S_1 + S_2 => S_W[i,j]$ = Cov$(x_i\ , x_j)$ + Cov$(y_i\ , y_j)$; (1.44)

$$S_W = \begin{pmatrix} 5{,}8333; & -4{,}5 \\ -4{,}5; & 7{,}5 \end{pmatrix}$$

S_W^{-1} – inverse matrix
The inverse matrix is a matrix S_W^{-1} multiplied by which the original matrix $S_W$gives the unit matrix E:$S_W * S_W^{-1} = S_W^{-1} * S_W$ = E (1.45)
We find the inverse matrix by the Gauss method, for the convenience of the definition we denote $S_W = A$ and $S_W^{-1} = X$, we use the equation A * X = E, where: X is the inverse matrix of the matrix A, and E is the identity matrix. We reduce the matrix A to a triangular matrix using the Gauss algorithm "direct motion". In parallel, we perform similar operations with the unit matrix E. We represent the obtained equation in its expanded form (see Figure 1.2)[9].

$$\begin{pmatrix} a_{11} + a_{12} + a_{13} + \dots & +a_{1n} \\ 0 + a_{22}^{(1)} + a_{23}^{(1)} + \dots & +a_{2n}^{(1)} \\ 0 + 0 + a_{33}^{(2)} + \dots & +a_{3n}^{(2)} \\ \vdots \quad \vdots & \\ 0 + 0 + 0 + \dots & +a_{nn}^{(n-1)} \end{pmatrix} \cdot \begin{pmatrix} x_{11} + x_{12} + x_{13} + \dots + x_{1n} \\ x_{21} + x_{22} + x_{23} + \dots + x_{2n} \\ x_{31} + x_{32} + x_{33} + \dots + x_{3n} \\ \vdots \quad \vdots \\ x_{n1} + x_{n2} + x_{n3} + \dots + x_{nn} \end{pmatrix} = \begin{pmatrix} e_{11} & + & 0 & + & 0 & + \dots & +0 \\ e_{21}^{(1)} & + & e_{22}^{(1)} & + & 0 & + \dots & +0 \\ e_{31}^{(2)} & + & e_{32}^{(2)} & + & e_{33}^{(2)} & + \dots & +0 \\ & & \vdots & & \vdots & & \\ e_{n1}^{(n-1)} & + & e_{n2}^{(n-1)} & + & e_{n3}^{(n-1)} & + \dots & +e_{nn}^{(n-1)} \end{pmatrix}$$

Figure.1.2 finding the inverse matrix by the Gauss method

Starting from the equation (Figure. 1.2), we express the values of the inverse matrix X. Multiplying the last row of the matrix A by the first column of the matrix X, we obtain the equality

$0 * X11 + 0 * X21 + 0 * X31 + 0 * X41 + \dots + Ann * Xn1 = En1$, Here we express $Xn1 = \frac{En1}{Ann}$. Similarly, we obtain from the following equation $0 * X11 + 0 * X21 + 0 * X31 + \dots + A(n-1)(n-1) * X(n-1)(1) + A(n-1)(n) * X(n)(1) = E(n-1)(1)$,
Express the $X(n-1)(1) = E(n-1)(1)/A(n-1)(n-1) - Xn1 * A(n-1)(n)/A(n-1)(n-1)$
and so on.
This operation is called the "reverse course of finding the elements of the inverse matrix X". Using formula (1.42), it is not difficult to calculate all the elements of the matrix X.
Matrix elements X.

$X_{ji} = \frac{1}{A_{jj}}(E_{ji} - \sum_{k=n}^{j} A_{jk}\, X_{ki}\ )$ (1.46)
i=1…n,
j=n…1

$$S_W^{-1} = \begin{pmatrix} 0{,}319; & 0{,}19 \\ 0{,}19; & 0{,}248 \end{pmatrix}$$

W – The normal of the hyperplane;

$W = S_W^{-1}(m_1 - m_2)$; (1.47)

$W = (-1{,}17;\ -1{,}035)$;

Finally, we calculate the hyperplane shift:

b = $(w * m_1 - w * m_2)$ / 2 (1.48)

$< w, x > -b \leq 0$;

$< w, x > \leq b$;

↙ ↘

Known Known b = 0,42337 ;

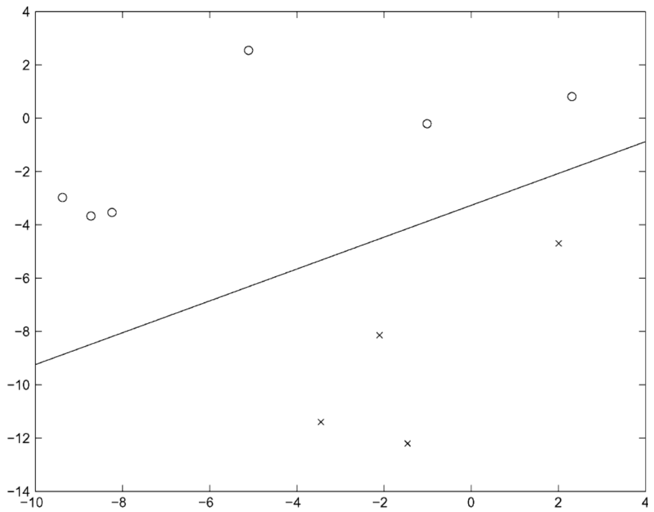


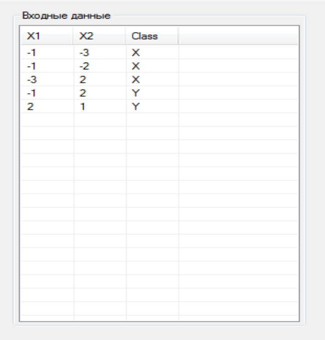
Figure 1.3. Hyperplane for linear Fisher discriminator.

The heuristic of the *linear Fisher discriminator* is in some way a simplification of the quadratic discriminant. It is used to obtain a more stable classification algorithm. It is most appropriate to use *the linear discriminant of Fisher* when Data for training is not enough. Owing to the main hypothesis on which the algorithm is based, it is most successfully solved by simple classification problems in which the forms are "similar" to each other in forms. The classification process by the linear Fisher discriminator can be described by the following scheme:

- Estimation of mathematical expectations
- Calculation of the general covariance matrix

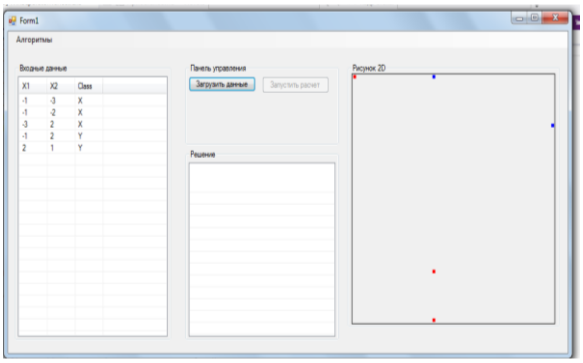
  and its inversion

Our sample of data


Входные данные

X1	X2	Class
-1	-3	X
-1	-2	X
-3	2	X
-1	2	Y
2	1	Y

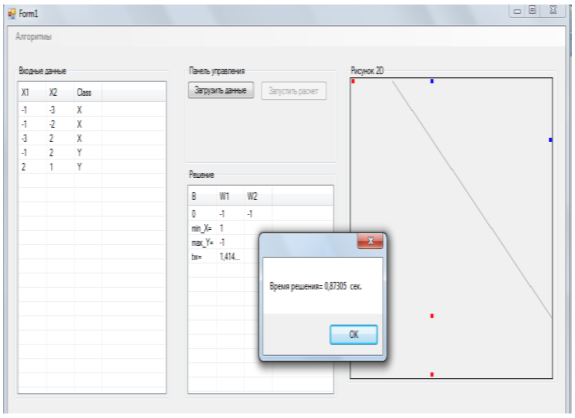
(Fig 1) Feeding this data to the program



Входные данные

X1	X2	Class
-1	-3	X
-1	-2	X
-3	2	X
-1	2	Y
2	1	Y

We start on calculation and we see the following result:



Входные данные

X1	X2	Class
-1	-3	X
-1	-2	X
-3	2	X
-1	2	Y
2	1	Y

Решение

B	W1	W2
0	-1	-1
min_X=	1	
max_Y=	-1	
tw=	1,414...	

Время решения= 0,87305 сек.

In this case, the points are linearly separable. And they are separated from each other by a separating hyperplane. Where *b* is the hyperplane shift, w is the normal of the hyperplane, *min_X* is the minimal scalar product of the vectors z_i by the normal, is calculated by the formula:

$$\min_{i=\overline{1,l}} < w, z_i >$$

max_Y is the maximum scalar product of the vectors z_i by the normal, is calculated by the formula:

$$\max_{j=\overline{1,m}} < w, p_j >$$

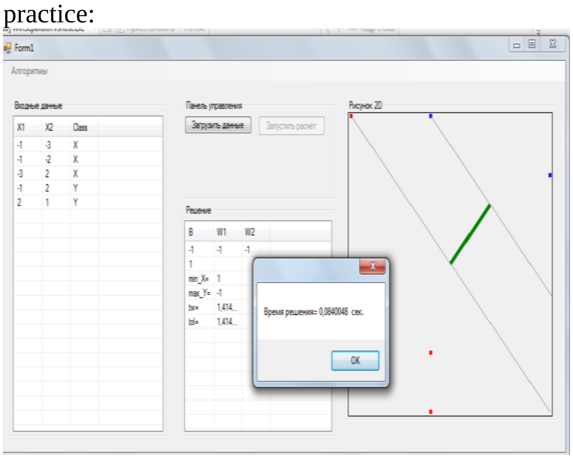
$< w, x > = b$ Hyperplane shift.

The thickness of the hyperplane t_w in separable cases

is calculated as follows:

$$\widehat{W} = \frac{W}{\|w\|}.t\left(\frac{W}{\|w\|}\right) = \left|\min_{i=\overline{1,l}} < z_i, \frac{W}{\|w\|} > - \max_{j=\overline{1,m}} < p_j, \frac{W}{\|w\|} > \right| t(\widehat{W}) = \left|\min_{i=\overline{1,l}} < z_i, \widehat{W} > - \max_{j=\overline{1,m}} < p_j, \widehat{W} > \right|$$

The quadratic method and see the following result in practice:



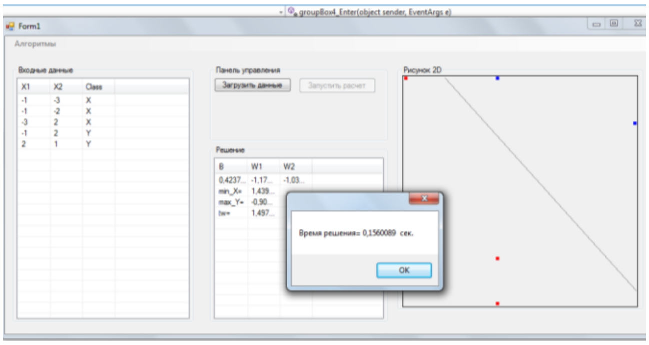
X1	X2	Class
-1	-3	X
-1	-2	X
-3	2	X
-1	2	Y
2	1	Y

B	W1	W2
-1	-1	-1
1		
min_X=	1	
max_Y=	-1	
tw=	1,414...	
tol=	1,414...	

Время решения= 0,0840048 сек.

Time consumed 0.0840048 Sec

The Fisher method in practice:



X1	X2	Class
-1	-3	X
-1	-2	X
-3	2	X
-1	2	Y
2	1	Y

B	W1	W2
0,4237...	-1,17...	-1,03...
min_X=	1,439...	
max_Y=	-0,90...	
tw=	1,497...	

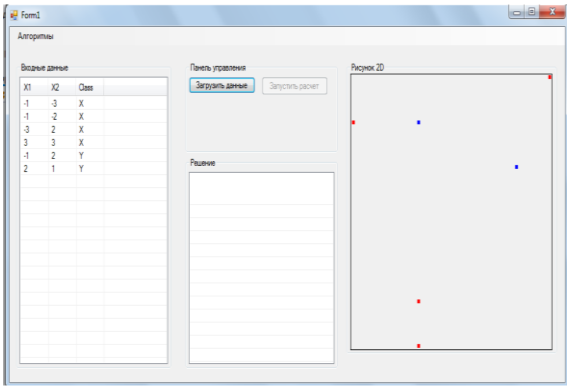
Время решения= 0,1560089 сек.

Time consumption 0.01560089 Sec

Results

METHOD	**TIME CONSUMPTION**
	EXAMPLE 1
Quadric	0.0840048
Fisher	0.01560089
	EXAMPLE 2
Quadric	0.0890051
Fisher	0.1480084
	EXAMPLE 3
Quadric	0.0320019
Fisher	0.0890051
	EXAMPLE 4
Quadric	0.035002
Fisher	0.0920052

Another side view where there is no solution



X1	X2	Class
-1	-3	X
-1	-2	X
-3	2	X
3	3	X
-1	2	Y
2	1	Y

The both methods did not find solution.

Conclusion

In the study, the problems related to the methods of solving the linear separation problem were considered. A software package was developed that numerically realizes the solution of the problem of linear separation of convex polyhedral by various computational means:

1) By solving the linear programming problem by the simplex method;

2) By reducing to the problem of quadratic programming;

3) Through the application of the Fisher method.

A general approach to solving the problem of linear separability of two data classes based on the listed methods was described. Computational experiments were conducted to analyze the effectiveness of the proposed approaches. A comparison of the effectiveness of the approaches considered was carried out for problems of the different dimensionality of space and the number of points defining separable sets with respect to the following indicators:

1) The time for solving the problem,

2) The thickness of the separation layer (formed between two hyperplanes supporting to shared sets) of two classes of points.

Comparison of methods

The task was to compare methods and determine which one is effective. When comparing, it is necessary to take into the complexity of the task, the time of performance of work and the value of the measure of the thickness of the compartment.

1. When input data of dimension $n = 2$, from the point of view of the thickness of separation by a hyperplane, the Simplex method and the quadratic programming method are effective. In most cases, they have the same thickness of separation t_w. The results obtained (for dimension $n = 2$) can also be compared by the time of the work. After doing a lot of experiments, we saw that the *quadratic programming method* is optimal for the time of execution of the work.
2. If the input data has the dimension $n = 3, \dots n = 10$, etc., then the best method is the method based on *quadratic programming*. The effectiveness of the method of quadratic programming does not depend on the dimension. When comparing the thickness of the compartment, in these tests quadratic programming provided the greatest thickness.

And by the time of the work, the optimal is the simplex method.

References

1- [1] Article D. Elizondo "Linear Separation: Some Test Methods".

2- [2] Bierman, H., C.P. Bonini, and W.H. Hausman. (Richard D. Irwin, Inc., Homewood, IL, 1977). Quantitative Analysis for Business Decisions 642 pp. Dykstra, D.P.

3- [3] BARANKIN, E. W., AND R. DORFMAN, 2 (1958) "On Quadratic Programming," University of Califoy'nia Publications in Statistics, 285-318.

4- Ershova A.V. Method for solving the problem of strong separation for multi-processor systems with mass parallelism // Parallel computing technologies (PaVT'2010): Proceedings of the International Scientific Conference (March 29 - April 2, 2010, Ufa). Chelyabinsk: Publishing Center of SUSU, 2010. pp. 660-661.

5- Golikov AI, Yevtushenko Yu.G. Theorems on alternatives and their application in numerical methods. Zhurn. calculated. Math. and Math. fiz. 2003. P. 43, No. 3. P. 354-375.

6- Gurin LG, Polyak BT, Raik E.V. Methods of projections for finding a general point of convex sets. Zhurn. calculated. Math. and Math. fiz. 1967. Vol. 7, No. 6. P. 1212-1228.

7- [4] J. Fourier, 1827,Memoire de l'Academie Royale des Sciences de l'Institute de France, 7 (1824), xlvij-lv. Chez Firmin Didot Pere et Fils.

8- [5] H. W. Kuhn, 1956, "Solvability and consistency for linear equations and inequalities," American Mathematical Monthly, vol. 63, pp. 217–232.

9- Hemdi A. Taha ,Doing, 2005, research operations. Seventh Edition, University of Arkansas, Fayetteville. Publishing house "William" Moscow | St. Petersburg | Kiev.

10- [7]Hillier, F.S., and G.J. Lieberman. (McGraw-Hill, Inc., New York, 1995). Introduction to Operations Research, sixth edition 998 pp.

11- [6] M. Sakarovitch, 1984 ,Optimisation Combinatoire Graphe et Programmation Lineaire. Paris: Hermann, Editeurs de Sciences et des Arts.

12- [8] M. A. Aizerman, E. M. Braverman, and L. I. Rozono´er , 1964, Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control, 25:821–837 .

13- [9] Mathematical Programming for Natural Resource Management (McGraw-Hill, Inc., New York, 1984). 318 pp.

14- [10] N. Aronszajn,1950Theory of reproducing kernels. Trans. Amer. Math. Soc., 686:337–404.

15- Vapnik VN, Chervonenkis A.Ya. Theory of pattern recognition. Moscow: Nauka, 1974. 416 p.

16- Vasiliev FP, Ivanitsky A.Yu. Linear programming. Moscow: Factorial, 2003. 352 p.

17- Voevodin Vl.V., Kapitonova A.P. Methods for describing and classifying the architectures of computing systems. M: Izd-vo MGU, 1994. 103 p.

18- Vorontsov K.V. Lectures on the method of support vectors. CC RAS, Moscow. URL: http: //www.ccas.ru/voron/download/SVM.pdf.

سرمد حمزة علي اسامة عبد الجليل علي سمير قيصر عجمي

مركز الحاسبة الالكترونية /جامعة المثنى

sarmad@mu.edu.iq osama@mu.edu.iq Samir@mu.edu.iq

المستخلص :

في هذا البحث، نحاول حل العادلات البسيطة التي تستخدم في إيجاد الحل الأمثل ، وذلك باستخدام أنواع مختلفة من المعادلات الخطية، ان استخدام مفهوم تعلم الاله (Machine Learning) الان يعتبر من اهم المواضع على نطاق العالم حيث استخداماتة واسعة تصل حتى لقيادة السيارة ولهذا الغرض سنستخدم مفهوم الفصل الخطي يستخدم على نطاق واسع في دراسة تعلم الآلة، من خلال هذه البحث وسوف نجد الطريقة المثلى لحل عن طريق مقارنة الوقت المستهلكة من قبل كل من طرق كوادريك و فيشر.