Czech Pattern Recognition Workshop 2000, Tomáš Svoboda (Ed.)
Peršlák, Czech Republic, February 2–4, 2000
Czech Pattern Recognition Society

Linear and quadratic classification toolbox for Matlab*

Vojtěch Franc, Václav Hlaváč
Czech Technical University, Faculty of Electrical Engineering, Center for Machine Perception
121 35 Praha 2, Karlovo náměstí 13, Czech Republic
{xfrancv,hlavac}@cmp.felk.cvut.cz

Michail I. Schlesinger
International Research and Training Centre of Information Technologies and Systems
Ukrainian Academy of Sciences, 252207 Kiev, 40 Prospect Akademika Glushkova, Ukraine
schles@image.kiev.ua

Abstract *The toolbox builds on Matlab and performs linear and quadratic statistical classification. It implements methods published in the recently appeared monograph [5]. Several of the reported methods are not widely known, provide solution to a more general tasks than before, and give a new systematic insight to the classical pattern recognition tasks. The toolbox is intended to demonstrate and help to understand algorithms for synthesis of linear or quadratic discrimination functions, mimimax learning, and unsupervised learning. There is a small new contribution reported in implementation of the generalized Anderson's task.*

1 Introduction

The classification toolbox is being created as a diploma thesis at the CTU Prague. It is supposed to demonstrate the linear and quadratic decision rules described in the recently published pattern recognition monograph [5] (which will be further references as the Book). Feature-based statistical pattern recognition methods from the Book were of interest for us. The developed toolbox focuses to linear discriminant functions including its generalization by nonlinear data mapping. The issue of learning decision rules in the statistical pattern recognition framework is covered in the toolbox as well.

The toolbox should help to understand relevant algorithms from the Book better and to demonstrate their functionality. The visualisation of the process leading to the solution and experimentation feasibility is stressed for this reason. The toolbox is not optimized for specific tasks deliberately.

A substantial attention was devoted to the *generalized Anderson's task*. Besides implementing the algorithms described in the Book we attempted to improve the method a little.

The toolbox is built on top of the Matlab, version 5.2. The reason for this choice is that Matlab provides many useful tools for data visualization, calculation with matrices, and the user interface independent on the operating system. The demonstrator environment is provided that allows the user to choose different algorithms, compare their behavior, provides tools to control the algorithm run interactively, creates synthetic input data or uses real ones.

2 Linear discriminant function, generalized Andersons's task, task formulations

Let X be a multidimensional linear space. The result of object observation is a point in the (feature) space X. Let k be an unobservable state k. Let us start with only two possible states $k \in \{1, 2\}$ for simplicity. It is assumed that conditional probabilities $p_{X|K}(x \,|\, k)$, $x \in X$, $k \in K$ are multidimensional Gaussian distributions. Mathematical expectations μ_k and covariance matrices σ_k, $k = 1, 2$, of these distributions are not known. The only knowledge available is that parameters (μ_1, σ_1) belong to a certain known set of parameters $\{(\mu^j, \sigma^j) \mid j \in J_1\}$, similarly for (μ_2, σ_2). Both upper and lower indices were used. Parameters μ_1 a σ_1 denote real but unknown statistical parameters of an object in the state 1. Parameters $\{\mu^j, \sigma^j\}$ for a certain upper index j represent one pair from possible pairs of values.

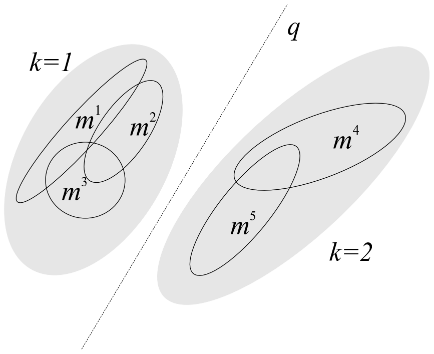

Figure 1: Generalized Anderson's task in 2D feature space.

*This research was supported by the Czech Ministry of Education under the grant VS96049 and the Research Topic JD MSM 212300013 Decision and Control for Industry.

Let us illustrate the mentioned case in Fig. 1, where five ellipses denote five Gaussian variables. Let us ignore the separating line q for a moment. Let, for instance, $J_1 = \{1, 2, 3\}$ and $J_2 = \{4, 5\}$. This assignment means that object is in a first state characterized by random vector with first or second or third distribution but it is not known which of them it is. A similar situation is for the second state and fourth and fifth distribution.

There are two classes of objects. Each class is described by a mixture of Gaussian distributions. Components of two mixtures are known. Only weights in the mixture are unknown. The task is to determine the state k when x is observed under the described partial knowledge of the apriori statistical model. It is shown in the Book that the studied task should be formulated as a statistical decision task with unknown intervention. In our particular case, we look for a strategy $q \colon X \to \{1, 2\}$, that minimizes the value

$$\max_{j \in J_1 \cup J_2} \varepsilon(j, \mu^j, \sigma^j, q) \,, \tag{1}$$

where $\varepsilon(j, \mu^j, \sigma^j, q)$ is a probability of the phenomenon that the Gaussian random vector x with mathematical expectation μ^j and covariance matrix σ^j fullfils either constraint $q(x) = 1$ for $j \in J_2$ or $q(x) = 2$ for $j \in J_1$. The Book states that the statistical decision task reduces to search for minimax solution in a space of mixture weights.

We are interested in the solution of the formulated task under an additional constraint on the decision strategy (discriminant function) q. The requirement is that the discriminant function should be linear, i.e. a hyperplane $\langle \alpha, x \rangle = \theta$ and

$$q(x) = \begin{cases} 1\,, & \text{if} \quad \langle \alpha, x \rangle > \theta\,, \\ 0\,, & \text{if} \quad \langle \alpha, x \rangle < \theta\,, \end{cases} \tag{2}$$

for a certain vector $\alpha \in X$ and the θ. The expression in angle brackets $\langle x, y \rangle$ denotes scalar product of vectors x, y.

The task (1) satisfying condition (2) minimizes the mean classification error and can be rewritten as

$$\{\alpha, \theta\} = \arg\min_{\alpha,\theta} \max_{j \in J_1 \cup J_2} \varepsilon(j, \mu^j, \sigma^j, q(x, \alpha, \theta)) \,. \tag{3}$$

This is a generalization of the known Anderson's and Bahadur's task [1] that was formulated and solved for a simpler case, where $|J_1| = |J_2| = 1$. Schlesinger proposed the mentioned generalized formulation and calls it *generalized Anderson's task*.

The generalized Anderson's task comprises two special cases that are important. The first one, the *optimal separation of finite point sets*, where the covariance matrices σ^j, $j \in J_1 \cup J_2$ are identity matrices. The finite point set $\tilde{X} = x_1, x_2, \ldots, x_n$ from the space X should be divided into two subsets $\tilde{X}_1$ and $\tilde{X}_2$ separated by a hyperplane. The hyperplane should be as distant as possible from both subsets $\tilde{X}_1$ and $\tilde{X}_2$. Actually, a vector α and a threshold θ are looked for that (a) for all $x \in \tilde{X}_1$ fulfills $\langle \alpha, x \rangle > \theta$, (b) for all $x \in \tilde{X}_2$ fulfills $\langle \alpha, x \rangle < \theta$, and maximizes the value

$$\min\left(\min_{x \in \tilde{X}_1} \frac{\langle \alpha, x \rangle - \theta}{|\alpha|} \,,\; \min_{x \in \tilde{X}_2} \frac{\theta - \langle \alpha, x \rangle}{|\alpha|} \right) \,. \tag{4}$$

The second special case, called the *simple separation of finite point set*, simplifies the previous case further. The subsets $\tilde{X}_1$, $\tilde{X}_2$ should be separated by any hyperplane, i.e. the condition (4) is ignored.

A separation of a finite set of points is an important step in the attempt to solve the generalized Anderson's task. If its solution only up to arbitrarily small ε is searched for is such a task called ε-solution. It is shown in the Book that it is a breakthrough to the linear discrimination. A substantial attention is devoted to in this paper.

The Book analyses the formulated tasks thoroughly. Let us sketch the main ideas here. The good news is that the minimized criterion (4) is unimodal. This allows to optimize using "hill climbing methods" without danger of ending up in local extreme. There are two bad news that relate to the criterion. It is neither convex nor differentiable. Thus the gradient does not exist and the fact that the gradient in the extreme equals to zero cannot be used. The Book shows how the Perceptron [4] and the algorithm proposed by Russian mathematician Kozinec [3] solves the most special task – the simple separation of a finite point set. For the more general task (generalized Anderson's task), the optimal separation of infinite point sets, it is proven in the Book that an optimization of a quadratic function on a convex polyhedron suffices, i.e. the convex optimization can be used. The solution was originally proposed in [6].

3 Toolbox overview

Let us present the structure of the toolbox and list the implemented algorithms first to give the reader the overview. Individual methods will be described in the sequel. If they are treated in detail in the Book they are just sketched here. If we had to make choices not described in the Book we devote to their description more space.

1. *Linear discriminant function.*
   - Separation of the finite sets of the points.
     - Perceptron.
     - Kozinec‘s algorithm.
     - ε-solution.
     - Linear Support Vector Machine.
     - Fisher's classifier [2].
       - Modified Perceptron.
       - Modified Kozinec‘s algorithm.
   - Generalized Anderson‘s task.
     - Original Anderson‘s solution.
     - ε-solution.
     - General solution (both method from the Book and its improvement).
     - Methods of the generalized gradient optimization.
2. *Quadratic discriminant function* part provides functions for nonlinear data maping that allow a synthesis of the quadratic discriminant function using the linear decision-making methods.

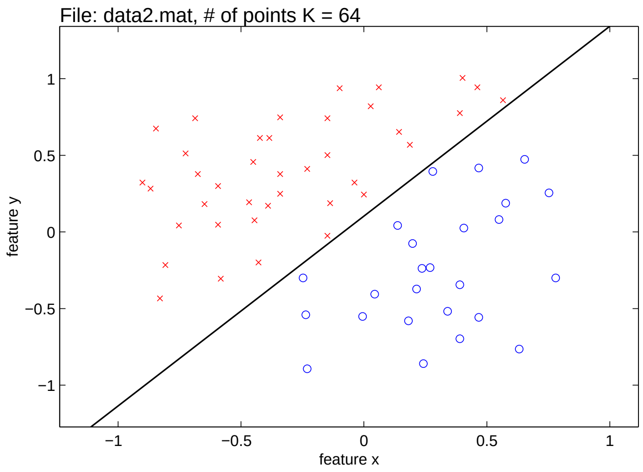

Figure 2: Linear separation of the finite set of points in a 2D feature space.

3. *Learning algorithms* part comprises both unsupervised and minimax learning.

4 Sketch of the methods implemented in the toolbox and described in the Book

4.1 Linear discriminant function

The linear discriminant function constitutes a substantial part of the toolbox. Both the algorithms for separating finite and infinite sets of points are implemented. The latter are also known as linear decision on the mixture of normal distributions.

4.1.1 Separation of finite sets of the points The separation of the finite sets of the points comprises both the algorithms for *nonoptimal separation* (as Perceptron, Kozinec‘s algorithm) and the algorithms for *optimal separation* (as Schlesinger’s ε-optimal separation). Besides the algorithms described in the Book the Vapnik’s linear Support Vector Machine [7] was added to the toolbox as it provides another approach to the task as compared to iterative algorithms described in the Book. Matlab optimization toolbox allowed us to implement the Support Vector Machine algorithm simply and efficiently.

Fig. 2 illustrates separation of the finite points in the two-dimensional feature space.

Mentioned iterative algorithms for separating finite sets can be used to create the *Fisher’s classifiers* as well. Let us notice that the construction of the *Fisher’s classifier* is an equal problem as solving the finite sets of non-equalities. The toolbox implements algorithms that find *Fisher’s classifiers* using the modified *Perceptron* and *Kozinec’s algorithm*.

Fig. 3 shows the obtained *Fisher’s classifier* for finite sets of the points in two dimensions. Hyperplanes dividing classes are shown as dashed lines. Full lines correspond to class vectors that determine the classifier.

4.1.2 Generalized Anderson‘s task Substantial part of the toolbox is devoted to solution of the generalized Ander-

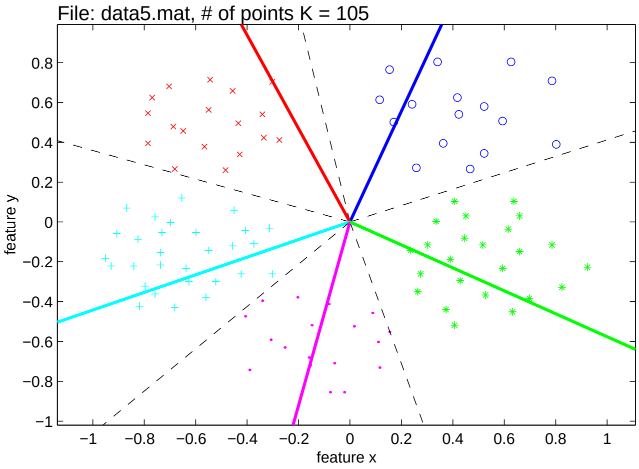

Figure 3: Fisher’s classifier.

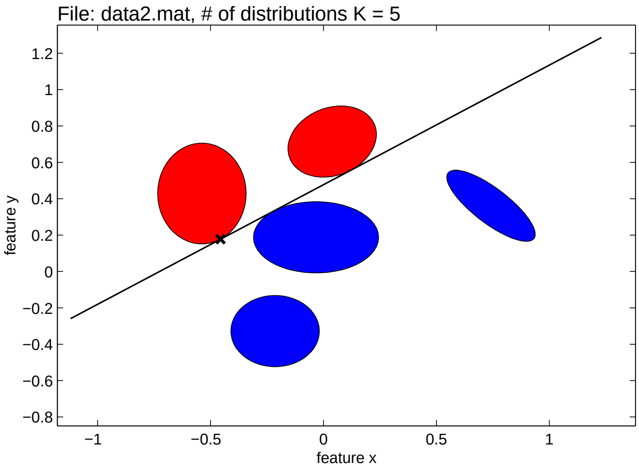

Figure 4: Generalized Anderson‘s problems.

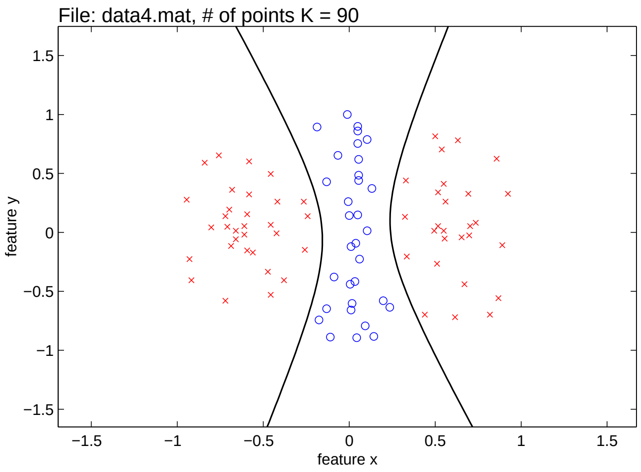

Figure 5: Separation of the finite point sets.

son's task. The more detailed description will be given in Section 5. The usage of the generalized Anderson's task is illustrated in Fig. 4 for two classes. The first class is determined by three Gaussian distributions and the second class by two distributions. The position of the separating hyperplane is determined, said informally, by pushing the hyperplane by growing ellipsoids in a certain way.

4.2 Quadratic discriminant function

The synthesis of a linear discriminant function is well understood in the literature. Many algorithms are available including those implemented in our toolbox. In general, the linear separation of points in the feature space does not suffice and the nonlinear separation hypersurface should be used instead. In some cases, it is of advantage to re-map the original feature space nonlinearly to a new space where the separation by a hyperplane is again possible. The new feature space has often higher dimension.

In our toolbox, the re-mapping is implemented for a quadratic discriminant function which is important in pattern recognition. For instance, the Bayesian strategy leads to the quadratic discriminant function. When a linear separation is found, the parameters of the linear hyperplane can be transformed to the parameters of the original feature space with the quadratic separating rule. The classification can be performed in both the original feature space using of the quadratic separation or in the re-mapped feature space using the linear hyperspace.

Fig. 5 shows the example where the quadratic discriminant function is applied to the data that are not linearly separable.

4.3 Learning algorithms can improve the statistical model

Learning algorithms can be used when there is not enough apriori knowledge about the classified objects. The statistical model of classes of recognized objects is expressed by the probability $p(x|k)$ which represents dependence between the observation x and the object state k. The probability

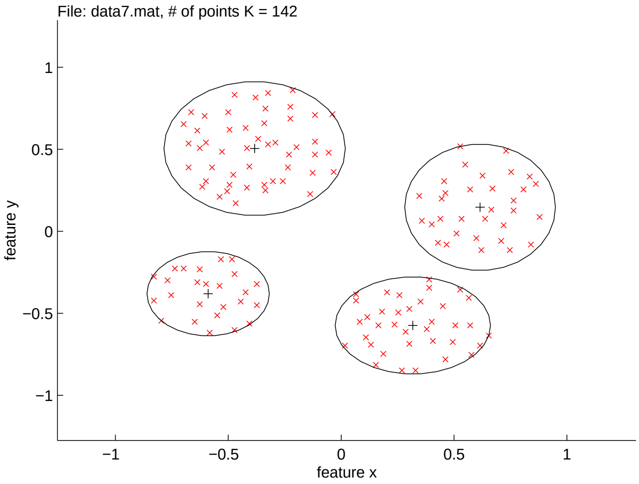

Figure 6: Unsupervised learning algorithms. Each ellipse shows the statistical model for one particular class, μ gives the center of the ellipse and σ determines the shape.

$p(x|k)$ is needed, for example, when the optimal Bayesian strategy is to be found. The toolbox allows to complete missing knowledge by learning.

4.3.1 Unsupervised learning algorithms A general class of unsupervised algorithms that learn the statistical model directly from unclassified data set is introduced in the Book. These algorithms classify the data set iteratively using Bayesian approach first. The *learning* using maximum likelihood estimate is performed on the outcome. The clustering algorithm *ISODATA* (called also k-means) and the *empirical Bayesian approach* by H. Robbins belong to this general class, for example. The Book proves the *convergence of the learning process* to the local or to the global maximum.

The learning algorithm finding parameters of the statistical model assuming normal distribution and apriori known number of classes is implemented in the toolbox. Algorithms for both cases, i.e. for statistically independent and dependent features are included. Fig. 6 demonstrates the obtained solution in the case of 4 classes and independent features. Ellipses are streched in the directions of axes x, y.

4.3.2 Minimax learning Described unsupervised learning algorithms, based on the maximal likelihood estimate, expect training data of a random nature and deteriorate their behavior severely if this condition is not fulfilled. If the random data are not available, the algorithms based on the *minimax learning* should be used instead. These algorithms search for the statistical model using nonrandom training set that describes the recognized classes well. The task is to find such a statistical model for which the data represent the given classes well, i.e. have high value of $p(x|k)$.

The algorithm implemented in the toolbox seeks a statistical model for one class with a normal distribution and correlated features. Notice that this task is equivalent to the search for a minimal ellipsoid that contains all data points from the training set.

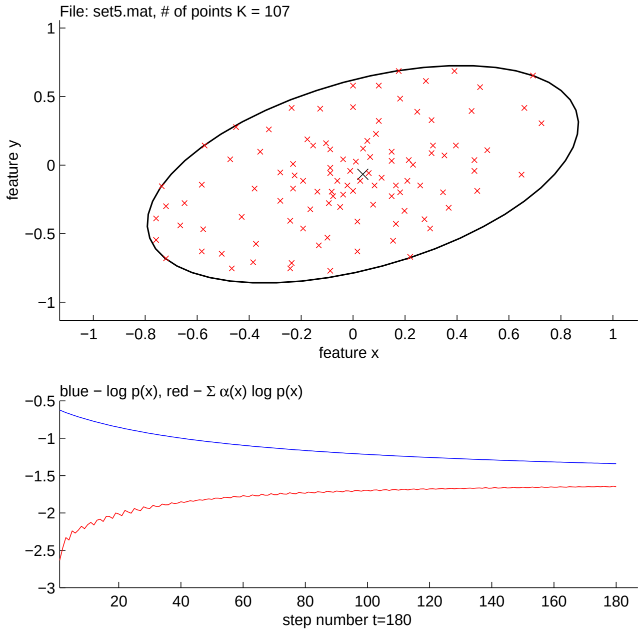

Figure 7: Minimax learning algorithm finds the statistical model for on clas only.

Fig. 7 demonstrates the minimax learning. The upper part illustrates the found ellipsoid. The lower part shows two curves indicating the quality of the solution. The Book is referred to for a more detailed description.

5 Solution of the generalized Anderson's task

Algorithms implemented in the toolbox that solve the *generalized Anderson's task* are described in this section. Recall that the generalized Anderson's problem was formulated in section 2. Besides sketching the solution to it that is proposed in the Book we attempted to explain the algorithms implemented in the toolbox and to improve the solution a little.

5.1 Equivalent task formulation enabling simpler solution

The original generalized Anderson's task can be easily transformed to the equivalent one that is more suitable for the analysis. The aim is to place the decision boundary $\langle\alpha, x\rangle = \theta$ into the origin of coordinates, i.e. $\langle\alpha, x\rangle = 0$.

This modification adds one more feature space dimension and one constant feature to each feature vector. The Gaussian distribution $N(\mu, \sigma)$ will transform into higher dimension so that $\mu' = \{\mu_1, \mu_2, \ldots, \mu_n, 1\}$. The covariance matrix has the last column and the last row filled by zeros in the new feature space now, since the last coordinate is constant. The decision boundary in the original space and decision boundary in new space are related by $\alpha' = \{\alpha_1, \alpha_2, \ldots, \alpha_n, -\theta\}$. We shall use only the transformed features μ'^j,σ'^j in the sequel and thus primed coordinates can be omitted, i.e. μ^j, σ^j, α will be used instead.

Thanks to the transformation, the linear hyperplane passing through the origin of coordinates can be found. Moreover, the set J_2 can be reflected symmetrically to the origin and sets of Gaussians J_1 and J_2 are merged into one set. Vectors μ'^j are introduced

$$\mu'^j = \begin{cases} \mu^j, & \text{for} \quad j \in J_1, \\ -\mu^j, & \text{for} \quad j \in J_2 \end{cases}$$

The covariance matrices do not change. The original sets of Gaussians $N(\mu^j, \sigma^j)$, $j \in J_1$ and $N(\mu^j, \sigma^j)$, $j \in J_2$, are transformed into the one set of Gaussians $N(\mu'^j, \sigma'^j)$, $j \in J$. Further on, primes can be omitted for simplicity again.

After the changes, the *new optimization criterion* can be written for the generalized Anderson's task as

$$\alpha = \arg\min_{\alpha} \max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j). \qquad (5)$$

Several characteristics of the function

$$\max_{j \in J} \varepsilon(\alpha, \mu^j, \sigma^j)$$

are proved in the Book that allow to design an elegant solution of the task. Here, only the principal characteristics will be reminded.

The solution is based on the geometric imagination when to each Gaussian distribution $N(\mu^j, \sigma^j)$ corresponds a set of points circumscribed by the multidimensional ellipsoid $E(\mu^j, \sigma^j)$. The set of points fullfils the unequation $\langle(\mu - x), \sigma^{-1}(\mu - x)\rangle \leq r^2$.

The book proves that the error $\varepsilon(\alpha, \mu^j, \sigma^j)$ decreases sharply if the radius of the ellipse is increased, i.e. the distance between the decision hyperplane and the ellipsoid center. That is the reason why we attemt the decision hyperplane in such a way that the radius of the smallest ellipsoid restricted by the hyperplane were the biggest. This leads to maxmin optimization.

When increasing ellipsoids we attempt to find a position when the hyperplane is pushed by ellipsoids that any any change of hyperplane position would decrease the size of one (the closest) ellipsoid.

We will need a contact point x_0^j between ellipsoid $E(\mu^j, \sigma^j)$ and a hyperplane passing origin that is given by a normal vector α. The contact point is determined as

$$x_0^j = \mu^j - \frac{\langle\alpha, \mu^j\rangle}{\langle\alpha, \sigma^j\alpha\rangle}\, \sigma^j\, \alpha\,.$$

We will need a radius $r(\alpha, \mu^j, \sigma^j)$ of the ellipsoid $E(\mu^j, \sigma^j)$ which is determined by the decision hyperplane. Radius r is determined after substitution as

$$r(\alpha, \mu^j, \sigma^j) = \frac{\langle\alpha, \mu^j\rangle}{\sqrt{\langle\alpha, \sigma^j\alpha\rangle}}\,,$$

Fig. 8 illustrates the idea.

Thanks to the relation between the $\varepsilon(\alpha, \mu^j, \sigma^j)$ and the distance $r(\alpha, \mu^j, \sigma^j)$, the criterion (5) can be modified to a form suited to the minimization better

$$\alpha = \arg\max_{\alpha} \min_{j \in J} r(\alpha, \mu^j, \sigma^j)\,.$$

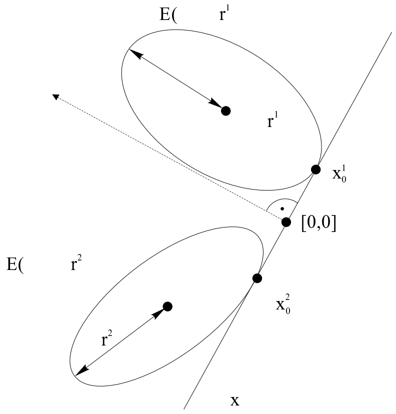

Figure 8: Geometrical illustration of the distance between the mean value μ^j and the contact point x_0^j that helps to define optimization criterion.

The good news is that the optimized function $\min_{j\in J} r(\alpha,\mu^j,\sigma^j)$ has one extreme only (it is unimodal). The bad news is that it is not differentiable.

In the Book, the next important fact is presented in the theorem about the necessary and sufficient conditions specifying the solution to the generalized Anderson's task. If the convex hull of set of the contact points x_0^j, $j \in J^0$ contains the origin of the coordinates then an arbitrary vector α', that is not collinear with the vector α, satisfies

$$\max_{j\in J} \varepsilon(\alpha',\mu^j,\sigma^j) > \max_{j\in J} \varepsilon(\alpha,\mu^j,\sigma^j) \ ,$$

where set J^0 contains indices of the distributions that have the bigger error for given α, i.e. $(j \mid j \in J^0) = \arg\min_{j\in J} r(\alpha,\mu^j,\sigma^j)$.

The proof of the above mentioned theorem, as given in the Book, provides an algorithm that solves the generalized Anderson's task. The algorithm outline is given there as well. Several subtask are mentioned in the Book and it is not specified which one ought to be used. We had several choices that, of course, determine the final properties of the algorithm. We shall describe those ones we picked up.

5.2 The outline of the algorithm solving the generalized Anderson's task

Input of the algorithm is given by the sets of Gaussian distributions, characterized with mean values μ^j and covariance matrices σ^j. Gaussian distribution $N(\mu^j,\sigma^j), j \in J_1$ determines the first class and Gaussian distribution $N(\mu^j,\sigma^j), j \in J_2$ determines the second class.

Result of the algorithm is the decision hyperplane given by the normal vector α and the threshold θ. The mean classification error is given by the criterion (1) and the algorithm minimizes it.

1. (Transformation of the distribution) The Gaussian distributions $N(\mu^j,\sigma^j)$, $j \in J_1 \cup J_2$, are transformed in such a way that the hyperplane parameters $\{\alpha,\theta\}$ are found that satisfy

$$(\alpha,\theta) = \arg\min_{\alpha,\theta} \max_{j\in J_1\cup J_2} \varepsilon(j,\mu^j,\sigma^j,(\alpha,\theta)) \ .$$

The obtained set of Gaussian distributions $N(\mu'^j,\sigma'^j), j \in J$, for which we find α' satisfies

$$\alpha' = \arg\min_{\alpha'} \max_{j\in J} \varepsilon(\alpha,\mu'^j,\sigma'^j) \ .$$

The mean values $\mu'^j, j \in J$ are calculated as

$$\mu'^j = \begin{cases} \{\mu_1^j,\mu_2^j,\ldots\mu_n^j,1\} & \text{for} \quad j \in J_1 \ , \\ -\{\mu_1^j,\mu_2^j,\ldots\mu_n^j,1\} & \text{for} \quad j \in J_2 \ . \end{cases}$$

Covariance matrices σ'^j, $j \in J$ are computed as

$$\sigma'^j = \begin{bmatrix} \sigma_{1,1}^j & \ldots & \sigma_{1,n}^j & 0 \\ \vdots & & \vdots & \vdots \\ \sigma_{n,1}^j & \ldots & \sigma_{n,n}^j & 0 \\ 0 & \ldots & 0 & 0 \end{bmatrix} \ , \ \text{for } j \in J_1 \cup J_2 \ .$$

The obtained variables α', μ'^j and σ'^j after the transformation will be written without primes in the sequel to simplify the notation, i.e. α, μ^j and σ^j.

2. *(Algorithm initialization).* Such a vector is found that all scalar products $\langle\alpha_1,\mu^j\rangle, j \in J$ were positive.

If such α_1 does not exist then the algorithm exits and indicates that there is not a solution with an error $< 50\%$.

3. *(Iterations).* The improving direction vector $\Delta\alpha$ is found which satisfies

$$\min_{j\in J} r(\alpha_t + k\cdot\Delta\alpha,\mu^j,\sigma^j) > \min_{j\in J} r(\alpha_t,\mu^j,\sigma^j) \ , \quad (6)$$

where $0 < k$, $k \in R$, $r(\alpha,\mu^j,\sigma^j)$ is the radius indirectly proportional to the uncertainty of the Gaussian distribution $N(\mu^j,\sigma^j)$, and t is the iteration number. The distance can be written as

$$r(\alpha,\mu^j,\sigma^j) = \frac{\langle\alpha_t,\mu^j\rangle}{\sqrt{\langle\alpha_t,\sigma^j\alpha_t\rangle}} \ .$$

If no vector $\Delta\alpha$ satisfying (6) is found then the current vector α_t solves the task. Go to the last step 5.

4. *(The new solution α_{t+1} is searched for as a point lying on a line segment between the old solution α_t and the improving direction $\Delta\alpha$ with smallest error)* The positive number k is looked for which satisfies

$$k = \arg\min_{k} \max_{j\in J} \varepsilon(\alpha_t + k\cdot\Delta\alpha,\mu^j,\sigma^j) \ . \quad (7)$$

A new vector α_{t+1} is calculated as

$$\alpha_{t+1} = \alpha_t + k\cdot\Delta\alpha \ .$$

If a quality change is smaller than a given limit Δ_r, i.e.

$$|r_{min}^t - r_{min}^{t-1}| < \Delta_r \ ,$$

then go to step 5 else continue in iterations by jumping to step 3.

5. *(End of the algorithm).* The inverse transformation is performed as in the step 1. The vector α_t should be primed again as $\alpha' = \alpha_t$. The solution of the task in the original n-dimensional space writes as

$$\begin{aligned} \alpha &= \{\alpha'_1, \alpha'_2, \ldots, \alpha'_n\} \,, \\ \theta &= -\alpha'_{n+1} \,. \end{aligned}$$

The algorithm exits in two cases. The first possibility is in the step 3 when the improved direction is looked for. It occurs if a deviation between the optimal and found solution is given by the precision of the algorithm finding the improving direction.

The second possibility can occur when a change in the solution quality is smaller than prescribed threshold after the optimization is performed in the step 4. Ideally, this case case should not occur but due to numerical reasons during optimization it is possible. Occurrence of this case means that the algorithm "got stuck" in some improving direction $\Delta\alpha$ so that current solution α_t does not need to be optimal. This case is undesirable and thus we intended to find suitable method that finds improving direction in the step 3 and avoids an event treated by the step 4.

5.3 Three subtasks where choices were made

The algorithm solving the general Anderson's task as described in Section 5.2 consists of several subtasks. We had to make some choices when implementing the toolbox. We suggested a few modifications or improvements. These are described below.

The first subtask concerns the need to find α that satisfies $\langle \alpha, x_0^j \rangle > 0$, $j \in J_0$. Any algorithm that sepparates finite sets of the points can be used. We have chosen the *linear Support Vector Machine* since it finds the optimal solution directly without troubles with numerical instabilities. The calculation is also the fastest of all algorithms we implemented. We could use an efficient *optimization toolbox* included in Matlab as well.

The second subtask corresponds to the step 3 of the algorithm described in Section 5.2. It calculates the improved $\Delta\alpha$. Several possibilities how to compute it are described in the Book. We suggest a modification here that was superior to original methods, in our experiments at least.

The third subtask we had to solve is the optimization of the criterion (7) as used in the step 4 of the algorithm in Section 5.2. A complicated function of one real variable has to be minimized.

Solutions to subtasks that were implemented in the toolbox are described below.

5.4 Search for an improved direction $\Delta\alpha$

The vector $\Delta\alpha$ must ensure that the error decreases in this direction, i.e.

$$\min_{j \in J} r(\alpha_t + k \cdot \Delta\alpha, \mu^j, \sigma^j) > \min_{j \in J} r(\alpha_t, \mu^j, \sigma^j) \,, \tag{8}$$

where k is any positive real number. It is proved in the Book that the vector $\Delta\alpha$ satisfying the condition (8) must fulfill

$$\langle \Delta\alpha, x_0^j \rangle > 0, j \in J^0 \,. \tag{9}$$

The set J^0 contains the distributions with biggest error, i.e.

$$\{j \mid j \in J^0\} = \arg\min_{j \in J} r(\alpha, \mu_j, \sigma_j) \,.$$

5.4.1 Approach A – Immediate use of $\Delta\alpha$ definition

The set of contact points x_0^j, $j \in J^0$ should be found first. The improvement must satisfy $\langle \Delta\alpha, x_0^j \rangle > 0$, $j \in J^0$. Algorithms that separate finite sets of the points as mentioned in Section 4.1.1 were used. The approach follows directly from the definition of $\Delta\alpha$. However, the results were the worst among all approaches we tested.

5.4.2 Approach B – Direction improving the error caused by the worst distribution

The direction $\Delta\alpha$ is searched so that the error corresponding to the worst distribution $N(\mu^j, \sigma^j)$, $j \in J^0$ decreases the quickest. This direction equals to the direction where the negatively taken derivative of the error function $\varepsilon(\alpha, \mu^j, \sigma^j)$, $j \in J^0$ is the biggest. The vector $\Delta\alpha$ fullfils

$$\Delta\alpha = \arg\max_{\Delta\alpha} \min_{j} \left(- \frac{\partial \varepsilon \left(\alpha + k \cdot \frac{\Delta\alpha}{|\Delta\alpha|} \right)}{\partial k} \right) \,.$$

or in other form

$$\Delta\alpha = \arg \max_{\{\Delta\alpha \mid |\Delta\alpha| = 1\}} \min_{j} \left\langle \Delta\alpha, \frac{x_0^j}{\sqrt{\langle \alpha, \sigma^j . \alpha \rangle}} \right\rangle \,.$$

If $\frac{x_0^j}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}}$ is denoted as vector y^j then we can write

$$\Delta\alpha = \arg\max_{\Delta\alpha} \min_{j} \frac{\langle \Delta\alpha, y^j \rangle}{|\Delta\alpha|} \,,$$

that is equivalent to the optimal separation of finite sets of the points using a hyperplane. We used the linear Support Vector Machine.

5.4.3 Approach C – Local approximation with Gaussian distribution with identity covariance matrix

The improved direction $\Delta\alpha$ is searched for that satisfies

$$\Delta\alpha = \max_{\Delta\alpha} \min_{j \in J} \frac{\langle \Delta\alpha, x_0^j \rangle}{|\Delta\alpha|} \,, \tag{10}$$

where vectors x_0^j are vectors given by

$$x_0^j = \mu^j - \frac{r_{min}}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}} \sigma^j \cdot \alpha \,,$$

where r_{min} is the radius of the smallest ellipsoid and the point x_0^j is the closest point between the hyperplane and the ellipsoid for all Gaussian distributions $N(\mu^j, \sigma^j)$, $j \in J$. Such improving direction $\Delta\alpha$ satisfies the necessary condition (9) due to the fact that $J^0 \subseteq J$. Moreover, it is the direction in which the error decreases for all distributions $\{\mu^j, \alpha^j\}$, $J \in J_0$ with the biggest error, i.e.

$$\varepsilon(\alpha, \mu^j, \sigma^j) < \varepsilon(\alpha + k \cdot \Delta\alpha, \mu^j, \sigma^j) \,, \quad j \in J \,.$$

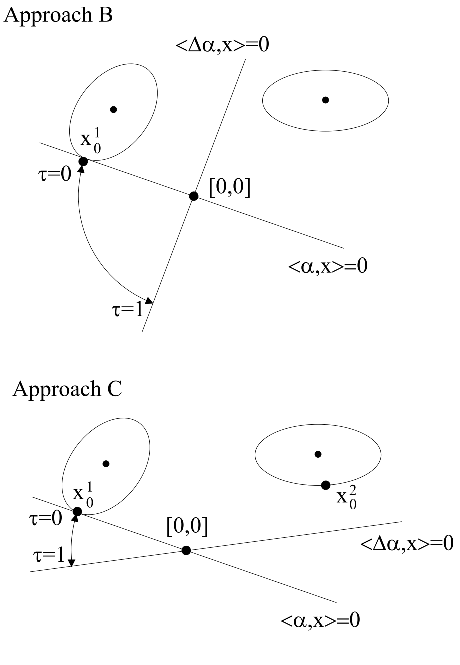

Figure 9: The improving direction finding

If the improving direction $\Delta\alpha$ is found according to (10) then all distributions $N(\mu^j, \sigma^j)$ in the point $x_0^j(\alpha, \mu^j, \sigma^j)$ are approximated by the Gaussian distribution $N(\mu^j, E)$, where E denotes identity matrix, i.e. covariance matrices are identity. Next, the optimal direction $\Delta\alpha$ is searched for this approximation. In the close neighborhood of points $x_0, j \in J$, thus $dalpha$ makes the optimal improving direction. This is a special case of the Anderson task with identity covariance matrices, which was mentioned in Section 2 as optimal separation of of finite point sets. For this special case the algorithm finds solution in one step, i.e. no iterations are needed.

Due to (10) the improving vector $\Delta\alpha$ can be found using any algorithm separating finite sets of the points by linear decision boundary. The linear Support Vector Machine was used in our implementation.

Fig. 9 shows the difference between approach B and C. A simplified 2D case is presented.

5.5 Optimization of the criterion of one real variable

Let us discuss several approaches how to solve the subtask in step 4 of the algorithm described in Section 5.2. Having finished the step 3 the current solution α and the improving direction $\Delta\alpha$ are available. The aim is to find vector $(\alpha + k \cdot \Delta\alpha)$ which determines the next value of the solution in the generalized Anderson's task. This vector has to minimize the error of the solution given by $\max_{j\in J} \varepsilon(\alpha_t + k \cdot \Delta\alpha, \mu^j, \sigma^j)$.

As was mentioned above, the error of the solution can be expressed using the distance $r(\alpha + k \cdot \Delta\alpha, \mu^j, \sigma^j)$. Having done that this subtask can be expressed as optimization of the criterion

$$k = \arg\max_{k} \min_{j\in J} r(\alpha(1-k) + k \cdot \Delta\alpha, \mu^j, \sigma^j)) \,, \qquad (11)$$

where k is a positive real number and the distance $r(\alpha + k \cdot \Delta\alpha, \mu^j, \sigma^j)$ is given by

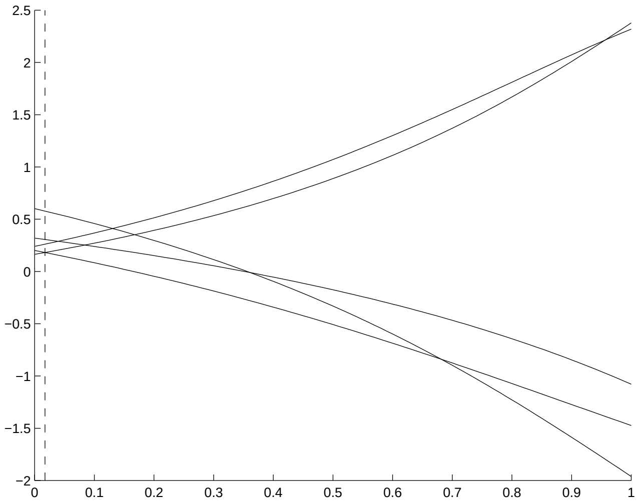

Figure 10: Optimization of functions $r(\alpha\cdot(1-\tau)+\tau\cdot\Delta\alpha, \mu^j, \sigma^j)$. The function values are on the vertical axis. The interval $\tau \in (0, 1\rangle$ is on the horizontal axis. The found maximum $f(\tau)$ is marked by a dashed line.

$$\frac{\langle(\alpha + k \cdot \Delta\alpha), \mu^j\rangle}{\sqrt{\langle(\alpha + k \cdot \Delta\alpha), \sigma^j \cdot (\alpha + k \cdot \Delta\alpha)\rangle}} \,. \qquad (12)$$

The found value k belongs to the infinite interval $(0, \infty)$. Thanks to the special property of the optimized function the maximization on the infinite interval can be avoided. Since the distance $r(\alpha, \mu^j, \sigma^j)$ depends on the direction of the vector α and does not depend on its absolute value, it holds

$$r(\alpha, \mu, \sigma) = r(c \cdot \alpha, \mu, \sigma) \,, \qquad (13)$$

where c is an arbitrary positive number. Hence we can rewrite the argument $\alpha + k \cdot \Delta\alpha$ of the r to the form

$$\frac{1}{1+k} \cdot \alpha + \frac{1}{1+k} k \cdot \Delta\alpha \,, \quad 0 < k < \infty. \qquad (14)$$

This is equivalent to

$$\alpha \cdot (1-\tau) + \tau \cdot \Delta\alpha \,, \quad 0 < \tau \leq 1 \,. \qquad (15)$$

The maximization on the infinite $(0, \infty)$ interval to the maximization on the finite interval $(0, 1\rangle$ was transformed. Let us denote the original optimization problem as

$$\tau = \arg \max_{0<\tau\leq 1} f(\tau) \,, \qquad (16)$$

where the function $f(\tau)$ is equal to

$$\min_{j\in J} \frac{\langle(\alpha \cdot (1-\tau) + \tau \cdot \Delta\alpha), \mu^j\rangle}{\sqrt{\langle(\alpha \cdot (1-\tau) + \tau \cdot \Delta\alpha), \sigma^j \cdot (\alpha \cdot (1-\tau) + \tau \cdot \Delta\alpha)\rangle}} . \qquad (17)$$

The function $f(\tau)$ has only one extreme and it is not differentiable as well as the radius $r(\alpha, \mu^j, \sigma^j)$. Refer to the Book for the proof.

Fig. 10 shows a set of the functions $r(\alpha \cdot (1-\tau) + \tau \cdot \Delta\alpha, \mu^j, \sigma^j)$ on the interval $\tau \in (0, 1\rangle$. The maximum of the $\min f(\tau)$ is marked by a dashed line.

Below we shall list the algorithms we tested. All of them are implemented in the toolbox.

5.5.1 Function maximization by sampling of the independent variable This algorithm finds the value τ in which the function $f(\tau)$ reaches its maximum. The τ is determined in such a way that a deviation between the optimal value τ^* and the found value τ is smaller than a prescribed precision, $|\tau^* - \tau| \leq \varepsilon_\tau$.

When finding the maximum, the original interval $\langle\tau_{beg}, \tau_{end}\rangle$ is split into $\langle\tau_{beq}, \tau_{mid}\rangle$ and $\langle\tau_{mid}, \tau_{end}\rangle$ according to a given ratio d. The ratio given by Fibonacci series is used instead of common bisection of an interval. We have a triplet $(\tau^t_{beg}, \tau^t_{mid}, \tau^t_{end})$ in each step t of the algorithm, which satisfies

$$f(\tau^t_{beg}) \leq f(\tau^t_{mid}) \geq f(\tau^t_{end}) \,. \tag{18}$$

The new triplet is calculated as follows. The bigger interval from the $(\tau^t_{mid} - \tau^t_{beg})$ and $(\tau^t_{end} - \tau^t_{mid})$ is split into two ones in given ratio d. The algorithmic description is

1. If $(\tau^t_{mid} - \tau^t_{beg}) > (\tau^t_{end} - \tau^t_{mid})$ then we calculate $\tau = \tau^t_{beg} + d \cdot (\tau^t_{mid} - \tau^t_{beg})$ and evaluate the expression $f(\tau^t_{mid}) > f(\tau)$. If the expression is satisfied then we assign
$$\tau^{t+1}_{beg} = \tau \,,$$
else
$$\tau^{t+1}_{end} = \tau^t_{mid} \,; \quad \tau^{t+1}_{mid} = \tau \,.$$
2. If $(\tau^t_{mid} - \tau^t_{beg}) \leq (\tau^t_{end} - \tau^t_{mid})$, then we calculate $\tau = \tau^t_{mid} + d \cdot (\tau^t_{end} - \tau^t_{mid})$ and evaluate the expression $f(\tau^t_{mid}) > f(\tau)$. If the expression is satisfied we assign
$$\tau^{t+1}_{end} = \tau \,,$$
else
$$\tau^{t+1}_{beg} = \tau^t_{mid} \,; \quad \tau^{t+1}_{mid} = \tau \,.$$

The condition (18) holds after performing the mentioned steps as the function $f(\tau)$ is unimodal. This procedure is iterated until the desired precision is reached, i.e $(\tau_{end} - \tau_{beg}) < \Delta\tau$.

The division ratio $d(t)$ is determined in each step t according to the Fibonacci series $F(t)$ given recursively $F(1) = 1$, $F(2) = 2$, $F(t) = F(t-1) + F(t-2)$.

The algorithm is useful for solving the generalized Anderson's task. Hundreds of steps were sufficient.

5.5.2 Function maximization by sampling of the function value The value of the variable τ is searched in which the function $f(\tau)$ reaches its maximum. The τ is determined so that difference between the optimal functional value $f(\tau^*)$ and the found value $f(\tau)$ is smaller than a given precision, $|f(\tau) - f(\tau^*)| \leq \varepsilon_f$.

The value of the independent variable τ lies inside the interval $T = (0, 1\rangle$. At the beginning, the algorithm has to determine the upper limit f_{up} and the lower limit f_{low} satisfying the following: such $\tau \in T$ exists that $f(\tau) \geq f_{low}$ and does not exist $\tau \in T$ that $f(\tau) \geq f_{up}$. In each step, the algorithm checks whether such a $\tau \in T$ exists that fulfills $f(\tau) \geq f_{mid} = \frac{1}{2}(f_{low} + f_{up})$. If it is true the lower limit increases to $f_{low} = f_{mid}$ else the upper limit decreases to $f_{up} - f_{low} \leq \varepsilon_f$.

The key problem is how to evaluate expression $f(\tau) \geq c$ and if it holds how to determine interval of validity $\tau \in T$. The function (17) has to be analyzed. More detailed analysis including the solution can be found in the Book.

At last, the initial values of the limits f_{low} a f_{up} have to be determined. The book does not provide specific solution. We set the initial values having in mind properties of the generalized Anderson's task as follows. The lower limit can be set as

$$f_{low} = \min_{j \in J} r(\alpha, \mu^j, \sigma^j) \,.$$

In the case that the τ satisfying

$$\min_{j \in J} r(\alpha \cdot (1-\tau) + \tau.\Delta\alpha, \mu^j, \sigma^j) \geq \min_{j \in J} r(\alpha, \mu^j, \sigma^j)$$

does not exist, there is no chance to find the improving direction $\Delta\alpha$ and the optimization cannot be successful.

On the other hand, the following holds having in mind a geometric interpretation of the generalized Anderson's task

$$\min_{j \in J} r(\alpha \cdot (1-\tau) + \tau \cdot \Delta\alpha, \mu^j, \sigma^j) \leq \max_{j \in J} r(\alpha, \mu^j, \sigma^j) \,.$$

For the upper limit we obtain

$$f_{up} = \max_{j \in J} r(\alpha, \mu^j, \sigma^j) \,.$$

Function maximization by sampling of the function value seemed to be slightly worse in our experiments than the method based on sampling the independent variable described in the previous subsection.

5.6 ε-solution of the Generalized Anderson's task

The ε-solution method finds such a decision boundary α, θ that corresponds to the classifier error smaller than a given limit ε_0, i.e.

$$\max_{j \in J_1 \cup J_2} \varepsilon(j, \mu^j, \sigma^j, q(x, \alpha, \theta)) < \varepsilon_0 \,. \tag{19}$$

Previous formula defines *ε-solution of the Generalized Anderson's task.*

The optimal value of the criterion (3) does not need to be found. The algorithm is thus easier. It does not matter that the strict optimum is not found in many practical tasks.

Classes in the Anderson's task are determined by the set of Gaussian distributions. The error of one Gaussian distribution corresponds to the radius of the (multidimensional) uncertainty ellipsoid lying in the halfspace given by the decision hyperplane. If the maximal allowed classifier error is known then the set of the Gaussian distribution $N(\mu^j, \sigma^j)$, $j \in J_1$ can be expressed by an infinite set of the points. Let denote it as X_1. Points are positioned inside the geometric intersection of the ellipsoids restricted by the decision boundary $\{\alpha, \theta\}$. The centers of ellipsoids correspond to mean values μ^j, $j \in J_1$ and the ellipsoids shape is determined by covariance matrices σ^j, $j \in J_1$. An infinite set X_2 for the second class given by Gaussian distributions $N(\mu^j, \sigma^j)$, $j \in J_2$ is expressed analogously.

The task can be reduced to the separation of infinite sets of the points $X_1(r)$ a $X_2(r)$ provided that the maximal limit of the classifier error is given. The idea is proven in the Book.

A finite set of points can be separated by the Kozinec's or Perceptron, as was described in Subsection 4.1.1. Moreover, they are able to separate infinite sets of points after small modification. It is possible on condition the infinite sets are creative described. The sets X_1 and X_2 satisfy just this condition. So that modified Kozinec's algorithm or Perceptron are able to find solution of the ε-solution of the Generalized Anderson's problem.

Very important feature of these algorithms is capability to find solution in finite number of steps, if such solution exist.

6 Conclusions

The described linear and quadratic classification toolbox is still being developed. The current version is available for experiments at the `http://cmp.felk.cvut.cz/~hlavac/Public/Pu/LinClassToolbox`

I mentioned algorithms were tested on synthetically generated data sets. Experiments with real data are being prepared. They should appear in the first author's diploma thesis that is supposed to be submitted in January 2000, i.e. before the Czech Pattern Recognition Workshop. It is likely that we could report about it at the workshop.

Anyway, we can summarize experience gained in experiments with algorithms implemented in the toolbox. In general, the practical experience matches to the expected properties of algorithms as described in the Book.

When implementing the generalized Anderson's task we had to make several choices that are not described in the Book in needed detail. The algorithm outline, as described in Subsection 5.2, was filled by the methods their combination seemed to perform the best. Regarding the improving direction, as specified by the step 3, the local approximation with Gaussian distribution with identity covariance matrix as described in Subsubsection 5.4.3 performed the best.

The step 4 of the algorithm searches for new solution between the old solution and the improving direction with the smallest error. When doing so the optimization of the criterion of one real variable should be performed. The maximization of an unimodal criterial function by sampling of the independent variable, as described in Subsubsection 5.5.1, gave the best results.

It can be of advantage to use the ε-solution to the generalized Anderson's task as described in the Subsection 5.6. This occurs in practical cases when the strict optimal solution is not needed and the solution with error smaller than any predefined error suffices. If such a solution exists then the algorithm finds it in a finite number of steps. Otherwise, there the information when to stop the algorithm is not available. This is a disadvantage, of course.

References

[1] T.W. Anderson and R.R. Bahadur. Classification into two multivariate normal distributions with different covariance matrices. *Annals Math. Stat.*, 33:420–431, June 1962.

[2] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, Part II:179–188, 1936.

[3] B.N. Kozinec. Rekurentnyj algoritm razdelenia vypuklych obolocek dvuch mnozestv, in Russian (Recurrent algorithm separating convex hulls of two sets). In V.N. Vapnik, editor, *Algoritmy obucenia raspoznavania*, pages 43–50. Sovetskoje radio, Moskva, 1973.

[4] F. Rosenblatt. *Principles of Neurodynamiscs: Perceptron and theory of Brain Mechanisms*. Spartan Books, Washington, D.C., 1962.

[5] M.I. Schlesinger and V. Hlaváč. *Deset přednášek z teorie statistického a strukturního rozpoznávání, in Czech (Ten lectures from the statistical and structural pattern recognition theory)*. Vydavatelství ČVUT, Praha, Czech Republic, 1999. The English version is under preparation and is expected to be published by the Kluwer Academic Publishers.

[6] M.I. Schlesinger, V.G. Kalmykov, and A.A. Suchorukov. Sravnitelnyj analiz algoritmov sinteza linejnogo resajuscego pravila dlja proverki sloznych gipotez, in Russian (Comparative analysis of the synthesis algorithms of the linear decision rule for check of complex hypotheses). *Avtomatika*, (1):3–9, 1981.

[7] V.N. Vapnik. *The nature of the statistical learning theory*. Springer-Verlag, New York, 1995.