Computer Analysis of Images and Patterns : Proceedings of the 9th International Conference, W. Skarbek (Ed.), pages 169–176, Warsaw, Poland, September 2001, Springer.

A contribution to the Schlesinger's algorithm separating mixtures of Gaussians

Vojtěch Franc and Václav Hlaváč

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

Abstract. This paper contributes to the statistical pattern recognition problem in which two classes of objects are considered and either of them is described by a mixture of Gaussian distributions. The components of either mixture are known, and unknown are only their weights. The class (state) of the object k is to be found at the mentioned incomplete a priori knowledge of the statistical model and the known observation x. The task can be expressed as a statistical decision making with non-random interventions. The task was formulated and solved first by Anderson and Bahadur [1] for a simpler case where each of two classes is described by a single Gaussian. The more general formulation with more Gaussians describing each of two classes was suggested by M.I. Schlesinger under the name generalized Anderson's task (abbreviated GAT in the sequel). The linear solution to GAT was proposed in [5] and described recently in a more general context in a monograph [4].

This contribution provides (i) a formulation of GAT, (ii) a taxonomy of various solutions to GAT including their brief description, (iii) the novel improvement to one of its solutions by proposing better direction vector for next iteration, (iv) points to our implementation of GAT in a more general Statistical Pattern Recognition Toolbox (in MATLAB, public domain) and (v) shows experimentally the performance of the improvement (iii).

1 Definition of the Generalized Anderson's Task

Let X be a multidimensional linear space. The result of object observation is a point x in the n-dimensional feature space X. Let k be an unobservable state which can have only two possible value $k \in \{1, 2\}$. 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 $\sigma_k, k = 1, 2$, 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 $\{(\mu^j, \sigma^j) | j \in J_1\}$ and similarly for (μ_2, σ_2) it is set $\{(\mu^j, \sigma^j) | j \in J_2\}$ (J_1, J_2 denote set of indexes). Parameters μ_1 and σ_1 denote real but unknown statistical parameters of an object in the state $k = 1$. Parameters $\{\mu^j, \sigma^j\}$ for a certain upper index j represents one pair from possible pairs of values.

The goal is to find a decision strategy $q\colon X \to \{1, 2\}$ mapping feature space X to space of the classes K that minimizes the value

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

where $\varepsilon(q, \mu^j, \sigma^j)$ is a probability that the Gaussian random vector x with mathematical expectation μ^j and covariance matrix σ^j fulfills either constraint $q(x) = 1$ for $j \in J_2$ or $q(x) = 2$ for $j \in J_1$. In other words, it is the probability that the random vector x will be classified to the different class then it actually belongs to.

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

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

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

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

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

This is a generalization of the known Anderson's and Bahadur's task [1] that was formulated and solved for a simpler case, where each class is described by only one distribution, i.e. $|J_1| = |J_2| = 1$. A toy example is shown in Figure 1.

2 Solution to the generalized Anderson's task

There are several approaches how to solve GAT. They are thoroughly analyzed in [4]. First, we will list them and (in next section) we will focus on one of them which we have improved.

- **General Algorithm Framework.** The general method based on proofs which leads to the optimal solution defined by the criterion (3). We shell devote our attention to this approach in Section 3
- **Solution by the help of optimization using general gradient theorem.** The criterion (3) defining GAT is unimodal but it is neither convex nor differentiable [4]. Thus standard hill climbing methods cannot be used but so called *generalized gradient optimization theorem* [6] can be used instead.

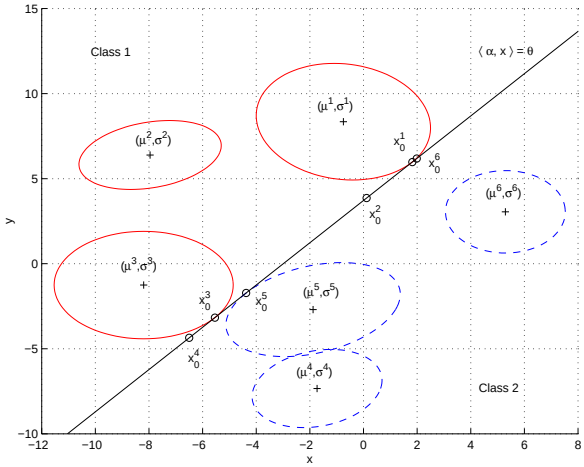

Fig. 1. An example of GAT. The first class is described by Gaussians with parameters $\{(\mu^1, \sigma^1), (\mu^2, \sigma^2), (\mu^3, \sigma^3)\}$ and the second class by $\{(\mu^4, \sigma^4), (\mu^5, \sigma^5), (\mu^6, \sigma^6)\}$. Mean values μ^j are denoted by crosses and covariance matrices σ^j by ellipsoids. The line represents found linear decision rule which maximizes Mahalanob is distance from the nearest distribution $\{(\mu^1, \sigma^1), (\mu^3, \sigma^3), (\mu^4, \sigma^4)\}$. The points x_0^j laying and the decision hyperplane have the nearest Mahalanobis distance from given distribution.

– ε**-optimal solution.** The ε-solution method finds such a decision hyperplane (α, θ) that the probability of wrong classification is smaller than a given limit ε_0, i.e.

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

The optimal solution (3) does not need to be found so that the problem is thus easier. The task is reduced to splitting two sets of ellipsoids their radius is determined by the ε_0. This task can be solved by Kozinec's algorithm [4] which is similar to Perceptron learning rule.

3 Algorithm framework

In this section we will introduce the general algorithm framework which solves GAT. Our contribution to this algorithm will be given in Section 4. The algorithm framework as well as concepts we will use are thoroughly analyzed and proved in [4]. We will introduce them without proofs.

Our goal is, in accordance with the definition of GAT (see Section 1), to optimize following criterion

$$\{\alpha, \theta\} = \operatorname*{argmin}_{\alpha,\theta} \max_{j \in J_1 \cup J_2} \varepsilon(q(x, \alpha, \theta), \mu^j, \sigma^j). \tag{4}$$

Where α, θ are parameters of a decision hyperplane $\langle \alpha, x \rangle = \theta$ we are searching for. Vectors $\mu^j, j \in J_1 \cup J_2$ and matrices $\sigma^j, j \in J_1 \cup J_2$ are parameters of

Gaussians describing the first class J_1 and the second class J_2. This optimization task can be transformed to equivalent optimization task

$$\alpha = \underset{\alpha}{\operatorname{argmax}} \min_{j \in J} r(\alpha, \mu^j, \sigma^j). \tag{5}$$

The task (5) is more suitable for both analysis and computation. The transformation consists of (i) introducing homogeneous coordinates by adding one constant coordinate, (ii) merging both the classes together by swapping one class along origin of coordinates, (iii) expressing of probability $\varepsilon(q(x,\alpha), \mu^j, \sigma^j)$ using number $r(\alpha, \mu^j, \sigma^j)$.

(i) Introducing homogenous coordinates leads to formally simpler problem since only vector $\alpha' = [\alpha, -\theta]$ is looked for and the threshold θ is hidden in the $(n+1)$-th cooridinate of vector α. New mean vectors $\mu'^j = [\mu^j, 1]$ and covariance matrices $\sigma'^j = \begin{bmatrix} \sigma^j & 0 \\ 0 & 0 \end{bmatrix}$ are used after it. Notice that new covariance matrices have the last column and the last row zero since constant coordinate was added.

(ii) Having decision hyperplane $\langle \alpha', x' \rangle = 0$, which passes the origin of coordinates, it holds that $\varepsilon(q(x', \alpha'), \mu'^j, \sigma'^j) = \varepsilon(q(x', \alpha'), -\mu'^j, \sigma'^j), j \in J_2$. It allows us to merge the input parameter sets into one set of parameters $\{(\mu''^j, \sigma''^j) | j \in J\} = \{(\mu'^j, \sigma'^j) | j \in J_1\} \cup \{(-\mu'^j, \sigma'^j) | j \in J_2\}$. *To make notation simpler we will use further on notation $x, \alpha, \mu^j, \sigma^j$ instead of $x'', \alpha'', \mu''^j, \sigma''^j$.*

(iii) The number $r(\alpha, \mu^j, \sigma^j)$ is the Mahalanobis distance between the normal distribution $N(\mu^j, \sigma^j)$ and a point x_0^j laying on the hyperplane $\langle \alpha, x \rangle = 0$ which has the smallest distance. It has been proven [4] that $\varepsilon(q(x, \alpha, \theta), \mu^j, \sigma^j)$ monotonically decreases when $r(\alpha, \mu^j, \sigma^j)$ incereases which allows us exchange minmax criterion (4) to maxmin (5). The point with the smallest distance is

$$x_0^j = \underset{x | \langle \alpha, x \rangle = 0}{\operatorname{argmin}} \langle (\mu^j - x), (\sigma^j)^{-1} \cdot (\mu^j - x) \rangle = \mu^j - \frac{\langle \alpha, \mu^j \rangle}{\langle \alpha, \sigma^j \cdot \alpha \rangle} \sigma^j \cdot \alpha.$$

The number $r(\alpha, \mu^j, \sigma^j)$ can be computed as

$$r(\alpha, \mu^j, \sigma^j) = \langle (\mu^j - x_0^j), (\sigma^j)^{-1} \cdot (\mu^j - x_0^j) \rangle = \frac{\langle \alpha, \mu^j \rangle}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}}.$$

The objective function in criterion (5) is unimodal and monotonically decreasing function. The algorithm which solves the criterion is similar to hill climbing methods but the direction in which the criterion impoves cannot be computed as a derivative since it is not differentiable. The main part of algorithm consists of (i) finding of an improving direction $\Delta\alpha$ in which criterion descends and (ii) determining how much to move in the direction $\Delta\alpha$. The convergence of the algorithm crucially depends on the method found the improving direction $\Delta\alpha$.

First, we will introduce the algorithm framework, then the original method finding $\Delta\alpha$ will be given and finally we will introduce our improvement which concerns finding $\Delta\alpha$.

3.1 General algorithm framework for GAT

Algorithm:

1. **Transformations.** First, as we mentioned above, we have to perform transformations of $(\mu^j, \sigma^j), j \in J_1 \cup J_2$. Then we obtain one set $(\mu^j, \sigma^j), j \in J$. The algorithm processes the transformed parameters and its result is vector α also in the transformed space. When the algorithm exits we can easily transform the α back into the original space.
2. **Initialization.** Such a vector is found that all scalar products $\langle \alpha_1, \mu^j \rangle, j \in J$ are positive. If such α_1 does not exist then the algorithm exits and indicates that there is not a solution with probability of wrong classification less than 50%. Lower index t of the vector α_t denotes iteration number.
3. **Improving direction**. The improving direction $\Delta\alpha$ is found which satisfies

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

   where k is a positive real number. If no vector $\Delta\alpha$ satisfying (6) is found then the current vector α_t solves the task and algorithm exits. In the opposite case the algorithm proceeds to the following step.
4. **Movement in the improving direction.** A positive real number is looked for which satisfies

$$k = \operatorname*{argmax}_{k>0} \min_{j \in J} r(\alpha_t + k \cdot \Delta\alpha, \mu^j, \sigma^j).$$

   A new vector α_{t+1} is calculated as $\alpha_{t+1} = \alpha_t + k \cdot \Delta\alpha$.
5. **Additional stop condition.** If a change in criterial function value during t_{hist} iterations is less than giving limit Δ_r, i.e.

$$|\min_{j \in J} r(\alpha_t, \mu^j, \alpha^j) - \min_{j \in J} r(\alpha_{(t-t_{hist})}, \mu^j, \alpha^j)| \leq \Delta_r,$$

   then the algorithm exits else continues in iterations by jumping to step 3.

The algorithm can exit in two cases. The first possibility is in the step 3 when the improve is not found then the vector α_t corresponds to the optimal solution (proof in [4]).

The second possibility can occur when a change in criterial function value after t_{hist} iterations is smaller than prescribed threshold Δ_r. This phenomenon is checked in step 5. Ideally, this case should not occur but due to numerical solution during optimization it is possible. The occurance of this case means that the algorithm got stuck in some improving direction $\Delta\alpha$ and the 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 this case.

The main part of the algorithm is step 3 and step 4. The improving direction is searched for in the step 3. Having found the improving direction we should decide how much to move in this direction, it is solved in the step 4. Following subsections deal with these two subtasks.

3.2 Numerical optimization of the criterion depending on one real variable

Having finished the step 3 the current solution α and the improving direction $\Delta\alpha$ are available. The aim is to find the vector $\alpha_{t+1} = \alpha_t + k \cdot \Delta\alpha$ which determines the next value of the solution. This vector has to maximize $\min_{j \in J} r(\alpha + k \cdot \Delta\alpha, \mu^j, \sigma^j)$, so we have new optimization problem

$$k = \operatorname*{argmax}_{k>0} \min_{j \in J} r(\alpha + k \cdot \Delta\alpha, \mu^j, \sigma^j),$$

where k is a real positive number. To solve this optimization task we have to find a maximum of a real function of one real variable. This task we solve numerically (details are given in [4]).

3.3 Search for an improving direction $\Delta\alpha$

Here we will describe step 3 of the algorithm, that finds a direction in which the error decreases. Overall effectivity of the algorithm crucially depends upon this direction as the performed experiments have shown. Such vector $\Delta\alpha$ must ensure that the classification error decreases in this direction, i.e.

$$\min_{j \in J} r(\alpha_t + k \cdot \Delta\alpha, \mu^j, \sigma^j) > \min_{j \in J} r(\alpha_t, \mu^j, \sigma^j), \tag{7}$$

where k is any positive real number. It is proved in [4] that the vector $\Delta\alpha$ satisfying the condition (7) must fulfill

$$\langle \Delta\alpha, x_0^j \rangle > 0, j \in J^0. \tag{8}$$

The set J^0 contains the distributions with highest error or lowest Mahalanobis distance, i.e. $\{j | j \in J^0\} = \operatorname{argmin}_{j \in J} r(\alpha, \mu^j, \sigma^j)$.

The original approach, proposed in [4], determines improving direction as

$$\Delta\alpha = \operatorname*{argmax}_{\Delta\alpha} \min_{j \in J^0} \frac{\langle \Delta\alpha, y^j \rangle}{|\Delta\alpha|}, \tag{9}$$

where $y^j = \frac{x_0^j}{\sqrt{\langle \alpha, \sigma^j \cdot \alpha \rangle}}$, then *dalpha* is a direction in which the classification error for the worst distributions $j \in J^0$ decreases the quickest. The task (9) is equivalent to the separation of finite point set with maximal margin. We used linear Support Vector Machines (SVM) algorithm [2].

Following section describes the new method, which approximates Gaussian distribution with an identity covariance matrix, tries to improve the algorithm.

4 Local approximation of the Gaussian distribution by the identity covariance matrix

The main contribution of the paper is described in this section. We have proposed the new approach how to find the improving direction in which the error of the optimized criterion decreases (see Section 3.3).

Each distribution $N(\mu^j, \sigma^j)$ is approximated in the point x_0^j by the Gaussian distribution $N(\mu^j, E)$, where E denotes the identity matrix. In the case when all the covariance matrixes are identity, GAT is equivalent to the optimal separation of finite point sets. So we determine the improving vector $\Delta\alpha$ as the optimal solution for the approximated distributions.

The points x_0^j for all the distributions are found first as

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

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

then the improving direction $\Delta\alpha$ is computed as

$$\Delta\alpha = \operatorname*{argmax}_{\Delta\alpha} \min_{j \in J} \frac{\langle \Delta\alpha, x_0^j \rangle}{|\Delta\alpha|} \ . \tag{10}$$

The optimization problem (10) is solved by linear SVM algorithm.

5 Experiments

The aim of the experiments was to compare several algorithms solving GAT. We have tested algorithms on synthetic data. Experiments on real data are foreseen.

The experiments have to determine (i) the ability of algorithms to find an accurate solution (close to the optimal one) and (ii) their robustness with regard to various input data. We created 180 data sets corresponding to task with known solutions.

We tested three algorithms. The first two algorithm GAT-ORIG and GAT-NEW fulfill the general algorithm framework (see Section 3) and the third one GAT-GGRAD uses the generalized gradient theorem (see Section 2). The algorithm GAT-ORIG uses the original method (see Section 3.3) and the algorithm GAT-NEW uses our improvement (see Section 4). All the algorithms mentioned in this paper are implemented in the Statistical Pattern Recognition Toolbox for Matlab [3].

For each algorithm we had to prescribe a stopping condition. The first stop condition is given by a maximal number of algorithm steps which was set to 10000. The second one is a minimal improvement in the optimized criterion which was set to $1e-8$.

Using of synthetically generated data allows us to compare the solution found by an algorithm to the known optimal solution. Moreover, we could control

complexity of the problem: (i) the number of distributions describing classes varying from 4 to 340; (2) dimension of data varying from 2 to 75; (3) the number of additional distributions which do not affect the optimal solution but which make work of the tested algorithm harder. Total number of randomly generated testing instances was 180 used.

Having results from the algortihms tested on the synthetic data we computed following statistics from: (i) mean deviation between the optimal and the found solution $E(|\varepsilon_{found} - \varepsilon_{optimal}|)$ in [%], (ii) maximal deviation between the optimal and the found solution $\max(\varepsilon_{found} - \varepsilon_{optimal})$ in [%], (iii) number of wrong solutions, i.e. their probability of wrong classification is worse by 1% compared to the optimal solution. When this limit is exceeded we consider the solution as wrong.

Table 1 summarizes the results. We conclude that the algorithm GAT-NEW appeared as the best. This algorithm found the optimal solution in all tests. The algorithm GAT-GRAD failed in 7% in our tests. The algorithm GAT-ORIG failed in 91.5% in our tests.

Table 1. Experiment results

	GAT-ORIG	GAT-NEW	GAT-GGRAD
Mean deviation in [%]	8.96	0	0.29
Maximal deviation in [%]	35.14	0	10.56
Wrong solutions in [%]	91.5	0	7

6 Conclusions

We have proposed an improvement of the Schlesinger's algorithm separating the statistical model given by the mixture of Gaussians (Generalized Anderson's task, GAT) [4]. We composed and extensively tested three algorithms solving GAT. One of them was our improvement. The tests were performed on 180 test cases given by synthetic data as needed the ground truth. Our improvement outperformed the other algorithms. All the tested methods are implemented in the Statistical Pattern Recognition Toolbox for Matlab [3] that is free for use.

Acknowledgement

V. Franc was supported by the Czech Ministry of Education under Research Programme J04/98:212300013 Decision and control for industry. V. Hlaváč was supported by the Czech Ministry of Education under Project LN00B096.

References

1. T.W. Anderson and R.R. Bahadur. *Classification into two multivariate normal distributions with differrentia covariance matrices.* Annals Math. Stat., 33:420-431, june 1962.
2. Burges C., J. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, Vol. 2, Number 2, p. 121-167*, 1998.
3. V. Franc. Statistical pattern recognition toolbox for Matlab, Master thesis, Czech Technical University in Prague, 2000. `http://cmp.felk.cvut.cz`.
4. M. I. Schlesinger and V. Hlaváč. *Deset přednášek z teorie statistického a strukturního rozpoznávání, in Czech (Ten lectures on statistical and structural pattern recognition).* Czech Technical University Publishing House, Praha, Czech Republic, 1999. English version is supposed to be published by Kluwer Academic Publishers in 2001.
5. M.I. Schlesinger, V.G. Kalmykov, and A.A. Suchorukov. Sravnitelnyj analiz algoritmov sinteza linejnogo reshajushchego pravila dlja proverki slozhnych gipotez, in Russian (Comparative analysis of algorithms synthesising linear decision rule for analysis of complex hypotheses). *Automatika*, 1(1):3–9, 1981.
6. N.Z. Shor. *Nondifferentiable optimization and polynomial problems.* Kluwer Academic Publisher, Dordrecht, The Netherlands, 1998.