Two approaches for solving tasks of pattern recognition and reconstruction of functional dependencies

Tatjana Lange[1], Pavlo Mozharovskyi[2], Gabor Barath[3]

[1,3] University of Applied Sciences, Merseburg, Automation and Feedback Control, Germany
[2] National Technical University of Ukraine “Kyiv Polytechnic Institute”

Abstract: The presentation compares two different methods of pattern recognition – the so-called Alpha-Procedure (or α-procedure), that has been developed by V.I. Vasil’ev and T.I. Lange since the Seventies, and the Support Vector Machine, proposed by V. Vapnik in parallel in the same time period.

The Alpha-Procedure can be considered as an inductive approach where the dimension of the pattern space is stepwise extended by adding the “best” features as new “axes”. The “best” features are the features with the maximum “discrimination power” that is defined on basis of the local optimum principle.

The Support Vector Machine follows a deductive approach where the optimal division of the patterns (in sense of the so-called “generalized portrait”) is searched in the complete space of measured features.

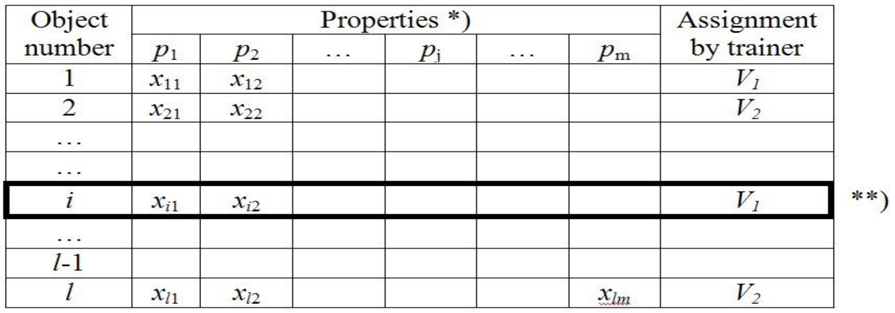
Together with the geometric-algorithmic comparison of the two approaches there will be also shown their connection to the computation of the multidimensional Data Depth.

Key words: Pattern recognition, Alpha-Procedure, Support Vector Machine, Data Depth

1 The Task of Pattern Recognition

Normally, we start with given (measured) data for different objects and their properties that are assigned to different classes by a “trainer”. This set of classified data we call “training set”. The table below shows an example.

We search a separating hyperplane (fig. 1) or a separating hypersurface (fig. 2) using for that the initially measured and classified data (the training set) and we hope that this plane (or surface) will afterwards work **automatically** for other measured data and define without the trainer to which class V_1 or V_2 the object $\mathbf{x}_{l+k}$ belongs.



Object number	Properties *)						Assignment by trainer	
	p_1	p_2	...	p_j	...	p_m		
1	x_{11}	x_{12}					V_1	
2	x_{21}	x_{22}					V_2	
...								
...								
i	x_{i1}	x_{i2}					V_1	**)
...								
l-1								
l	x_{l1}	x_{l2}				x_{lm}	V_2	

*) A property p_j can be the weight, the height, the number of horns, the lengths of the wool e.t.c. Let’s assume that we have 2 classes: V1 - wolves and V2 – sheep

**) The row *i* corresponds to the object No. *i* that can be considered as vector $\mathbf{x}_i$ in the *m*-dimensional space of properties. x_{ij} are the elements (coordinates) of the vector $\mathbf{x}_i$.

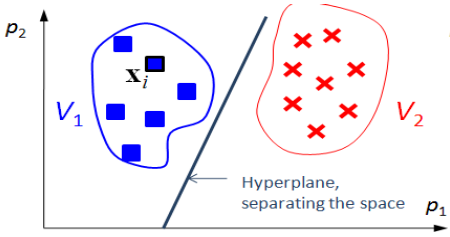


Fig. 1: Separation by hyperplane

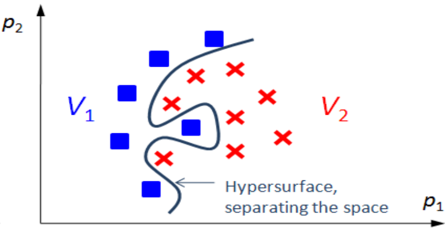


Fig. 2: Separation by hypersurface

Rectifying space

In the case shown in fig. 2 the separating surface can be converted into a separating plane but only in the so-called rectifying or extended space.

For that purpose we use together with or instead of the properties p_j combined properties which express the interaction of some original properties which can be described mathematically as $\varphi_k(p_j, p_{j\pm k})$. The table below shows an example.

Object number	Properties *)					Assignment by trainer
	...	...	$\varphi_k(p_1, p_5) = a \cdot x_1 \cdot x_5^2$	...	p_m	
1	...	...	$x_{11} \cdot x_{15}^2$			V_1
...	...	...				
i	...	...	$x_{i1} \cdot x_{i5}^2$			V_1
...	...	...				
l	...	...	$x_{l1} \cdot x_{l5}^2$		x_{lm}	V_2

The separating plane is described by $\sum_{i=1}^{M} a_i \varphi(p) = 0$ where *M* – is the number of properties within the new “extended” or “rectifying” space.

This is the reason that in the **Support Vector Machine** (SVM) and in the **α-Procedure** only the term **hyperplane** is used.

Both the α-procedure and SVM correspond to the nature of task described above. But they are based on different ideas and they often have different application areas and also different strengths and weaknesses.

SVM is well suited for tasks where we have a very large number of objects in a range of 10^3 and more. These objects may be the picture elements of scans which are used for the recognition and comparison of handwritings, e.g. for the recognition of forged signatures in criminology. The essential of this method is that only support vectors of the objects are considered but this is done in the complete property space. That means only the vectors of objects are taken into account which are on a critical trajectory.

As consequence this method is very sensitive according the existence or absence of critical support points within the data. The algorithmic calculation comes to the Kuhn-Tucker optimization.

The **α-procedure** is well suited for tasks where we have natural properties "defining" the objects of a class but these properties are often hidden, e.g. (important) physical components within the smelting in metallurgy or hidden signs telling about the prosperity of a bank in the financial world. Let us use another clear example for the explanation: The property p_i describing the existence of horns (p_i =1) or the absence of horns (p_i =0) is such a "defining" property for the automated separation of wolves and sheep. The α-procedure finds these properties in an automated way and uses them for the separation of the object classes.
For the α-procedure all objects (but not all of their properties) are equally important. This is the reason for a large number of operations. But it gives a stable separation using a very small number (2-4) of "**useful**" properties which we call **features** (but with another meaning as in SVM) after certain transformations. During the calculation process the investigator can easily observe the stepwise separation of the object clouds V_1 and V_2 together with the accumulation of features because they are always represented in a **plane.** This is the main idea of the α-procedure. In addition, the focus on feature selection instead of support object selection depends less on coincidences and is therefore more robust.
Both methods have one weakness which will be discussed in the outlook.

2 SVM, Generalized Portrait and their geometric interpretation

The "Generalized Portrait" is the basis of the Support Vector Machine (SVM). The term "Generalized Portrait" stands for the clarification which positioning parameters of **<u>one</u>** class V_1 within the property space are important and measurable and can be used for the comparison of V_1 with other classes, e.g. V_2.
Such parameters are:

- Vector Ψ builds a "generalized portrait" which is characterized by the "average" direction of all object vectors of class V_1 (see fig. 3). This vector Ψ describes the direction to class V_1 and the distance of V_1 from the origin of coordinates.

$\psi = \varphi / c_1(\varphi)$, with φ - as unit vector with the same direction as Ψ. $\varphi = \Psi / |\Psi|$.

$c_1(\varphi)$ characterizes the distance of class V_1. $c_1(\varphi)$=0B is vertical to the "tangential" plane of class V_1 which is the nearest to the origin of coordinates.
The minimum vector Ψ is searched, that means Ψ is defined from the condition

$$\langle \Psi, \Psi \rangle \to \min \text{, i.e. } 1/|\Psi| = c_1(\varphi) \to \max .$$

The search of the "generalized portrait" Ψ comes to the task of optimization:

$$\max_{|\varphi|=1} \min_{i} (\varphi, x_i) = \max_{|\varphi|=1} c_1(\varphi)$$

In the case where the origin of the coordinates meets the "centre" of class V_1 then the generalized portrait vector Ψ=0 (see fig. 4).

● The SVM uses the idea of the “generalized portrait” for the definition of the mutual locations of two classes (patterns) in the space of properties p_i (see fig. 5). For this purpose a separating hyperplane is searched under the condition

$$\Pi(\varphi) = [c_1(\varphi)] - [c_2(\varphi)] \to \max$$

with $[c_1(\varphi^*) + c_2(\varphi^*)]/2 = \langle x, \varphi \rangle$ and φ^* as optimal unit vector.

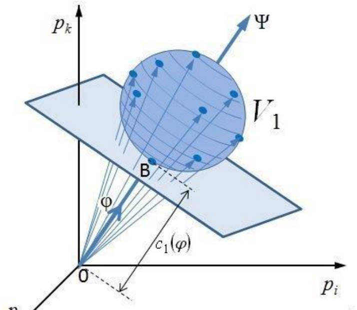


Fig. 3: Geometric interpretation of “Generalized Portrait”

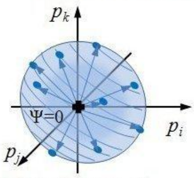


Fig. 4: Special case of “Generalized Portrait”

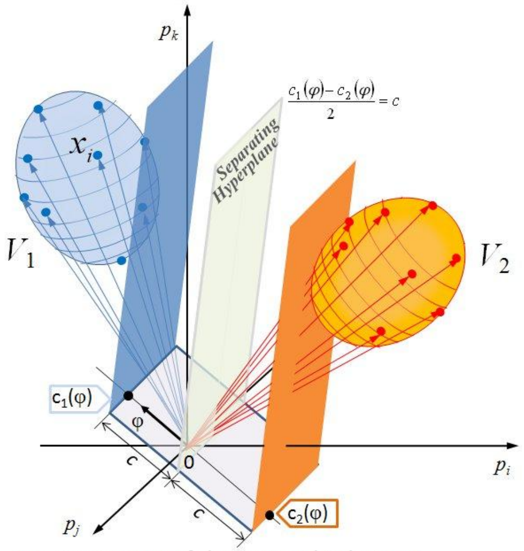


Fig. 5: Separation of classes V_1 and V_2 by SVM

3 α-Procedure

First, let us explain the idea. We consider a number of objects belonging to two different classes V_1 and V_2. All objects are characterized by a certain amount of properties with different (continuous) values.

Using a given training set the α-procedure selects out of all existing properties only those properties which separate the objects faultlessly or, at least, with a minimum of faults. These “useful” properties will be transformed into new **features** by the procedure described below.

At the beginning we define for each property the **discrimination power** $F(p_i) = \omega_i / l$ with ω_i - number of correctly classified objects and l – number of all objects. For the classification we use only the properties for which $F(p_i) \geq 1/n_0$ with n_0 as dimension of the property space. Then we select the property with the best discrimination power **as basis feature** f_0 and represent it together with its values for the objects as an axis as shown in fig 6.

On the next stage we add a second property p_k to the coordinate system and define the positions of the objects in the plane that is built by the axes f_0 and p_k.

After that we create a new axis $\tilde{f}_1$and turn it around the origin of the coordinate system by the angle α up to the moment when the **projections** of the objects onto this new axis give us the best separation of the objects (fig. 6).
We remember that property together with its separation power and its optimal value α (this is from what the name of the procedure comes).

We repeat this procedure for all remaining properties and select that one property that gives the best separation of the objects on its corresponding axis $\tilde{f}_1$. That we take now as next **feature.**

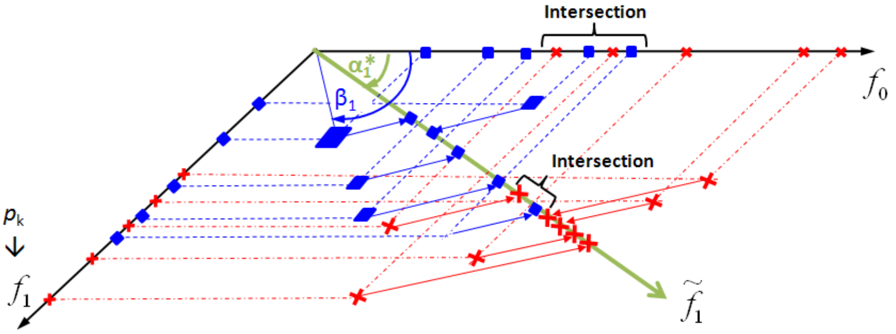


Fig: 6: Mapping of objects onto the optimal rotated axis $\tilde{f}_1$

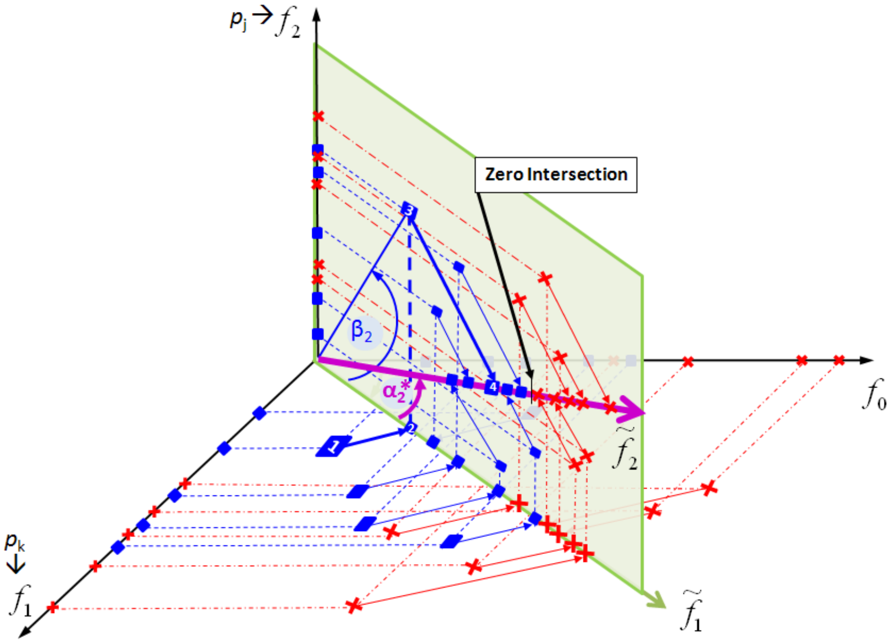


Fig: 7: Final separation of objects in the plane ($\tilde{f}_1$,f_2) on axis $\tilde{f}_2$.

On the third stage we add another property p_j as a third axis and define the position of the objects <u>in a new plane</u> that is built by the axes $\tilde{f}_1$ and p_j (fig. 7). We repeat the same procedure as on the second stage and define this way the third feature $\tilde{f}_2$.

In our simple example in fig. 7 the third feature already leads to a faultless separation of the objects on its axis $\tilde{f}_2$.

Now, let's shortly describe the way of calculation.

As we can see from the description of the idea, the procedure is always the same for each step except the first basic step defining f_0.
Let's assume that we have already selected k-1 properties as features. We will use the symbol $\tilde{x}_{i,(k-1)}, i = 1,...,l$ for the projections of the objects onto the feature $\tilde{f}_{k-1}$ and ω_{k-1} for the cardinality of the set of correctly classified objects.
Performing now the **step *k*** we compute the projection

$$\tilde{x}_{i,(k)} = \rho_i \cos(\beta_i + \alpha_q) \qquad (1)$$

onto the new axis $\tilde{f}_k$ for all remaining properties f_q and for all objects with $\rho_i = \sqrt{\tilde{x}^2_{i,(k-1)} + x^2_{iq}}$ and $\beta_i = \arctan(x_{iq} / \tilde{x}_{i,(k-1)})$. x_{iq} is the value of the property p_q of the object i.
Now we turn the axis $\tilde{f}_k$ by the variable angle $\alpha_q \in [0, \pi]$ and define the optimal angle α_q^* providing the largest number ω_q of correctly classified objects for the optimal threshold f_q^*. When this property p_q satisfies the condition

$$F_{\min}(f_k) = \frac{1}{l}\left(\frac{l - \omega_{k-1}}{n_0 - k + 1}\right) \qquad (2)$$

then we will select the corresponding axis $\tilde{f}_k$ as the next feature. The direction of this feature is given with the angle α_q^* and the projections of the objects onto this feature are defined by $\tilde{x}_{i,(k)} = \rho_i \cos(\beta_i + \alpha_q^*)$.

The execution of the procedure results in a structure consisting of the number of the initial basic property, a set of n-1 pairs of property numbers and corresponding angles (describing the features), and also of the threshold value for the last feature.
With it, the normal vector of the separating hyperplane consists of a sequence of values

$$\left(\prod_{k=2}^{n} \cos\alpha_k^*, -\sin\alpha_2^* \prod_{k=3}^{n} \cos\alpha_k^*, ..., -\sin\alpha_q^* \prod_{k=q+1}^{n} \cos\alpha_k^*, ..., -\sin\alpha_n^*\right) \qquad (3)$$

where the position of the number in the sequence corresponds to the step of the procedure and the value of the bias corresponds to the threshold of the last feature.

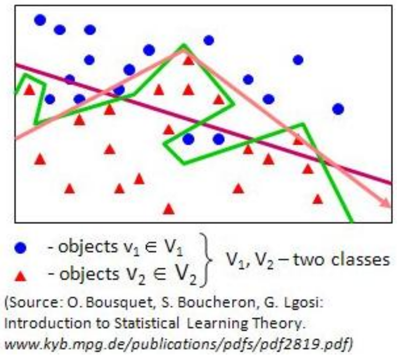
Due to the fact that in (3) the elements of the normal vector are arranged by the induction of the procedure they must be assigned backwards to their properties for practical classification. Non-selected properties will not be used at all.

4. Two Trojan Horses in SVM and α-Procedure

Both the SVM and the α-procedure do not solve the so called “gap problem” of the compromise between the precision of the separation on the training sequence and the stability of the algorithm on new data points or the compromise between fit and complexity (see fig. 8). In the case that the original table with training data leads to an incomplete separation of the classes V_1 and V_2 we can use the rectifying or extended table of properties as described above. Then, the separating hyperplane that we will get in the extended space corresponds to a hypersurface in the original space.

The second Trojan horse is the “outlier problem” where the empirical functional Q_e deviates significantly from the mean square functional Q_m (see fig. 9) This problem was solved by Vapnik with the help of a generalization of the Glivenko-Cantelli Theorem of Uniform Convergence where Vapnik used the C-metric instead of the Lp2-metric. The use of the C-metric limits the effect of the outlier sample up to a certain width of the corridor.

But unfortunately the use of support vectors reduces the robustness because the SVM is very sensitive against chances of these vectors.



(Source: O. Bousquet, S. Boucheron, G. Lgosi: Introduction to Statistical Learning Theory. www.kyb.mpg.de/publications/pdfs/pdf2819.pdf)

Fig.8: The Gap Problem

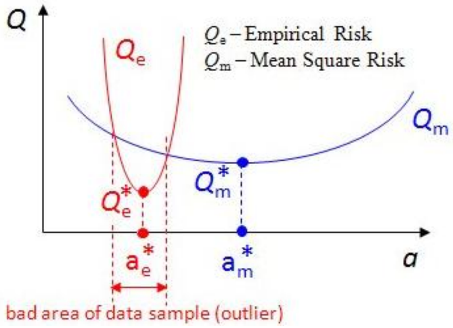


Fig. 9: The Outlier Problem

5. Outlook

One famous idea of Tukey was the introduction of the non-parametric term **data depth** in 1974 which has relieved us from the binding to a concrete distribution function.

After **Tukey** (see Zuo and Serfling [8]) different methods for computing the data depths were proposed. But all corresponding tasks were restricted to a dimension of m=2.

The problem of extending the dimensioning to m > 2 was solved by Mosler and Lange [9] and new opportunities for pattern recognition have been opened.

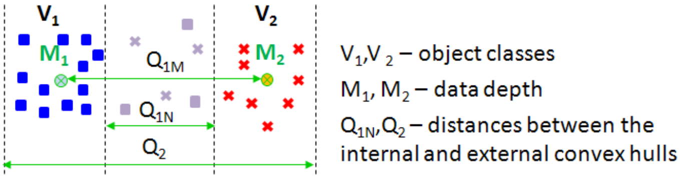
Using the term data depth we can reword the Machalanobis distance and the Novikov distance (see Fig. 10).

The relative Machalanobis distance Q_M is the lower estimation of the quality of the decision rule when the lengh of the data sample is fixed.

The relative Novikov distance Q_N is the upper border of the quality of the decision rule.

$$Q_M = \frac{Q_{1M}}{Q2}$$

$$Q_N = \frac{Q_{1N}}{Q2}$$



V_1
V_2
M_1
M_2
Q_{1M}
Q_{1N}
Q_2
V_1,V_2 – object classes
M_1, M_2 – data depth
Q_{1N},Q_2 – distances between the internal and external convex hulls

Fig. 10: New interpretation of the Machalanobis and Novikov distances

We see that Q_{1N} corresponds to distance $|c_1 - c_2|$. For that reason the term data depth of classes can be applied to the definition of the separating hyperplane as shown in fig 5.

This opens the opportunity for using the reworded definitions of the Machalanobis and Novikov distances for the separation of classes.

References

1. V.I. Vasil'ev. The reduction principle in problems of revealing regularities (in Russian). Cybernetics and System Analysis 5, Part I: 69—81, 2003. Part II: 7—16, 2004
2. V.I. Vasil'ev. Recognition Systems – Reference Book (in Russian). Naykova dumka, Kiev, 1969
3. V. Vapnik and A. Chervonenkis. Theory of Pattern Recognition (in Russian). Nauka, Moscow, 1974.
4. C. Cortes, and V. Vapnik. Support vector networks. Mach. Learn 20, pp. 1—25, 1995.
5. T. Lange. Ein Beitrag zur strukturellen Modellierung bei kleinen Datenmengen unter Anwendung der Methode der gruppenweisen Erfassung der Argumente. PhD Thesis, University of Technology, Ilmenau, Germany, 1983.
6. T. Lange. New Structure Criteria in GMDH. Proceedings of the First US/Japan Conference on the Frontiers of Statistical Modelling, Kluwer Academic Publishers. 1994
7. V.I. Vasil'ev and T. Lange. The principle of duality within the training problem during pattern recognition (in Russian). Cybernetics and Computer Engineering. Kiev, 1998.
8. Y. Zuo Y and R. Serfling General notions of statistical depth function. Ann.Statist. 28, 461 – 482, 2000
9. K. Mosler, T. Lange, P. Bazovkin, Computing zonoid trimmed regions of dimension d > 2. Computational Statistics & Data Analysis. Elsevier Science Publishers B.V, Amsterdam, 2009.
10. T. Lange. Solution Tuning - an attempt to bridge existing methods and to open new ways. Compstat 2010, Paris, August 22-27, 2010.
11. T. Lange and P. Mozharovskyi. Depth determination for multivariate samples (in Russian). Cybernetics and Computer Engineering. Kiev, 2011 (submitted and accepted in 2010)