OPTIMUM STOCHASTIC CLASSIFIER IN MATLAB

J. Kukal [1], *L. Hainc* [1], *D. Majerová* [2]

[1] DCCE FCE ICT Prague
[2] DSEE FNSPE CTU in Prague

Abstract

Any bipolar deterministic classifier is a function $\mathrm{f} : \mathbb{R}^n \to \{-1, +1\}$ **where** n **is an input space dimension. Having** m **patterns, we can evaluate its classification ability in the terms of sensitivity** se **and specificity** sp**. The optimum deterministic classifier is thus a result of multicriteria optimization task** $(se, sp) = \max$**. Using Minkowski distance from ideal alternative** $(1, 1)$**, we obtain compromise optimization task**

$$\left((1 - se)^p + (1 - sp)^p\right)^{1/p} = \max \ \textbf{for}\ p \geq 1$$

with a special form

$$\min(se, sp) = \max \ \textbf{for}\ p \to \infty.$$

Our approach is based on the set of H **fixed classifiers** $\mathrm{f}_1, \mathrm{f}_2, \ldots, \mathrm{f}_H$ **with estimated sensitivities** $se_1, se_2, \ldots, se_H$ **and specificities** $sp_1, sp_2, \ldots, sp_H$**. The stochastic classifier is based on the random selection of classifiers with disjoint probabilities** $q_1, q_2, \ldots, q_H$**. After the application of mean value and convex hull of point set in** $\mathbb{R}^2$**, we recognized that optimum stochastic classifier is based on two fixed deterministic ones at most. Remaining** $H - 2$ **classifiers are ignored. The pair of classifiers** f_u**,** f_v **with selection probabilities** q_u**,** q_v**,** $q_u + q_v = 1$ **is the main result of optimization. Our methodology is an alternative to ROC[1] analysis. Both the individual deterministic classifiers and the selection of classifier pair were realized in the Matlab environment.**

1 Compromise Classifier

Let $m_+, m_- \in \mathbb{N}$ be numbers of positive and negative patterns. Let $n, H \in \mathbb{N}$ be a dimension of pattern space and a number of classifiers. Let $\mathrm{f}_k : \mathbb{R}^n \to \{-1, +1\}$ be a function of k^{th} classifier and $se_k, sp_k \in [0, 1]$ be its sensitivity and its specificity for $k = 1, \ldots, H$. According to MIA[2] method, the best classifier satisfies

$$F_k = (1 - se_k)^p + (1 - sp_k)^p = \min_{1 \leq k \leq H} \quad \text{for } p \geq 1$$

$$\text{or} \qquad F_k = \max(1 - se_k,\ 1 - sp_k) = \min_{1 \leq k \leq H} \quad \text{for } p \to +\infty, \ \text{respectively.}$$

The orientation to the best possible compromise classifier is not the only one approach. It is also possible to organize the optimum stochastic classification with fixed probabilities of classifier selection.

Let $q_k \geq 0$ be a probability of k^{th} classifier selection for $k = 1, \ldots, H$ where $\sum\limits_{k=1}^{H} q_k = 1$. The stochastic classifier begins with the random selection of classification index k^* according to the probabilities $q_1, \ldots, q_H$ in the first step. Then the final decision is given by formula $y = \mathrm{f}_{k^*}(\bar{x})$ with the sensitivity $se^* = \sum\limits_{k=1}^{H} q_k se_k$ and specificity $sp^* = \sum\limits_{k=1}^{H} q_k sp_k$.
Introducing $u_k = 1 - se_k$, $v_k = 1 - sp_k$, $u^* = 1 - se^*$, $v^* = 1 - sp^*$, we obtain

$$u^* = \sum_{k=1}^{H} q_k u_k, \ v^* = \sum_{k=1}^{H} q_k v_k$$

[1] Reciever Operating Characteristic
[2] Minkowski distance from ideal alternative $(1, 1)$

as distances from ideal alternative $se^* = sp^* = 1$.

Using MIA technique for $p \geq 1$ and unknown selection probabilities, we obtain a convex optimization task

$$(u^*)^p + (v^*)^p = \min_{u^*,v^*},$$

$$u^* = \sum_{k=1}^{H} q_k u_k,$$

$$v^* = \sum_{k=1}^{H} q_k v_k,$$

$$\sum_{k=1}^{H} q_k = 1.$$

$$q_k \geq 0 \text{ for } k = 1, \ldots, H$$

It is easy to recognize than the task can be rewritten to more clear form

$$(u^*)^p + (v^*)^p = \min_{u^*,v^*},$$

$$u^*, v^* \in \mathrm{CH}(\mathbf{S})$$

where $\mathrm{CH}(\mathbf{S})$ is a convex hull of point set $\mathbf{S} = \{(1 - se_1,\ 1 - sp_1), \ldots, (1 - se_H,\ 1 - sp_H)\}$.

The special case $p \to +\infty$ comes to

$$\max(u^*, v^*) = \min,$$

$$(u^*, v^*) \in \mathrm{CH}(\mathbf{S}).$$

When $(0, 0) \in \mathbf{S}$ then at least one classifier is perfect, and thus selected with probability $q_k = 1$. But in the opposite case, the minimum of the convex objective function stands outside the convex hull $\mathrm{CH}(\mathbf{S})$ and the constrained minimum exists in a boundary point of the convex hull. The boundary of $\mathrm{CH}(\mathbf{S})$ is a convex polygon which can be systematically decomposed to vertices and edges. Thus, the optimum classifier corresponds to the single vertex or to the interval point of any edge which is a stochastic compromise between two vertices or classifiers, respectively.

Let $\mathrm{F}^*(u^*, v^*) = \begin{cases} (u^*)^p + (v^*)^p & \text{for } p \geq 1 \\ \max(u^*, v^*) & \text{for } p \to +\infty \end{cases}$. Then the algorithm of optimum classification consists of four steps:

1. We define $\mathbf{S} = \big\{(1 - se_k,\ 1 - sp_k)\ \big|\ k = 1, \ldots, H\big\}$.
2. We define $\mathbf{V} \subset \mathbf{S}$ as the set of convex hull vertices.
3. For every pair of neighborhood vertices $\mathbf{r}_1, \mathbf{r}_2 \in \mathbf{V}$ where $\mathbf{r}_1 = (1 - se_{k_1},\ 1 - sp_{k_1})$, $\mathbf{r}_2 = (1 - se_{k_2},\ 1 - sp_{k_2})$, we solve the optimization task

$$\mathrm{F}^*(u^*, v^*) = \min,$$

$$u^* = q_{k_1}(1 - se_{k_1}) + q_{k_2}(1 - se_{k_2}),$$

$$v^* = q_{k_1}(1 - sp_{k_1}) + q_{k_2}(1 - sp_{k_2}),$$

$$q_{k_1}, q_{k_2} \geq 0,\ q_{k_1} + q_{k_2} = 1.$$

4. Find the best pair of vertices.

The resulting values $k_1, k_2, q_{k_1}, q_{k_2}$ enables to realize optimum stochastic classifier, which is based on two deterministic classifiers at most. There are three possibilities:

(i) $q_{k_1} = 1$. Then the optimal classifier is $y = \mathrm{f}_{k_1}(\bar{x})$.

(ii) $q_{k_2} = 1$. Then the optimal classifier is $y = \mathrm{f}_{k_2}(\bar{x})$.

(iii) $q_{k_1}, q_{k_2} > 0$. Then the optimal classifier is $y = \begin{cases} \mathrm{f}_{k_1}(\bar{x}) & \text{with the probability } q_{k_1} \\ \mathrm{f}_{k_2}(\bar{x}) & \text{with the probability } q_{k_2} \end{cases}.$

2 Generation of Individual Classifier

There are many possibilities how to generate various classifiers for given sets of positive and negative patterns:

(i) Forming alternative pattern sets randomly (Bootstrap, any random selection).

(ii) Learning OLAM with various regularization parameter $\lambda > 0$.

(iii) Learning MLP, MLL, RBF with various initial estimates of weights.

(iv) Variation of given weight from optimum solution (w_0 at least).

(v) Variation of ANN structure (setting several weights to zero in the process of learning).

Example: $m_+ = 5$, $m_- = 4$, $H = 12$.

Critical points (vertices of CH)

k	se_k	sp_k
1	0.4	1
2	0.8	0.5
3	1	0
4	0.8	0
5	0	1

Single point classifier

p	se_k	sp_k
1	0.4	1
2	0.8	0.5
∞	1	0

Pair of classifiers

p	se^*	sp^*	q_1	q_2
1	2/5	1	1	0
2	26/41	29/41	17/41	24/41
∞	2/3	2/3	1/3	2/3

3 Source Codes in the MATLAB Environment

The library of general functions was created in the Matlab environment. The main function `STCLAMIA` is a general tool for multicriteria classification when $p \in [1, +\infty) \cup \{+\infty\}$. This function calls `fmincon` and `STCLAMIAOBJ` functions in the case of $p \in (1, +\infty)$. Here, `STCLAMIAOBJ` is only a convex objective function for constrained minimization. There are two useful functions: `STCLAAIA` which is called for $p = 1$, and `STCLAIIA` which is called for $p \to +\infty$. The main function `STCLAMIA` calls the previous functions, collects the results and depicts them in the ROC diagram (see the Fig. 1). The source codes are demonstrated in the Figs. 2–5.

4 Conclusions

A new stochastic method of binary classification was developed to improve the sensitivity and specificity of classification process. Applying the theory of multicriteria decision making, the resulting optimum system consists of two deterministic classifiers, which are taken at random. The supporting library was created and tested in the Matlab environment.

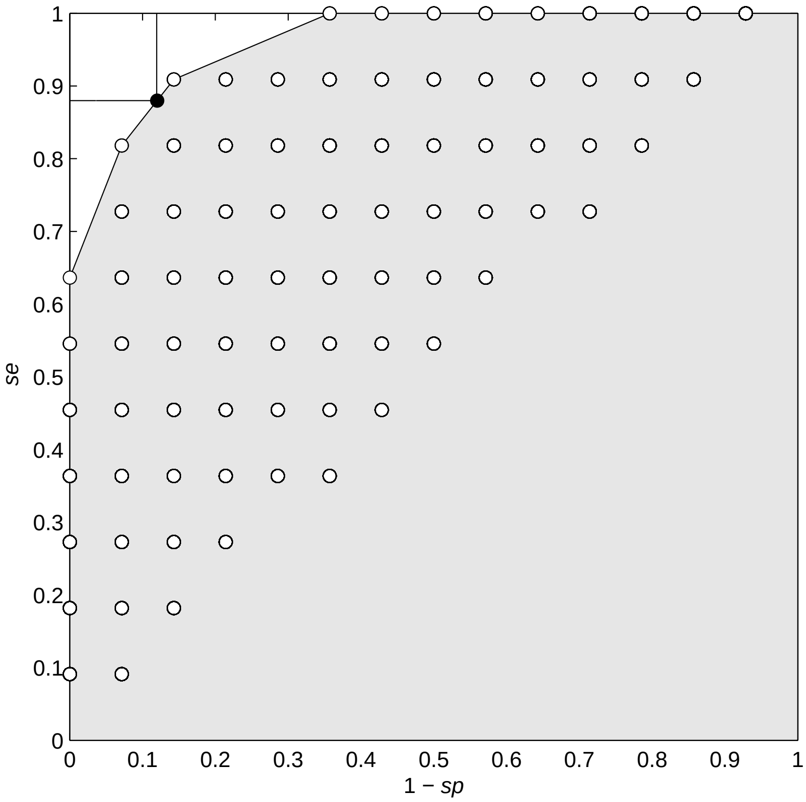

Figure 1: Convex hull of ROC space and optimum classifier for $p \to \infty$

```
function [Fopt,seopt,spopt,Wopt,qopt]=STCLAAIA(se,sp,W)
n=length(se); Fopt=1; kopt=1;
for k=1:n
    F=1-se(k)+1-sp(k);
    if F<Fopt
        Fopt=F; kopt=k;
    end
end
seopt=se(kopt); spopt=sp(kopt); Wopt=W(kopt,:); qopt=1;
```

Figure 2: Source code of `STCLAAIA.m`

```
function [Fopt,seopt,spopt,Wopt,qopt]=STCLAMIA(se,sp,p,W)
global STCLA_SE STCLA_SP STCLA_P
sele=se+sp>1; se=se(sele); sp=sp(sele);
sextd=[1;0;0;se]; spxtd=[0;1;0;sp]; nw=size(W,2);
Wxtd=[+ones(1,nw)*inf; -ones(1,nw)*inf; ones(1,nw)*NaN; W(sele,:)];
k = convhull(sextd,spxtd); sextd=sextd(k); spxtd=spxtd(k);
STCLA_SE=sextd; STCLA_SP=spxtd; STCLA_P=p;
W=Wxtd(k,:); n=length(k);
if p==inf
    [Fopt,seopt,spopt,Wopt,qopt]=STCLAIIA(sextd,spxtd,W);
end
if p==1
    [Fopt,seopt,spopt,Wopt,qopt]=STCLAAIA(sextd,spxtd,W);
end
if p>1 & p<inf
    options = optimset('LargeScale','off','Display','off','TolX',1e-5);
    xopt = fmincon('STCLAMIAOBJ',zeros(n,1),[],[],ones(1,n),1,...
           zeros(n,1),ones(n,1),[],options);
    [Fopt,seopt,spopt]=STCLAMIAOBJ(xopt);
    selector=xopt>1e-6;
    Wopt=W(selector,:); qopt=xopt(selector);
end

fill(1-spxtd,sextd,[0.9 0.9 0.9]); hold on
phi=0:pi/2000:pi/2;
if p<inf
    r=Fopt./(cos(phi).^p+sin(phi).^p).^(1/p);
else
    r=Fopt./max(cos(phi),sin(phi));
end
xxx=r.*sin(phi); yyy=1-r.*cos(phi); plot(xxx,yyy,'k-')
plot(1-sp,se,'ko','markerfacecolor',[1 1 1])
plot(1-spopt,seopt,'ko','markerfacecolor',[0 0 0])
axis equal; axis([0 1 0 1]);
xlabel('1 -{\it sp}');ylabel('{\it se}'); hold off
```

Figure 3: Source code of `STCLAMIA.m`

```
function [F,se,sp]=STCLAMIAOBJ(x)
global STCLA_SE STCLA_SP STCLA_P
u=(1-STCLA_SE)'*x; v=(1-STCLA_SP)'*x;
p=STCLA_P; se=1-u; sp=1-v;
F=(u^p+v^p)^(1/p);
```

Figure 4: Source code of `STCLAMIAOBJ.m`

```
function [Fopt,seopt,spopt,Wopt,qopt]=STCLAIIA(se,sp,W)
n=length(se); seopt=0; kopt=1; lambdaopt=0;
for k=1:n-1
    d1=se(k)-sp(k); d2=se(k+1)-sp(k+1); den=d1-d2;
    if den~=0
        lambda=d1/den;
        if lambda>=0 & lambda<=1
            seast=se(k)+(se(k+1)-se(k))*lambda;
            if seast>seopt
                seopt=seast; kopt=k; lambdaopt=lambda;
            end
        end
    end
end
spopt=seopt; Fopt=1-seopt;
Wopt=W([kopt kopt+1],:);
qopt=[1-lambdaopt; lambdaopt];
selector=qopt>1e-6;
Wopt=Wopt(selector,:);
qopt=qopt(selector);
```

Figure 5: Source code of `STCLAIIA.m`

Acknowledgments

The work has been supported by the fund No. MSM 6046137306 of Ministry of Education of the Czech Republic. This support is very gratefully acknowledged.

References

[1] K.F. Riley, M.P. Hobson, S.J. Bence. *Mathematical Methods for Physics and Engineering.* Cambridge University Press, Cambridge, 1992.

[2] M. Meloun, J. Militký. *Statistické zpracování experimentálních dat.* PLUS, Praha, 1994.

[3] J. Anděl. *Matematická statistika.* SNTL, Praha, 1978.

[4] L. Fausett. *Fundamentals of Neural Networks: Architectures, Algorithms and Applications.* Prentice Hall, New Jersey, 1994.

Jaromír Kukal, Luboš Hainc
Institute of Chemical Technology, Prague
Department of Computing and Control Engineering
Jaromir.Kukal@vscht.cz

Dana Majerová
Czech Technical University in Prague
Department of Software Engineering in Economy
Dana.Majerova@dc.fjfi.cvut.cz