Comparison of Various Definitions of Proximity in Mixture Estimation

Ivan Nagy¹,², Evgenia Suzdaleva¹ and Pavla Pecherková¹,²

¹Department of Signal Processing, The Institute of Information Theory and Automation of the Czech Academy of Sciences, Pod vodárenskou věží 4, 18208, Prague, Czech Republic
²Faculty of Transportation Sciences, Czech Technical University, Na Florenci 25, 11000, Prague, Czech Republic
{nagy, suzdalev, pecherkova}@utia.cas.cz

Keywords: Classification, Recursive Mixture Estimation, Proximity, Bayesian Methods, Mixture based Clustering.

Abstract: Classification is one of the frequently demanded tasks in data analysis. There exists a series of approaches in this area. This paper is oriented towards classification using the mixture model estimation, which is based on detection of density clusters in the data space and fitting the component models to them. A chosen function of proximity of the actually measured data to individual mixture components and the component shape play a significant role in solving the mixture-based classification task. This paper considers definitions of the proximity for several types of distributions describing the mixture components and compares their properties with respect to speed and quality of the resulting estimation interpreted as a classification task. Normal, exponential and uniform distributions as the most important models used for describing both Gaussian and non-Gaussian data are considered. Illustrative experiments with results of the comparison are provided.

1 INTRODUCTION

Mixture models are a powerful class of models used for description of multi-modal systems, which work in different switching regimes. Detection of an active regime is a significant task in various application fields such as, for example, fault detection, diagnostics systems, medicine, big data related problems, etc., see, e.g., (Yu, 2012a; Yu, 2011; Yu, 2012b).

This task is often solved within (supervised) classification issues, which distribute data observed on the described system among some categories, using the training set of already classified data. A range of such methods is rather wide and includes, e.g., decision trees, Bayesian classifiers, rule-based methods, including fuzzy rules, neural networks, k-nearest-neighbor classifiers, genetic algorithms and other techniques. The detailed overview of these numerical techniques can be found, e.g., in sources (Larose, 2005; Han et al., 2011; Zaki and Meira, 2014; Calders and Verwer, 2010; Zhang, 2000; Ishibuchi et al., 2000), etc.

Unlike them, the unsupervised classification methods are directly based on clustering solutions looking for data clusters in the untrained data set, such as hierarchical and partitioning methods (centroid-, density-based) and many others, see an overview, in e.g., (Berkhin, 2006; Jaín, 2010; Ester et al., 1996).

A separate group of such methods is created by model-based clustering and classification algorithms presented e.g., by (Bouveyron and Brunet-Saumard, 2014; Zeng and Cheung, 2014; Ng and McLachlan, 2014), etc. These methods are more complex and demanding.

Specific methods of this type of classification approaches are based on estimation of mixture models. The mixture model consists of components in the form of probability density functions (pdfs) describing individual system regimes and a model of their switching. The mixture-based classification starts from some pre-specified (mostly resulted from the initial data analysis) locations of components and performs a search for density clusters in the data space with the aim of fitting component models to data. The search for the data clusters is one of the most critical parts of the mixture estimation algorithms.

Fundamental approaches in this area mostly focus on: (i) the use of the EM algorithm (Gupta and Chen, 2011), see, e.g., (Boldea and Magnus, 2009; Wang et al., 2004); (ii) Variational Bayes methods (McCormy and Titterington, 2009; Smid and Quinn, 2006); (iii) Markov Chain Monte Carlo (MCMC) methods (Frühwirth-Schnatter, 2006; Doucet and Andrieu, 2001; Chen and Liu, 2000), often used for mixtures of state-space models; and (iv) recursive Bayesian estimation algorithms (Kárný et al., 1998;
2 MODELS

Let’s consider a multi-modal system, which at each discrete time instant \( t = 1, 2, \ldots \) generates the continuous data vector \( y_t \). It is assumed that the observed system works in \( m_c \) working modes, each of them is indicated at the time instant \( t \) by the value of the unmeasured dynamic discrete variable \( c_t \in \{1, 2, \ldots , m_c\} \), which is called the pointer (Kárný et al., 1998).

The observed system is supposed to be described by a mixture model, which (in this paper) consists of \( m_c \) components in the form of the following pdfs

\[
f(y_t; \Theta, c_t = i), \; i \in \{1, 2, \ldots , m_c\},
\]

where \( \Theta = \{\Theta_i\}_{i=1}^{m_c} \) is a collection of unknown parameters of all components, and \( \Theta_i \) includes parameters of the \( i \)-th component in the sense that \( f(y_t; \Theta, c_t = i) = f(y_t; \Theta_i) \) for \( c_t = i \).

The component, which describes data generated by the system at the time instant \( t \) is said to be active.

2.1 Dynamic Pointer Model

Switching the active components (1) is described by the dynamic model of the pointer

\[
f(c_t = i|c_{t-1} = j, \alpha), \; i, j \in \{1, 2, \ldots , m_c\},
\]

which is represented by the transition table

\[
\begin{array}{cccc}
  c_{t-1} = 1 & c_t = 1 & \cdots & c_t = m_c \\
  \alpha_{1|1} & \alpha_{2|1} & \cdots & \alpha_{m_c|1} \\
  c_{t-1} = 2 & \alpha_{1|2} & \cdots & \cdots \\
  \cdots & \cdots & \cdots & \cdots \\
  c_{t-1} = m_c & \alpha_{1|m_c} & \cdots & \cdots
\end{array}
\]

where the unknown parameter \( \alpha \) is the \((m_c \times m_c)\)-dimensional matrix, and its entries \( \alpha_{i,j} \) are non-negative probabilities of the pointer \( c_t = i \) (expressing that the \( i \)-th component is active at time \( t \) under condition that the previous pointer \( c_{t-1} = j \). According to (Kárný et al., 2006), the parameter \( \alpha \) of the pointer model (2) is estimated using the conjugate prior Dirichlet pdf in the Bayes rule, recomputing its initially chosen statistics and its normalizing.

Depending on the nature of the measured data, the pdfs of components can be specified as follows.

2.2 Normal Components

Under assumption of normality of measurements, the pdf (1) can be specified as

\[
(2\pi)^{-N/2}|r_i|^{-1/2} \exp \left\{ -\frac{1}{2} |y_t - \theta_i|^2 r_i^{-1} |y_t - \theta_i| \right\},
\]

where \( N \) denotes a dimension of the vector \( y_t \); for each \( i \in \{1, 2, \ldots , m_c\} \) the parameter \( \theta_i \) represents the center of the \( i \)-th component; \( r_i \) is the covariance matrix of the involved normal noise, which defines the shape of the component, and here \( \{\theta_i, r_i\} \equiv \Theta_i \).

The point estimates of parameters \( \theta_i \) and \( r_i \) of individual components are used in computing the proximity (this will be explained later). Necessary basic facts about calculation of the point estimates are briefly given below.

According to (Peterka, 1981; Kárný et al., 1998), parameters \( \theta_i \) and \( r_i \) of the individual \( i \)-th component (3) (omitting here for simplicity the subscript \( i \)) are estimated via the Bayes rule using the conjugate prior Gauss-inverse-Wishart pdf with the reproducible (initially chosen) statistics \( V_{i-1} \) and \( k_i \) of the appropriate dimensions. The statistics are updated as follows:

\[
V_t = V_{t-1} + \begin{bmatrix} y_t \\ 1 \end{bmatrix} [y_t, 1],
\]

\[
k_t = k_{t-1} + 1,
\]
and the point estimates of parameters are respectively

\[ \hat{\theta}_t = V_1^{-1}V_y, \quad \hat{r}_t = \frac{V_{yy} - V_y'V_1^{-1}V_y}{\kappa_t} \]  

(6)

with the help of partition \( V_1 = \begin{bmatrix} V_{yy} & V_y' \\ V_y & V_1 \end{bmatrix} \),

where \( V_{yy} \) is the square matrix of the dimension \((N \times N)\), \( V_y' \) is \( N \)-dimensional column vector and \( V_1 \) is scalar, see details in (Peterka, 1981).

### 2.3 Exponential Components

The exponential distribution of components (1), which is often suitable for situations, where the assumption of normality brings a series of limitations (non-negative, bounded data, etc.) can be specified as

\[ \exp \left\{ -a'_t(y_t - b_t) \right\}, \]

(8)
i.e., here \( \{a_t,b_t\} \equiv \Theta_t \), and \( (a_t) > 0 \) and \( (b_t) \in R \) are the \( l \)-th entries of the \( N \)-dimensional vectors \( a_t \) and \( b_t \) respectively with \( l = \{1,2,\ldots,N\} \). Currently the independence of entries of the vector \( y_t \) is assumed.

Basic facts about the recursive estimation of the parameters \( a_t \) and \( b_t \), which are necessary to obtain the proximity (explained later) are recalled below. With the help of the Bayes rule and the prior exponential pdf, the parameters of the individual \( i \)-th exponential component (8) are obtained, using the algebraic update of the initially chosen statistics \( S_{t-1} \), \( K_{t-1} \) and \( B_{i-1} \). It is done at each time instant \( t \) as follows (omitting here the subscript \( i \) for simplicity):

\[ S_t = S_{t-1} + y_t, \]
\[ K_t = K_{t-1} + 1, \]
\[ B_{i,t} = B_{i,t-1} - y_{i,t}, \]

(9)  

(10)  

(11)

where, denoting \( \delta_t \) for the \( l \)-th entry \( y_{i,t} \) of \( y_t \),

\[ \begin{cases} \delta_t > 0, & B_{i,t} = B_{i,t-1} - \delta_t; \\ else & B_{i,t} = B_{i,t} + \epsilon, \end{cases} \]

(12)  

(13)

where \( B_{i,t} \) is the \( l \)-th entry of the statistics \( B_t \), and \( \epsilon \) can be taken as 0.1. The point estimates of parameters are obtained from

\[ \hat{a}_{i,t} = \frac{K_t}{S_{i,t} - K_t B_{i,t}}, \]
\[ \hat{b}_{i,t} = B_{i,t}, \]

\[ \hat{a}_{t} = \frac{K_t}{S_{i,t} - K_t B_{i,t}}, \]
\[ \hat{b}_{t} = B_{i,t}, \]

where \( \hat{a}_{i,t} \) and \( \hat{b}_{i,t} \) are entries of vectors \( \hat{a}_t \) and \( \hat{b}_t \) respectively, see, e.g., (Yang et al., 2013), and \( S_{i,t} \) is the \( l \)-th entry of the statistics \( S_t \).

### 2.4 Uniform Components

The pdf (1) with the uniform distribution often serves as an appropriate tool for description of bounded data. In this paper the independence of entries of the vector \( y_t \) is assumed, and therefore, \( \forall i \in \{1,2,\ldots,m_l\} \) the pdf (1) takes the form

\[ f(y_i|L,R,c_i = i) = \begin{cases} \frac{1}{R_i-L_i} & \text{for } y_i \in (L_i,R_i), \\ 0 & \text{otherwise}, \end{cases} \]

(16)
i.e., here \( \{L_i,R_i\} \equiv \Theta_i \), and their entries \( (L_i) \) and \( (R_i) \) are respectively minimal and maximal bounds of the \( l \)-th entry \( y_{i,t} \) of the vector \( y_t \) for the \( i \)-th uniform component.

The estimation of parameters of the individual \( i \)-th uniform component (16) in the case of independent data entries is performed using the initially chosen statistics \( L_{t-1} \) and \( R_{t-1} \) with the update of their \( l \)-th entries for each \( i \in \{1,\ldots,N\} \) in the following form, see, e.g., (Casella and Berger, 2001):

\[ \begin{cases} L_{t,i} = L_{t-1,i}, & \text{if } y_{i,t} < L_{t,i-1}, \\ \text{else } & R_{t,i} = R_{t,i-1}, \end{cases} \]

(17)  

(18)

where the subscript \( i \) is omitted for simplicity. The point estimates of parameters are computed via

\[ \hat{L}_t = L_t, \quad \hat{R}_t = \hat{R}_t. \]

(19)

### 3 PROXIMITY DEFINITIONS

#### 3.1 General Estimation Algorithm

The mixture estimation algorithm is derived using the joint pdf for all variables to be estimated, i.e., \( \Theta \), \( \alpha \) and \( c_i \), and its gradual marginalization over all of them. In general, based on (Kárny et al., 1998; Kárný et al., 2006; Peterka, 1981; Nagy et al., 2011), after initialization, the recursive mixture estimation algorithm includes the following steps at each time instant \( t \) (here they are given only to help a reader to be oriented in the discussed field):

1. Measure new data.
2. Compute the proximity of the current data item to each component. It is done by substituting the measured data item and the point parameter estimates from the previous time instant into corresponding components.
3. Construct the weighting vector containing the probabilities of the activity of components at the actual time instant. It is obtained by using the obtained proximities, the prior pointer pdf and the previous point estimate of \( \alpha \).
4. Update the statistics of all components using the weighting vector, and statistics of the pointer model, using the weighting matrix joint for the current pointer $c_t$ and the previous one $c_{t-1}$.

5. Recompute the point estimates of all parameters and then use them as the initial ones in the first step of the on-line algorithm.

More detailed information can also be found in (Suzdaleva et al., 2015).

3.2 Proximity as the Approximated Likelihood

The proximity appeared in Step 2 of the above algorithm is explained in this section. Originally, according to (Kárny et al., 2006) the proximity has been introduced for the normal distribution as the likelihood for different variants of the model – as the so-called v-likelihood. In the considered context the variants of the normal distribution are given by normal components (3) labeled by the pointer value $c_t$. Generally the proximity is derived based on the following scheme. For each $i \in \{1, 2, \ldots, m_c\}$ under assumption of the mutual independence of $\Theta$ and $\alpha$, and $y_t$ and $\alpha$, and $c_t$ and $\Theta$ it takes the form

$$f(\Theta, c_t = i, c_{t-1} = j, \alpha|y(t)) \propto f(y_t|\Theta, c_t = i, c_{t-1} = j, \alpha|y(t-1))$$

via the chain rule and Bayes rule

$$= f(y_t|\Theta, c_t = i) f(\Theta|y(t-1))$$

The speed of decreasing should be fast enough. That’s why it is advantageous to apply the proximity function with a curve of the normal distribution.

Other functions also have similar properties and thus can be explored as the proximity. Generally a power function can decrease in the beginning of its course faster than an exponential function, but at a larger distance the exponential function is closer to zero. Comparison of these functions as the proximity can unambiguously assign the measured data item (if possible) to one of the components. Since components in the data space can be of a different shape and located on a different distance from each other, and also with a variously decreasing density of data items on their edges, it is important to choose a suitable distance measure. This is determined by the chosen definition of the proximity.

The main requirement for the proximity function is to be at its maximum at the center of a component and to decrease rapidly with the increasing distance. The speed of decreasing should be fast enough. That’s why it is advantageous to apply the proximity function with a curve of the normal distribution.

Based on this idea, different definitions of the proximity can be chosen using a rapidly decreasing function depending on

$$\Delta_t = y_t - E[y_t], \quad (23)$$

where the expectation $E[y_t]$ is computed for the corresponding type of components.

For the $i$-th normal component (3) the expectation is given by the point estimate of $\Theta_i$ according to (6).
For the $i$-th exponential component the expectation can be obtained as
\[
\frac{1}{(\hat{a}_{lt-1})_i} + (\hat{b}_{lt-1})_i
\]
(24)
using the point estimates (14)–(15) for the $l$-th entries of the vector $y_i$.

For the $i$-th uniform component the expectation is a simple average of the component bounds $L_i$ and $R_i$ for corresponding entries of the vector $y_i$.

Using $\Delta_e$ from (23) the following rapidly decreasing functions (as combinations of the mentioned above) are considered:

1. The polynomial of the form
\[
1/\Delta_e^5. \tag{25}
\]
2. The normal approximation of other distributions, optimal in the sense of the Kullback-Leibler divergence, see, e.g., (Kárny et al., 2006), which is the pdf (3) with the substituted expectation of the corresponding type of the components instead of $\theta$. The covariance matrices can be either used from individual distributions or chosen as the diagonal ones in case only the expectations should be estimated.
3. The polynomial in the form
\[
\exp\{-2\Delta_e^5\}. \tag{26}\]
4. The Student distribution based relation
\[
\left(1 + \frac{\Delta_e^2}{t}\right)^{-\frac{1}{2}}. \tag{27}\]

3.4 Algorithm

The mixture estimation algorithm tailored to the above proximity definitions is summarized as follows.

Initialization (for $t = 1$)
- Set the number of components $m_c$.
- Set the initial statistics for the corresponding type of components, i.e., either for (3) or (8), or (16), see Section 2.
- Set the initial Dirichlet statistics of the pointer model (2) according to (Kárny et al., 2006) as the $(m_c \times m_c)$-dimensional matrix denoted by $\gamma_0$.
- Using the initial statistics, compute the initial point estimates of parameters of all components using either (6) or (14)–(15), or (19).
- Compute the initial point estimate of $\alpha$ by normalizing the Dirichlet statistics $\gamma_0$.
- Set the initial $m_c$-dimensional weighting vector $w_0$.

On-line part (for $t = 2, \ldots$)
1. Measure the new data item $y_t$.
2. Compute the expectations $E[y_t]$ of all components using the corresponding point estimates.
3. For all components, obtain (23) by substituting $y_t$ and the expectations $E[y_t]$.
4. For all components, substitute the obtained $\Delta_e$ into the chosen type of the proximity, i.e., either (25), (3), (26) or (27).
5. In each case the result is the $m_c$-dimensional vector of proximities, which has one entry for each component. These entries are proportional to the inverse distance of the data item to the individual components. The higher the number is, the closer the data item is to the component.
6. Multiply entry-wise the resulted vector from the previous step, the previous weighting vector $w_{t-1}$ and the previous point estimate of $\alpha$. The result of the multiplication is the matrix of weights denoted by $W_{t,j2}$ joint for $c_t$ and $c_{t-1}$.
7. Perform the summation of the normalized matrix from the previous step over rows and obtain the updated weighting vector $w_t$. The maximal entry of the weighting vector gives the point estimate of the pointer $c_t$, i.e., the currently active component.
8. For all components, update the statistics with the help of multiplying the data-dependent increment in each statistics by the corresponding entry from the actualized weighting vector $w_t$, see (Kárny et al., 1998; Kárny et al., 2006; Peterka, 1981; Nagy et al., 2011; Suždaleva et al., 2015; Nagy et al., 2016).
9. Update the statistics of the pointer model as
\[
\gamma_{i,j2} = \gamma_{i,j2-1} + W_{i,j2} \tag{28}\]
based on (Kárny et al., 1998; Nagy et al., 2011).
10. Recompute the point estimates of all parameters and use them for Step 1 of the on-line part.

4 EXPERIMENTS

This section provides the experimental comparison of evolution of the proximities (25), (3), (26) and (27) for all three discussed types of components according to the above algorithm. The simulated two-dimensional data were used, which is quite sufficient for adequate judging the proximity evolution. Three components of each type were simulated. A series of experiments with the various number of data was performed. Here typical results are demonstrated.
The quality of classification as the detection of the active component at each time instant according to the maximal weights was compared. Firstly, easily distinguishable components as a simple case were tested. Results for them are provided in Table 1. Then, a case with two components closer to each other and the third at a larger distance was tested, see Table 2. Finally, the most complicated case with overlapped components was tested, see results in Table 3.

Table 1: Average number of incorrect pointer point estimates (%) for easily distinguishable components.

<table>
<thead>
<tr>
<th>Proximities / components</th>
<th>(3)</th>
<th>(8)</th>
<th>(16)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/\Delta^5_t$</td>
<td>1.2</td>
<td>0.0</td>
<td>3.4</td>
</tr>
<tr>
<td>Approximation (3)</td>
<td>0.655</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>$\exp{-2(2\Delta)^5}$</td>
<td>0</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>$(1 + \frac{\Delta^5_t}{t})^{-\frac{1}{2}}$</td>
<td>26.4</td>
<td>0.4</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Average number of incorrect pointer point estimates (%) for variously located components.

<table>
<thead>
<tr>
<th>Proximities / components</th>
<th>(3)</th>
<th>(8)</th>
<th>(16)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/\Delta^5_t$</td>
<td>7.4</td>
<td>6.6</td>
<td>15.8</td>
</tr>
<tr>
<td>Approximation (3)</td>
<td>27.2</td>
<td>67.4</td>
<td>11.2</td>
</tr>
<tr>
<td>$\exp{-2(2\Delta)^5}$</td>
<td>15.8</td>
<td>3.6</td>
<td>11.2</td>
</tr>
<tr>
<td>$(1 + \frac{\Delta^5_t}{t})^{-\frac{1}{2}}$</td>
<td>21.4</td>
<td>4.6</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 3: Average number of incorrect pointer point estimates (%) for overlapped components.

<table>
<thead>
<tr>
<th>Proximities / components</th>
<th>(3)</th>
<th>(8)</th>
<th>(16)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/\Delta^5_t$</td>
<td>39.4</td>
<td>39.2</td>
<td>19.2</td>
</tr>
<tr>
<td>Approximation (3)</td>
<td>37.8</td>
<td>67.4</td>
<td>15.4</td>
</tr>
<tr>
<td>$\exp{-2(2\Delta)^5}$</td>
<td>39.6</td>
<td>33.4</td>
<td>15.4</td>
</tr>
<tr>
<td>$(1 + \frac{\Delta^5_t}{t})^{-\frac{1}{2}}$</td>
<td>39.2</td>
<td>37.2</td>
<td>16</td>
</tr>
</tbody>
</table>

These tables briefly (due to a lack of space) present results of comparison. Nevertheless, to summarize results obtained for variously noised data (that determines distances among components) the following trend is observed.

For easily distinguishable components all the proximity definitions are similarly successful with the insignificant difference, excepting the Student distribution based function (27) for normal components, and the normal approximation of the exponential distributions.

For strongly overlapped components (which is the most complicated and also realistic case) the situation changes. For normal components all results are similar, but now the proximity definition (3) is the most successful.

It seems that a decision about a choice of the proximity definition is suitable to make in accordance to the noise covariance matrix estimation. For the less noised data the distance-based proximities $1/\Delta^5_t$ and $\exp\{-2(2\Delta)^5\}$ can be used. The normal approximation (3) is not suitable for exponential components, but it can be used for uniformed components. For more noised data the normal approximation (3) can be used both for normal and uniform components.

The graphical presentation of the classification results is given for normal components in Figure 1, where (due to a lack of space) the most successful combinations of the proximity definition and a distance among components are shown. In all figures three detected components are sharply visible. Selected results for exponential components can be seen in Figure 2, where the evolution of the component centers during the on-line estimation is plotted for the proximity (26) (top) and (27) (bottom). The
initial centers are denoted by the circle, the final – by ‘x’. The figures show that although the success of the estimation according to Table 1 were the same for the proximity (26) and (27), the stabilization of component centers took a different time. In the top figure the centers were correctly located just in the beginning of estimation, while in the bottom figure the search for correct locations was performed a bit longer.

For uniform components the classification results were close to each other for all tested types of location of components. It is worth noting that the normal approximation (3) and the distance-based proximity (26) were similarly successful. Results with one of them for variously located components are demonstrated in Figure 3 (top).

The proximity evolution during the chosen time interval is shown in Figure 3 (bottom). The distance-based proximity $1/\Delta t^5$ demonstrates the strongest sharpness. The normal approximation has a smoother course. The third proximity from Table 1 is influenced by the constant, which allows to move its course as necessary. The Student distribution based proximity is close to flat.

5 CONCLUSIONS

The aim of the described recursive classification is the analysis of real data in real time, which suggests that the detected clusters can be of various shapes. Thus solutions with mixtures of different components are highly desired.

Using the point estimates directly in the exponential and uniform models as the proximity gives unsuccessful results during the mixture estimation, which can be explained by asymmetric distributions. One of the attempts was the use of the normal approximation optimal from the point of view of the Kullback-Leibler divergence. The important property of the proximity is its rapidly decreasing course. This is satisfied by the normal approximation, but other functions can be also relevant for the considered task. The results of the comparison show that for different types of components a choice of the proximity is not unambiguous and influences the classification results. Thus, the significance of the proximity choice is confirmed.

However, there is still a series of open problems in this area, including (i) the solution for dependent entries of the data vector for exponential and uniform distributions, (ii) extension of the recursive mixture estimation theory for other distributions describing non-Gaussian data, and (iii) classification with a mixture of different distributions. This will be the plan of the future work within the present project.

ACKNOWLEDGEMENTS

The paper was supported by project GAČR GA15-03564S.
REFERENCES


