Global Explanations with Decision Rules:
a Co-learning Approach (supplementary material)

Géraldin Nanfack\(^1\)  
Paul Temple\(^1\)  
Benoît Frénay\(^1\)

\(^1\)PReCISE Research Center, Namur Digital Institute (NADI), University of Namur, Belgium.

A GRADIENT OF THE EXPECTED LOG-LIKELIHOOD

This section reports the gradient of the expected log-likelihood of STruGMA w.r.t. its parameters \(\beta\).

We remind that when omitting class conditioning \(c\), the expected log-likelihood is:

\[
Q(\beta, \beta') = \sum_n \sum_k r_{nk} \log \pi_k + \sum_n \sum_k r_{nk} \left[ \log \mathcal{N}(x_n; \mu_k, \Sigma_k) + \sum_d \log \sigma_d \left( x_{nd} - \alpha_{kd}^{(1)} \right) \right. \\
+ \log \left( 1 - \sigma_d \left( x_{nd} - \alpha_{kd}^{(2)} \right) \right) - \sum_n \sum_k r_{nk} \log \int \alpha_k^{(1)} \mathcal{N}(x; \mu_k, \Sigma_k) dx.
\]

We consider the case where \(\Sigma_k\) is diagonal i.e. \(\Sigma_k = \begin{bmatrix} \sigma_{k1}^2 & 0 & \ldots & 0 \\ 0 & \sigma_{k2}^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_{kd}^2 \end{bmatrix}\). To avoid any confusion, the \(\sigma_{kd}\) are different here from \(\sigma_d(x) = 1 / (1 + \exp(-\eta x))\).

Derivatives with respect to each parameter of \(\beta\) are:

\[
\frac{\partial Q(\beta, \beta')}{\partial \mu_{kd}} = \sum_n r_{nk} \left( \frac{x_{nd} - \mu_{kd}}{\sigma_{kd}^2} - \frac{\mathcal{N}(\alpha_{kd}^{(1)}; \mu_{kd}, \sigma_{kd}^2) - \mathcal{N}(\alpha_{kd}^{(2)}; \mu_{kd}, \sigma_{kd}^2)}{\mathcal{N}(\alpha_{kd}^{(1)}; \mu_{kd}, \sigma_{kd}^2) - \mathcal{N}(\alpha_{kd}^{(2)}; \mu_{kd}, \sigma_{kd}^2)} \right),
\]

\[
\frac{\partial Q(\beta, \beta')}{\partial \sigma_{kd}} = \sum_n r_{nk} \left( \frac{(x_{nd} - \mu_{kd})^2}{\sigma_{kd}^2} - \frac{1}{\sigma_{kd}^2} \right) - \frac{\mathcal{N}(\alpha_{kd}^{(1)}; \mu_{kd}, \sigma_{kd}^2) - \mathcal{N}(\alpha_{kd}^{(2)}; \mu_{kd}, \sigma_{kd}^2)}{\mathcal{N}(\alpha_{kd}^{(1)}; \mu_{kd}, \sigma_{kd}^2) - \mathcal{N}(\alpha_{kd}^{(2)}; \mu_{kd}, \sigma_{kd}^2)} \right),
\]

where \(d\) is a feature number, \(n\) is a data-instance number, \(k\) is a component number of STruGMA and \(F\) is the cumulative distribution function of the univariate normal distribution.

B BREAKING THE OVERLAPPING

This section provides details for the heuristic we used to break the overlapping.

Considering two hyper-rectangles \(i\) and \(j\), the non-overlapping constraint is

\[
\max_d \left( \frac{1}{2} \left( \alpha_{id}^{(1)} + \alpha_{id}^{(2)} \right) - \frac{1}{2} \left( \alpha_{jd}^{(1)} + \alpha_{jd}^{(2)} \right) \right) - \frac{1}{2} \left( \alpha_{id}^{(2)} - \alpha_{id}^{(1)} \right) - \frac{1}{2} \left( \alpha_{jd}^{(2)} - \alpha_{jd}^{(1)} \right) \geq 0.
\]

where \(d\) is a feature number, \(i\) and \(j\) are data-instance numbers and \(k\) is a component number of STruGMA.

By looking at the form of the constraint, it can be seen that breaking overlapping can be done on only one particular dimension \(d\). In cases where the gradient-based updates of parameters violate the constraint in Eq. (1), our heuristic considers 4 adaptations:

\[
\begin{align*}
(i) \quad & \alpha_{id}^{(2)} = \alpha_{id}^{(1)} & \text{if } \alpha_{id}^{(1)} > \alpha_{id}^{(1)} \\
(ii) \quad & \alpha_{id}^{(2)} = \alpha_{id}^{(1)} & \text{otherwise}
\end{align*}
\]

and

\[
\begin{align*}
(iii) \quad & \alpha_{id}^{(1)} = \alpha_{id}^{(2)} & \text{if } \alpha_{id}^{(2)} > \alpha_{id}^{(2)} \\
(iv) \quad & \alpha_{id}^{(1)} = \alpha_{id}^{(2)} & \text{otherwise}
\end{align*}
\]

Note that (ii) and (iv) are simply alternatives of (i) and (iii) respectively when permuting \(i\) and \(j\). Furthermore, by applying any of these adaptations, it can be checked that the constraints \(\alpha_{i}^{(2)} > \alpha_{i}^{(1)}\) and \(\alpha_{j}^{(2)} > \alpha_{j}^{(1)}\) are satisfied.
Table 1: Details of neural network architectures.

<table>
<thead>
<tr>
<th>Network</th>
<th>Architecture</th>
<th>Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network1</td>
<td>Dense layer - 128, ELU</td>
<td>Marketing, Credit, Pima, Waveform, Wine</td>
</tr>
<tr>
<td></td>
<td>Dense layer - 128, ELU</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dense layer - C, Softmax</td>
<td></td>
</tr>
<tr>
<td>Network2</td>
<td>Dense layer - 128, ELU</td>
<td>Ionosphere</td>
</tr>
<tr>
<td></td>
<td>Dense layer - 128, ELU</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dropout - 0.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dense layer - 256, ELU</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dense layer - 256, ELU</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dropout - 0.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dense layer - C, Softmax</td>
<td></td>
</tr>
<tr>
<td>Network3</td>
<td>Dense layer - 10, ELU</td>
<td>Magic gamma</td>
</tr>
<tr>
<td></td>
<td>Dropout - 0.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dense layer - C, Softmax</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1 illustrates adaptations along a particular dimension $d$. As we have 4 adaptations per dimension, therefore, there are $4D$ choices, where $D$ is the size of the input space. The best choice is taken as the choice that maximises the expected log-likelihood score. Another interpretation is that it is the choice that minimises the loss in coverage of data-instances by STruGMA.

In summary, with this heuristic, breaking overlapping has the algorithmic complexity $\mathcal{O}(K^2 \times D)$, where $K$ is the number of hyper-rectangles and $D$ is the number of features.

![Figure 1: Illustration of adaptations along a particular dimension $d$. (i) and (iii) correspond to adaptations described in Eq. 2 and Eq. 3.](image)

C DETAILS OF NEURAL NETWORK ARCHITECTURES

The three architectures in the table were used in the experiments. Neural networks with different architectures that gave best results were chosen for each dataset as the black-box.