Addressing class imbalance though multi-objective evolution of neural networks

Dr. Alex Shenfield
Sheffield Hallam University
a.shenfield@shu.ac.uk
Contents

• Problem definition
• Issues ...
• The CMA-PAES-HAGA algorithm
• Experimental design
• Results
• Conclusions
Fetal CTG monitoring

- Complications during childbirth are common and accounted for 256 perinatal deaths in the UK in 2014
- Visual analysis of fetal cardiotocograms is subjective and complex
Solutions

- Clinical judgements
- Automated feature extraction systems
  - System 8000
  - SisPorto 2.0
- Machine learning techniques for clinical decision support
The dataset

- 2126 samples
- 21 attributes
- 3 classes

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB</td>
<td>FHR baseline (beats per minute)</td>
</tr>
<tr>
<td>AC</td>
<td>Number of accelerations per second</td>
</tr>
<tr>
<td>FM</td>
<td>Number of fetal movements per second</td>
</tr>
<tr>
<td>UC</td>
<td>Number of uterine contractions per second</td>
</tr>
<tr>
<td>DL</td>
<td>Number of light decelerations per second</td>
</tr>
<tr>
<td>DS</td>
<td>Number of severe decelerations per second</td>
</tr>
<tr>
<td>DP</td>
<td>Number of prolonged decelerations per second</td>
</tr>
<tr>
<td>ASTV</td>
<td>Percentage of time with abnormal short-term variability</td>
</tr>
<tr>
<td>MSTV</td>
<td>Mean value of short-term variability</td>
</tr>
<tr>
<td>ALTV</td>
<td>Percentage of time with abnormal long-term variability</td>
</tr>
<tr>
<td>MLTV</td>
<td>Mean value of long-term variability</td>
</tr>
<tr>
<td>Width</td>
<td>Width of FHR histogram</td>
</tr>
<tr>
<td>Min</td>
<td>Minimum of FHR histogram</td>
</tr>
<tr>
<td>Max</td>
<td>Maximum of FHR histogram</td>
</tr>
<tr>
<td>Nmax</td>
<td>Number of histogram peaks</td>
</tr>
<tr>
<td>Nzeros</td>
<td>Number of histogram zeros</td>
</tr>
<tr>
<td>Mode</td>
<td>Histogram mode</td>
</tr>
<tr>
<td>Mean</td>
<td>Histogram mean</td>
</tr>
<tr>
<td>Median</td>
<td>Histogram median</td>
</tr>
<tr>
<td>Variance</td>
<td>Histogram variance</td>
</tr>
<tr>
<td>Tendency</td>
<td>Histogram tendency</td>
</tr>
<tr>
<td>NSP</td>
<td>Fetal state class (code (N = normal; S = suspect; P = pathological))</td>
</tr>
</tbody>
</table>
Issues ...

• Two connected issues:
  1) Class imbalance in the dataset
  2) Optimal classifier design
Class imbalance

- Datasets are said to be “balanced” if they have a roughly even distribution of samples from each class
- However, this is often the exception!
  - Disease screening
  - Fraud detection
  - Network intrusion detection
Class imbalance

• Datasets are said to be “balanced” if they have a roughly even distribution of samples from each class

• However, this is often the exception!
  – Disease screening
  – Fraud detection
  – Network intrusion detection

In all these cases, we are mostly interested in the minority class(es)!!
Class imbalance
Class imbalance

• Algorithmic modifications
  – Aim to bias the classifier towards learning the minority class

• Data sampling and augmentation strategies
  – Aim to rebalance the class distributions by resampling the dataset

• Cost sensitive learning approaches
  – Uses *a priori* knowledge of misclassification costs to influence classification
# Class Imbalance

## Confusion Matrix Without SMOTE

<table>
<thead>
<tr>
<th>Target Class</th>
<th>Output Class</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1586</td>
<td>70</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>60</td>
<td>213</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>9</td>
<td>12</td>
<td>135</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target Class</th>
<th>Output Class</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>95.8%</td>
<td>72.2%</td>
<td>76.7%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>74.6%</td>
<td>3.3%</td>
<td>0.5%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4.2%</td>
<td>27.8%</td>
<td>23.3%</td>
</tr>
</tbody>
</table>

## Confusion Matrix Using SMOTE

<table>
<thead>
<tr>
<th>Target Class</th>
<th>Output Class</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1507</td>
<td>49</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>117</td>
<td>226</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>31</td>
<td>20</td>
<td>151</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target Class</th>
<th>Output Class</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>91.1%</td>
<td>76.6%</td>
<td>85.8%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>8.9%</td>
<td>23.4%</td>
<td>14.2%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>23.4%</td>
<td>23.4%</td>
<td>14.2%</td>
</tr>
</tbody>
</table>
 Classifier design
Classifier design

• Weights & biases → backpropagation?
• Choosing topologies??
• Evolutionary methods:
  – Parametric
  – Structural
  – NEAT: Neuro-evolution of Augmented Topologies (Stanley and Miikkulainen, 2002)
CMA-PAES-HAGA

- Basically …

1) Initialise population and evaluate
2) Do variation based on covariance matrix adaptation
3) Evaluate new candidate solutions
4) Do selection for survival based around hypervolume sorted adaptive grid archiving
5) Update parameters
6) While not finished – go to step 2

Algorithm 1 CMA-PAES-HAGA execution cycle

1: // initialise the generation counter and extreme values
2: // vector
3: \( g \leftarrow 0 \)
4: \( E \leftarrow \{e_1 = 0, e_2 = 0, \ldots, e_M = 0\} \)
5: // initialise parent population, where \( V \) contains the
6: // solutions in the search space and \( X \) contains the
7: // vectors of objective values
8: initialise parent population \( X, V \)
9: while termination criteria not met do
10:   for \( n = 1, \ldots, \lambda \) do
11:     // variation of solutions
12:     \( V_{n+1} \leftarrow V_n \)
13:     \( V'_{n+1} \leftarrow V'_{n} + \sigma_{n} \cdot \mathcal{N}(0, C_{n}) \)
14:     // check solution is within bounds
15:     if \( v'_{i}^{(U)} \leq v_{in} \leq v'_{i}^{(L)} \) then
16:       \( v_{in} = \begin{cases} v'^{(U)}_{i} & \text{if } v'_{in} > v'^{(U)}_{i} \\ v'^{(L)}_{i} & \text{otherwise} \end{cases} \)
17:     end if
18:   end for
19:   // evaluate solution and update extreme values
20:   \( X'_{n} \leftarrow f(V'_{n}) \)
21:   \( X^{*} = X \cup X' \)
22: for \( m = 1, \ldots, M \) do
23:   \( e_{m} = \begin{cases} x^{*}_{mn} & \text{if } x^{*}_{mn} > e_{m} \\ e_{m} & \text{otherwise} \end{cases} \)
24: end for
25: // selection routine as in [41]
26: \( X, V \leftarrow \text{HypervolumeSortedAGA}(X^{*}, E) \)
27: // variation routine as in [27]
28: CMAPParameterUpdate()
29: \( g \leftarrow g + 1 \)
30: end while
Experimental design

- Encoding the problem

<table>
<thead>
<tr>
<th>ANN feature</th>
<th>HL1</th>
<th>HL2</th>
<th>$W_{IL}$</th>
<th>$W_{HL1}$</th>
<th>$W_{HL2}$</th>
<th>$B_{HL1}$</th>
<th>$B_{HL2}$</th>
<th>$B_{OL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of variables</td>
<td>1</td>
<td>1</td>
<td>288</td>
<td>576</td>
<td>72</td>
<td>24</td>
<td>24</td>
<td>3</td>
</tr>
</tbody>
</table>

Hidden layer sizes

Input layer weights

Hidden layer weights

Hidden layer biases

Output layer biases

989 parameters!
Experimental design

- Performance objectives

<table>
<thead>
<tr>
<th>Objective</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective 1</td>
<td>Classification accuracy for normal fetal state</td>
</tr>
<tr>
<td>Objective 2</td>
<td>Classification accuracy for suspect fetal state</td>
</tr>
<tr>
<td>Objective 3</td>
<td>Classification accuracy for pathological fetal state</td>
</tr>
<tr>
<td>Objective 4</td>
<td>Overall classification accuracy</td>
</tr>
</tbody>
</table>
Results

- Standard feed-forward multi-layer perceptron with "best-guess" parameter settings
Results

- Support vector machine with RBF kernel
Results

- Random Forest classifier

Confusion matrix of a random forests classifier
## Results

**Performance**

<table>
<thead>
<tr>
<th></th>
<th>Feed-forward multilayer perceptron</th>
<th>Support vector machine</th>
<th>Random Forests classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>90.4%</td>
<td>91.5%</td>
<td>94.4%</td>
</tr>
<tr>
<td>M score</td>
<td>0.894</td>
<td>0.875</td>
<td>0.901</td>
</tr>
</tbody>
</table>
Results

- Parallel coordinate plot of the 50 candidate solutions produced by the CMA-PAES-HAGA algorithm
Results

- Optimised ANN with the best “M-score” (0.911)
Conclusions

- Using a multi-objective optimisation approach, class imbalance can be addressed *without* *a priori* integration of existing domain knowledge.

- Providing a diverse population of candidate solutions allows a domain expert to specify* misclassification costs either *a posteriori* or progressively.

* this can be done qualitatively rather than quantitatively if desired (by specifying the relationship between misclassification of objectives rather than an exact cost)
Conclusions

• Using evolutionary algorithms for classifier design can provide decision support for medical diagnosis
Further work

- Rigorous comparison to other methods of handling class imbalance
  - SMOTE vs ADASYN vs MDO based oversampling
- Other medical applications
- Adapting these techniques for optimising the parameters of other classifiers
- Engineering applications
Thank you and any questions?