Online, GA based Mixture of Experts: a Probabilistic Model of UCS

Narayanan U. Edakunni  Gavin Brown  Tim Kovacs

GeCCO 2011
Outline

1. Introduction
2. UCS
3. Probabilistic model for UCS
   - MoE-batch
   - MoE-online
   - MoE-GA
4. Evaluation
5. Conclusion
Outline

1 Introduction
2 UCS
3 Probabilistic model for UCS
   - MoE-batch
   - MoE-online
   - MoE-GA
4 Evaluation
5 Conclusion
Motivation

- Learning Classifier Systems (LCSs) evolve condition/action rules.
- LCSs are largely heuristic and most studies of their properties are empirical.
- Some works model part of an existing LCS mathematically and study properties of the model.
- Very little work *designs* mechanisms based on a formal model
  - This approach gives confidence that we have designed the system the right way
  - It also gives insights into how the design works
Earlier work

- In earlier work we designed an entire LCS (apart from the GA) by building a probabilistic model of UCS
  - UCS is a popular supervised LCS
  - We built a Mixture of Experts (MoE) model which approximated UCS
  - MoE is a principled framework for probabilistic learning
  - It is well-understood how to train a MoE

- Our MoE had good accuracy, but unlike UCS it:
  - did not learn online
  - used a population containing all possible rules
    - this means it cannot be used on most problems (runs too slowly or not enough memory to run at all)
Here we extend our MoE to use online updates and sample the rule space with a GA.

We call it MoE-GA. In experiments, MoE-GA:

- has population size and final accuracy comparable to UCS
- needs more training cycles than UCS
- has more realistic confidence than UCS
  - So it should handle noisy data better
Outline

1. Introduction
2. UCS
3. Probabilistic model for UCS
   - MoE-batch
   - MoE-online
   - MoE-GA
4. Evaluation
5. Conclusion
sUpervised Classifier System (UCS) [Bernadó-Mansilla and Garrell-Guiu, 2003]

- UCS has a population of classifiers (rules).
- Each rule has a condition, action(class) pair.
- Each rule has a fitness.
- For a given input, the output class is decided by a weighted vote of all the matching rules.

Training
- The parameters of rules are determined by online updates.
- The population of rules is obtained by sampling the rule space with a GA.

We derive MoE-GA: the probabilistic version of UCS which preserves these characteristics.
Outline

1. Introduction
2. UCS
3. Probabilistic model for UCS
   - MoE-batch
   - MoE-online
   - MoE-GA
4. Evaluation
5. Conclusion
The basic probabilistic model behind UCS is a Mixture of Experts (MoE) [Edakunni et al., 2009].

We derive the probabilistic model in stages starting from the simplest model.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Ensemble</th>
<th>Online learning</th>
<th>Sparse popln.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MoE-batch</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>MoE-online</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>MoE-GA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
2. UCS
3. Probabilistic model for UCS
   - MoE-batch
   - MoE-online
   - MoE-GA
4. Evaluation
5. Conclusion
MoE-batch

We represent the probability of the data as,

\[ P(c|x) = \sum_{j=1}^{M} P(c|z = j)P(x|z = j)P(z = j) \]

where,

- \( c \) - observed class label.
- \( x \) - observed input.
- \( z \) - hidden multinomial variable (stands for the rule in the popln.)
- \( P(c|x) \) - probability of the observed data under the given model.
- \( P(c|z = j) \) - probability of the class label for a rule \( j \).
- \( P(x|z = j) \) - probability of the input for a rule \( j \).
- \( P(z = j) \) - probability of the rule in a population.
Model

\[ P(c|x) = \sum_{j=1}^{M} P(c|z=j)P(x|z=j)P(z=j) \]

Parameters

- \( P(c|z=j) \) - probability of the class label for a rule \( j \).
- \( P(x|z=j) \) - probability of the input for a rule \( j \).
- \( P(z=j) \) - probability of the rule in a population.

We fit the model to observed data by maximizing the probability of the data with respect to the parameters.
MoE-batch

Training

- We train the mixture of experts model by an iterative procedure called Expectation Maximization (EM) [Dempster et al., 1977].
- The EM algorithm monotonically increases the probability of data (likelihood).
MoE-batch

Training

- We train the mixture of experts model by an iterative procedure called Expectation Maximization (EM) [Dempster et al., 1977].
- The EM algorithm monotonically increases the probability of data (likelihood).
- We start with the entire set of rules and as EM progresses the probability of some of the rules go to zero.
- We can prune rules with zero probability to obtain a sparse population.
MoE-batch vs UCS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Ensemble</th>
<th>Online learning</th>
<th>Sparse popln.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MoE-batch</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

- The MoE algorithm has parameters similar to UCS [Edakunni et al., 2009].

<table>
<thead>
<tr>
<th>MOE</th>
<th>UCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class probability $P(c</td>
<td>z)$</td>
</tr>
<tr>
<td>Match probability $P(x</td>
<td>z)$</td>
</tr>
<tr>
<td>Rule probability $P(z)$</td>
<td>Inclusion function</td>
</tr>
</tbody>
</table>

- The MoE-batch algorithm cannot handle online training data.
Outline

1. Introduction
2. UCS
3. Probabilistic model for UCS
   - MoE-batch
   - MoE-online
   - MoE-GA
4. Evaluation
5. Conclusion
MoE-online

- We now modify the batch updates of the parameters to online updates.

- The estimate for a parameter $\theta$ typically takes the form of an average of a function of the data like,

$$
\theta = \frac{\sum_{i=1}^{N} f(c_i, x_i)}{N}
$$

- This form of update can be converted to an online update by using the moving average formulation,

$$
\theta_t = \theta_{t-1} + \frac{1}{t} (f(c_t, x_t) - \theta_{t-1})
$$
MoE-online vs UCS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Ensemble</th>
<th>Online learning</th>
<th>Sparse popln.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MoE-online</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>

- The MoE algorithm stores all the rules in its population and makes it sparse as the training proceeds.
- This strategy is infeasible since the initial population size increases exponentially with the length of the input bit string.
- We need to build the population *constructively* starting from an empty population.
Outline

1 Introduction

2 UCS

3 Probabilistic model for UCS
   - MoE-batch
   - MoE-online
   - MoE-GA

4 Evaluation

5 Conclusion
MoE-GA

We can constructively build the population of classifiers by sampling individuals from the rule space.

The probability of rules $P(z)$ are converted to numerosities:

$$P(z = j) = \frac{n_j}{\sum_{j=1}^{M} n_j}$$

where,
- $n_j$ is the numerosity of rule $j$.
- $M$ is the size of the population.

The numerosities are manipulated by a GA algorithm similar to UCS.

The fitness required by the GA is computed by the online EM updates.

The GA based sampling is interleaved with the online EM updates of the parameters.
MoE-GA vs UCS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Ensemble</th>
<th>Online learning</th>
<th>Sparse popln.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MoE-GA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- We see that MoE-GA has all the features of UCS.
- MoE-GA optimizes the likelihood instead of classification accuracy.
  - We expect MoE-GA to produce classifiers with the same accuracy as UCS.
  - We expect MoE-GA to produce classifiers which predict classes with the correct levels of confidence.
  - Hence, MoE-GA should be able to handle noisy datasets better than UCS.
- We now verify our claims through empirical evaluations.
Evaluation metrics

We use the following metrics to compare the performance of the MoE algorithms against UCS.

- Mean classification error on the test dataset.
- Population size.
- Confidence: measured by the log of the probability of the test dataset as predicted by the model.
  - Even when 2 systems have the same accuracy, a more realistic confidence is indirect evidence that one system will have better accuracy on some difficult, noisy problems.
Comparison on XOR-6 dataset

Classification error

Population size

Test error

Size of population

# training epochs

UCS

UCS
Comparison on XOR-6 dataset

Classification error

Population size
Comparison on XOR-6 dataset

Classification error

Population size
Comparison on XOR-6 dataset

Classification error

Population size
The confidence (y-axis) for varying levels of noise (x-axis) on the *mux3* dataset.
Effect of noise on confidence

- The confidence (y-axis) for varying levels of noise (x-axis) on the *mux3* dataset.
Effect of noise on confidence

- The confidence (y-axis) for varying levels of noise (x-axis) on the \textit{mux3} dataset.
The confidence (y-axis) for varying levels of noise (x-axis) on the *mux3* dataset.
Comparison on UCI data

**Classification error**

![Classification error chart]

**Population size**

![Population size chart]

**Likelihood**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MoE-GA</th>
<th>UCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>car</td>
<td>-0.41(0.07)</td>
<td>-7.08(3.91)</td>
</tr>
<tr>
<td>mushroom</td>
<td>-0.85(0.02)</td>
<td>-1.08(0.66)</td>
</tr>
<tr>
<td>kr-vs-kp</td>
<td>-0.94(0.02)</td>
<td>-27.77(5.43)</td>
</tr>
</tbody>
</table>
In experiments, MoE-GA:

- has population size and final accuracy comparable to UCS
- needs more training cycles than UCS
- has more realistic confidence in its predictions than UCS
  - So it should have better accuracy on some difficult, noisy datasets
  - However, our limited experiments have not shown that
Outline

1. Introduction
2. UCS
3. Probabilistic model for UCS
   - MoE-batch
   - MoE-online
   - MoE-GA
4. Evaluation
5. Conclusion
We derived a probabilistic framework of UCS in three stages.

- MoE-batch (previous work): offline and maximal population
- MoE-online converted to online learning
- MoE-GA added a GA to search the space of rules

The final stage is an efficient online learning algorithm that combines the robustness of a probabilistic formulation with the space efficiency (small populations) of UCS.
Future work

- Optimise the learning curve of MoE-GA
- Further empirical comparisons of MoE-GA and UCS
  - Is MoE-GA really better on difficult, noisy data?
- The generic formulation of MoE opens a number of avenues for extension e.g. regression, density estimation, dimensionality reduction and so on.
References

