Algebraic evaluation of independent classifiers on unlabeled data

The core mathematical theorem of our submission is explained here.

The Polynomial System - Algebra of Error for Independent Binary Classifiers

The following set of quartic polynomials define algebraic system that must be solved to estimate the accuracy of three classifiers as well as the true prevalence of the labels. There are $8 = 2^3$ polynomials, one for each of the possible voting pattern of three classifiers doing binary label classification. They define a polynomial ideal that we call an “evaluation ideal.”

```
In[∘]:= evaluationIdeal = MakeIndependentVotingIdeal[{1, 2, 3}]
```

Out[∘]=

$$\begin{aligned}\{&P_\alpha P_{1,\alpha} P_{2,\alpha} P_{3,\alpha} + (1 - P_\alpha)(1 - P_{1,\beta})(1 - P_{2,\beta})(1 - P_{3,\beta}) - f_{\alpha,\alpha,\alpha},\\
&P_\alpha P_{1,\alpha} P_{2,\alpha}(1 - P_{3,\alpha}) + (1 - P_\alpha)(1 - P_{1,\beta})(1 - P_{2,\beta}) P_{3,\beta} - f_{\alpha,\alpha,\beta},\\
&P_\alpha P_{1,\alpha}(1 - P_{2,\alpha}) P_{3,\alpha} + (1 - P_\alpha)(1 - P_{1,\beta}) P_{2,\beta}(1 - P_{3,\beta}) - f_{\alpha,\beta,\alpha},\\
&P_\alpha P_{1,\alpha}(1 - P_{2,\alpha})(1 - P_{3,\alpha}) + (1 - P_\alpha)(1 - P_{1,\beta}) P_{2,\beta} P_{3,\beta} - f_{\alpha,\beta,\beta},\\
&P_\alpha(1 - P_{1,\alpha}) P_{2,\alpha} P_{3,\alpha} + (1 - P_\alpha) P_{1,\beta}(1 - P_{2,\beta})(1 - P_{3,\beta}) - f_{\beta,\alpha,\alpha},\\
&P_\alpha(1 - P_{1,\alpha}) P_{2,\alpha}(1 - P_{3,\alpha}) + (1 - P_\alpha) P_{1,\beta}(1 - P_{2,\beta}) P_{3,\beta} - f_{\beta,\alpha,\beta},\\
&P_\alpha(1 - P_{1,\alpha})(1 - P_{2,\alpha}) P_{3,\alpha} + (1 - P_\alpha) P_{1,\beta} P_{2,\beta}(1 - P_{3,\beta}) - f_{\beta,\beta,\alpha},\\
&P_\alpha(1 - P_{1,\alpha})(1 - P_{2,\alpha})(1 - P_{3,\alpha}) + (1 - P_\alpha) P_{1,\beta} P_{2,\beta} P_{3,\beta} - f_{\beta,\beta,\beta}\}\end{aligned}$$

Evaluation on unlabeled data

In classification problems, the ground truth is the correct label for each item in a dataset. The statistics of ground truth solved by the above polynomial system are:
- the prevalence of the labels.
- the accuracies of the classifiers for each label.

Note that these are sample statistics. The method is not trying to infer anything about the process that created the data over which the binary classifiers have been run. There is some data, you ran binary classifiers over it and now you want to ask statistical questions about their decisions over that data. It would be trivial to answer them IF you had the true labels for each of the data points in your dataset. But you don’t have the ground truth so you need techniques like the one explained here.

Creating error independent binary classifiers

The purpose of this notebook is to demonstrate the correctness of exact algebraic evaluation in the somewhat unrealistic case of error independent binary classifiers. But although unrealistic, this exact solution is important theoretically. For one, exact solutions of anything are rare. A class of problems that frequently admit exact solution are those that assume some sort of independence. We are doing the same here. This unrealistic solution will turn out to be very useful in practice.

The following commands create the ground truth for this problem - a list of labels for the true label of each data point in the dataset.

In[ ]:=
```
desiredALabelPrevalence = 23 / 100
datasetSize = 10 000
labelGroundTruth =
  Table[If[RandomReal[] < desiredALabelPrevalence, α, β], {datasetSize}];
```

Out[ ]=

$$\frac{23}{100}$$

Out[ ]=

```
10 000
```

Let’s check that the ground truth makes sense

In[ ]:=
```
RandomSample[labelGroundTruth, 10]
```

Out[ ]=
```
{β, β, β, β, β, β, β, β, β, β}
```

In[ ]:=
```
labelGroundTruth // Tally
```

Out[ ]=
```
{{α, 2253}, {β, 7747}}
```

Now we construct the synthetic data for the classifier label decisions. Again, we are using synthetic data to create a simple example that focuses on the math of the algorithm. You can substitute numbers from real independent classifiers if you have them.

1. The first thing we need to specify is the accuracy of each of the classifiers. This accuracy is one of the statistics we seek to estimate with the polynomial system that is constructed from the classifier decisions alone. I’ll arbitrarily pick them uniformly from the same range. I need two accuracies, one for each label since we are doing binary classification.

In[ ]:=
```
classifierAccuracies = Table[
  {(* The α accuracy *) α → RandomReal[{0.7, 0.9}],
   (* The β accuracy *) β → RandomReal[{0.7, 0.9}]} // Association, {3}]
```

Out[ ]=
```
{<|α → 0.797697, β → 0.819408|>,
 <|α → 0.738878, β → 0.888438|>, <|α → 0.717426, β → 0.757177|>}
```

2. Now we need a function that takes a classifier's accuracies, a list true labels, and produces a sample of its classification decisions.
- OtherLabel is a convenience function to create incorrect classification decisions.
- SyntheticClassification is the actual function that produces a sample of what a classifier with the specified accuracies would produce on a dataset.

In[ ]:=
```
OtherLabel[α] = β
OtherLabel[β] = α
```

Out[ ]=
β

Out[ ]=
α

In[ ]:=
```
SyntheticClassification[classifierAccuracies_Association, dataset_List] :=
 dataset //
  Map[
    If[
      (* Throw a die, if below accuracy,
      pick the true label, if not, pick the other label *)
      RandomReal[] < classifierAccuracies[#], #, OtherLabel[#]] &,
    #] &
```

3. The classifier decisions are next. Three lists, one for each classifier, as long as the dataset.

In[ ]:=
```
classifierDecisions =
  Map[SyntheticClassification[#, labelGroundTruth] &, classifierAccuracies];
```

Let's make sure things are okay

In[ ]:=
```
classifierDecisions〚1〛 // {RandomSample[#, 10], # // Length} &
classifierDecisions〚2〛 // {RandomSample[#, 10], # // Length} &
classifierDecisions〚3〛 // {RandomSample[#, 10], # // Length} &
```

Out[ ]=
{{β, β, β, β, β, α, α, α, β, α}, 10 000}

Out[ ]=
{{α, β, α, α, β, β, α, α, β, β}, 10 000}

Out[ ]=
{{β, α, β, β, α, β, α, α, β, β}, 10 000}

Let's also make sure that we are getting noisy classification of the true labels

In[ ]:=
```
Map[Tally, classifierDecisions]
```

Out[ ]=
{{{α, 3186}, {β, 6814}}, {{α, 2509}, {β, 7491}}, {{α, 3437}, {β, 6563}}}

4. Now we can construct the left side input of the ground truth polynomials for independent binary classifiers - the frequency of the voting patters

In[ ]:=
```
votingPatternCounts = classifierDecisions // Transpose // Tally
```
Out[ ]=
```
{{{α, α, α}, 982}, {{β, β, β}, 4355}, {{α, α, β}, 474}, {{β, β, α}, 1406},
 {{α, β, β}, 1107}, {{β, α, β}, 627}, {{β, α, α}, 426}, {{α, β, α}, 623}}
```

Let’s sort them and pretty print them to see the pattern better

In[ ]:=
```
votingPatternCounts // Sort // Column
```
Out[ ]=
```
{{α, α, α}, 982}
{{α, α, β}, 474}
{{α, β, α}, 623}
{{α, β, β}, 1107}
{{β, α, α}, 426}
{{β, α, β}, 627}
{{β, β, α}, 1406}
{{β, β, β}, 4355}
```

Everything looks okay with our synthetic data. Note that votingPatternCounts no longer contains any information about the true label for any of the data points in labelGroundTruth. Hence, any algorithm that only uses votingPatternCounts - the observed frequency of classifier voting patterns, would be carrying out ground truth inference. The goal of the next section is to estimate classifierAccuracies and the prevalence of the labels using only votingPatternCounts.

Exact Solution for Three Independent Binary Classifiers

Let’s assemble all the pieces needed to carry out evaluation. We first take a look at the variables in the evaluation ideal and make sure we have the correct notation.

In[ ]:=
```
Variables /@ evaluationIdeal // Sort /@ # & // Column[#, Dividers → All] &
```
Out[ ]=

$\{P_{\alpha}, P_{1,\alpha}, P_{1,\beta}, P_{2,\alpha}, P_{2,\beta}, P_{3,\alpha}, P_{3,\beta}, f_{\alpha,\alpha,\alpha}\}$
$\{P_{\alpha}, P_{1,\alpha}, P_{1,\beta}, P_{2,\alpha}, P_{2,\beta}, P_{3,\alpha}, P_{3,\beta}, f_{\alpha,\alpha,\beta}\}$
$\{P_{\alpha}, P_{1,\alpha}, P_{1,\beta}, P_{2,\alpha}, P_{2,\beta}, P_{3,\alpha}, P_{3,\beta}, f_{\alpha,\beta,\alpha}\}$
$\{P_{\alpha}, P_{1,\alpha}, P_{1,\beta}, P_{2,\alpha}, P_{2,\beta}, P_{3,\alpha}, P_{3,\beta}, f_{\alpha,\beta,\beta}\}$
$\{P_{\alpha}, P_{1,\alpha}, P_{1,\beta}, P_{2,\alpha}, P_{2,\beta}, P_{3,\alpha}, P_{3,\beta}, f_{\beta,\alpha,\alpha}\}$
$\{P_{\alpha}, P_{1,\alpha}, P_{1,\beta}, P_{2,\alpha}, P_{2,\beta}, P_{3,\alpha}, P_{3,\beta}, f_{\beta,\alpha,\beta}\}$
$\{P_{\alpha}, P_{1,\alpha}, P_{1,\beta}, P_{2,\alpha}, P_{2,\beta}, P_{3,\alpha}, P_{3,\beta}, f_{\beta,\beta,\alpha}\}$
$\{P_{\alpha}, P_{1,\alpha}, P_{1,\beta}, P_{2,\alpha}, P_{2,\beta}, P_{3,\alpha}, P_{3,\beta}, f_{\beta,\beta,\beta}\}$

Note the extreme symmetry of these evaluation polynomials. All the members of the generating ideal shown here have exactly the same unknown sample statistics variables.

Preparing the observed voting frequencies rules

Recall that we have the voting pattern counts already

In[ ]:=
```
votingPatternCounts // Sort // Column
```

Out[ ]=
```
{{α, α, α}, 982}
{{α, α, β}, 474}
{{α, β, α}, 623}
{{α, β, β}, 1107}
{{β, α, α}, 426}
{{β, α, β}, 627}
{{β, β, α}, 1406}
{{β, β, β}, 4355}
```

Let’s turn them into rules so we can substitute their value into the ground truth polynomials. We need to normalize them so we need the total count first. We know it is 10K, but let’s check it by computation so we can verify everything is okay

In[ ]:=
```
totalVotes = votingPatternCounts // Last /@ # & // Total
totalVotes == datasetSize
```

Out[ ]=
```
10 000
```

Out[ ]=
```
True
```

In[ ]:=
```
frequencyRules = votingPatternCounts //
  Map[(Subscript[f, Sequence @@ #[[1]]] → #[[2]] / datasetSize) &, #] &
```

Out[ ]=

$$\left\{f_{\alpha,\alpha,\alpha} \to \frac{491}{5000},\ f_{\beta,\beta,\beta} \to \frac{871}{2000},\ f_{\alpha,\alpha,\beta} \to \frac{237}{5000},\ f_{\beta,\beta,\alpha} \to \frac{703}{5000},\ f_{\alpha,\beta,\beta} \to \frac{1107}{10\,000},\ f_{\beta,\alpha,\beta} \to \frac{627}{10\,000},\ f_{\beta,\alpha,\alpha} \to \frac{213}{5000},\ f_{\alpha,\beta,\alpha} \to \frac{623}{10\,000}\right\}$$

The final polynomial set

In[∘]:=
```
finalPolynomialSet = evaluationIdeal /. frequencyRules
```

Out[∘]=

$$\left\{-\frac{491}{5000}+P_{\alpha}P_{1,\alpha}P_{2,\alpha}P_{3,\alpha}+(1-P_{\alpha})\left(1-P_{1,\beta}\right)\left(1-P_{2,\beta}\right)\left(1-P_{3,\beta}\right),\right.$$
$$-\frac{237}{5000}+P_{\alpha}P_{1,\alpha}P_{2,\alpha}\left(1-P_{3,\alpha}\right)+(1-P_{\alpha})\left(1-P_{1,\beta}\right)\left(1-P_{2,\beta}\right)P_{3,\beta},$$
$$-\frac{623}{10\,000}+P_{\alpha}P_{1,\alpha}\left(1-P_{2,\alpha}\right)P_{3,\alpha}+(1-P_{\alpha})\left(1-P_{1,\beta}\right)P_{2,\beta}\left(1-P_{3,\beta}\right),$$
$$-\frac{1107}{10\,000}+P_{\alpha}P_{1,\alpha}\left(1-P_{2,\alpha}\right)\left(1-P_{3,\alpha}\right)+(1-P_{\alpha})\left(1-P_{1,\beta}\right)P_{2,\beta}P_{3,\beta},$$
$$-\frac{213}{5000}+P_{\alpha}\left(1-P_{1,\alpha}\right)P_{2,\alpha}P_{3,\alpha}+(1-P_{\alpha})P_{1,\beta}\left(1-P_{2,\beta}\right)\left(1-P_{3,\beta}\right),$$
$$-\frac{627}{10\,000}+P_{\alpha}\left(1-P_{1,\alpha}\right)P_{2,\alpha}\left(1-P_{3,\alpha}\right)+(1-P_{\alpha})P_{1,\beta}\left(1-P_{2,\beta}\right)P_{3,\beta},$$
$$-\frac{703}{5000}+P_{\alpha}\left(1-P_{1,\alpha}\right)\left(1-P_{2,\alpha}\right)P_{3,\alpha}+(1-P_{\alpha})P_{1,\beta}P_{2,\beta}\left(1-P_{3,\beta}\right),$$
$$\left.-\frac{871}{2000}+P_{\alpha}\left(1-P_{1,\alpha}\right)\left(1-P_{2,\alpha}\right)\left(1-P_{3,\alpha}\right)+(1-P_{\alpha})P_{1,\beta}P_{2,\beta}P_{3,\beta}\right\}$$

Let’s check that only the seven unknown statistics of the ground truth remain in the final polynomial system.

In[∘]:=
```
statisticsOfGroundTruth =
  finalPolynomialSet // Variables /@ # & // Flatten // DeleteDuplicates
```

Out[∘]=

$$\{P_{\alpha},\ P_{1,\alpha},\ P_{2,\alpha},\ P_{3,\alpha},\ P_{1,\beta},\ P_{2,\beta},\ P_{3,\beta}\}$$

The “magical” solution

In[∘]:=
```
sols = Solve[Map[# == 0 &, finalPolynomialSet], statisticsOfGroundTruth]
```

Out[∘]=

$$\Big\{\Big\{P_\alpha \to \frac{26\,351\,815\,728\,362\,500 - \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{52\,703\,631\,456\,725\,000},$$
$$P_{1,\alpha} \to \frac{23\,706\,303\,422\,472\,499 + \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{49\,047\,621\,338\,677\,046},$$
$$P_{2,\alpha} \to \frac{19\,580\,987\,545\,285\,165 + \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{45\,840\,001\,121\,224\,284},$$
$$P_{3,\alpha} \to \frac{28\,365\,918\,550\,451\,091 + \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{59\,022\,948\,171\,315\,388},$$
$$P_{1,\beta} \to \frac{25\,341\,317\,916\,204\,547 + \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{49\,047\,621\,338\,677\,046},$$
$$P_{2,\beta} \to \frac{26\,259\,013\,575\,939\,119 + \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{45\,840\,001\,121\,224\,284},$$
$$P_{3,\beta} \to \frac{30\,657\,029\,620\,864\,297 + \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{59\,022\,948\,171\,315\,388}\Big\},$$
$$\Big\{P_\alpha \to \frac{26\,351\,815\,728\,362\,500 + \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{52\,703\,631\,456\,725\,000},$$
$$P_{1,\alpha} \to \frac{23\,706\,303\,422\,472\,499 - \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{49\,047\,621\,338\,677\,046},$$
$$P_{2,\alpha} \to \frac{19\,580\,987\,545\,285\,165 - \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{45\,840\,001\,121\,224\,284},$$
$$P_{3,\alpha} \to \frac{28\,365\,918\,550\,451\,091 - \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{59\,022\,948\,171\,315\,388},$$
$$P_{1,\beta} \to \frac{25\,341\,317\,916\,204\,547 - \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{49\,047\,621\,338\,677\,046},$$
$$P_{2,\beta} \to \frac{26\,259\,013\,575\,939\,119 - \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{45\,840\,001\,121\,224\,284},$$
$$P_{3,\beta} \to \frac{30\,657\,029\,620\,864\,297 - \sqrt{212\,915\,589\,402\,826\,661\,294\,025\,489\,275\,545}}{59\,022\,948\,171\,315\,388}\Big\}\Big\}$$

These are algebraic numbers since they are produced by exact algebraic polynomials. Mathematica is able to return them in this way because it uses Buchberger’s algorithm inside its Solve function. These exact numbers are hard to interpret. So lets get a floating point representation of them -

In[◦]:=
```
sols // N
```

Out[◦]=

$\{\{P_\alpha \to 0.223138,\ P_{1,\alpha} \to 0.780832,\ P_{2,\alpha} \to 0.745476,\ P_{3,\alpha} \to 0.727811,\ P_{1,\beta} \to 0.814167,\ P_{2,\beta} \to 0.891157,\ P_{3,\beta} \to 0.766628\},\ \{P_\alpha \to 0.776862,\ P_{1,\alpha} \to 0.185833,\ P_{2,\alpha} \to 0.108843,\ P_{3,\alpha} \to 0.233372,\ P_{1,\beta} \to 0.219168,\ P_{2,\beta} \to 0.254524,\ P_{3,\beta} \to 0.272189\}\}$

Let’s pretty print them and compare them with what we used to simulate them.

In[◦]:=
```
{First@sols // First /@ # & // Rest,
  First@sols // N // Last /@ # & // Rest,
  classifierAccuracies // Values /@ # & // Transpose // Flatten} //
 Grid[#, Dividers → All] &
```

Out[◦]=

$P_{1,\alpha}$	$P_{2,\alpha}$	$P_{3,\alpha}$	$P_{1,\beta}$	$P_{2,\beta}$	$P_{3,\beta}$
0.780832	0.745476	0.727811	0.814167	0.891157	0.766628
0.797697	0.738878	0.717426	0.819408	0.888438	0.757177

We’ll leave you to ponder why the result is not closer to our distribution values.