Evaluation ideal and variety for a trio of error independentbinary classifiers

Introduction

This notebook will detail the algebraic geometry computations that take us from the “evaluation ideal” created from the voting patterns of a trio of binary classifiers to the “evaluation variety”. An evaluation ideal is a set of polynomials connecting observable voting pattern frequencies by the classifiers to unknown sample statistics of the ground truth that are our evaluation goal. We want to “grade” the classifiers using only the frequencies of their voting patterns.

That “grade” exists in sample statistics space. The test has already been taken. We have the decisions of the judges. We are faced with the task of grading them now. Not in the future, not in the past. This is another example of how the task of evaluation is much simpler than that of training. We have to estimate something that already exists, if you will. And there is only one time we have to do it. Training is much harder. You must create judges that, in the future, will behave correctly. And they have to do it many times. The task of evaluation is trivial in comparison. Why have we not conquered this much simpler space of the whole enterprise of learning?

Algebraic geometry of three error independent binary classifiers

The mathematics of algebraic evaluation is algebraic geometry. Every algebraic evaluation problem can be stated as a polynomial system relating observable decision events to unknown sample statistics. Here we are going to define that polynomial system assuming that the classifiers made errors independently on the sample. This is “the spherical cow” of Evaluation Land - the simplifying assumption that allows you to proceed forward and carry out computations that give you insight into the original problem. Workers in Training Land also have a preferred spherical cow - “consider an identically, independently drawn sample”. It may take some getting used to this new cow if you are a new visitor from Training Land.

The evaluation ideal of three error independent binary classifiers

Algebraic geometry is mainly the study of the connection between sets of polynomials and geometric objects in the variable space of those polynomials. The sets of polynomials are called “polynomial ideals". A set of linear equations is also a polynomial ideal. We define the "evaluation ideal" of our evaluation to be,

In[1]:=
```
Clear[MakeIndependentVotingIdeal]
MakeIndependentVotingIdeal[{i_, j_, k_}] :=
```

$$\{P_\alpha P_{i,\alpha} P_{j,\alpha} P_{k,\alpha} + (1-P_\alpha)(1-P_{i,\beta})(1-P_{j,\beta})(1-P_{k,\beta}) - f_{\alpha,\alpha,\alpha},$$
$$P_\alpha P_{i,\alpha} P_{j,\alpha}(1-P_{k,\alpha}) + (1-P_\alpha)(1-P_{i,\beta})(1-P_{j,\beta}) P_{k,\beta} - f_{\alpha,\alpha,\beta},$$
$$P_\alpha P_{i,\alpha}(1-P_{j,\alpha}) P_{k,\alpha} + (1-P_\alpha)(1-P_{i,\beta}) P_{j,\beta}(1-P_{k,\beta}) - f_{\alpha,\beta,\alpha},$$
$$P_\alpha P_{i,\alpha}(1-P_{j,\alpha})(1-P_{k,\alpha}) + (1-P_\alpha)(1-P_{i,\beta}) P_{j,\beta} P_{k,\beta} - f_{\alpha,\beta,\beta},$$
$$P_\alpha(1-P_{i,\alpha}) P_{j,\alpha} P_{k,\alpha} + (1-P_\alpha) P_{i,\beta}(1-P_{j,\beta})(1-P_{k,\beta}) - f_{\beta,\alpha,\alpha},$$
$$P_\alpha(1-P_{i,\alpha}) P_{j,\alpha}(1-P_{k,\alpha}) + (1-P_\alpha) P_{i,\beta}(1-P_{j,\beta}) P_{k,\beta} - f_{\beta,\alpha,\beta},$$
$$P_\alpha(1-P_{i,\alpha})(1-P_{j,\alpha}) P_{k,\alpha} + (1-P_\alpha) P_{i,\beta} P_{j,\beta}(1-P_{k,\beta}) - f_{\beta,\beta,\alpha},$$
$$P_\alpha(1-P_{i,\alpha})(1-P_{j,\alpha})(1-P_{k,\alpha}) + (1-P_\alpha) P_{i,\beta} P_{j,\beta} P_{k,\beta} - f_{\beta,\beta,\beta}\}$$

One convention in algebraic geometry may bother you. Ultimately we are interested in the geometrical object these polynomials define in the finite space needed for evaluating three independent binary classifiers. We want to consider the points in sample statistics space where all these equations are zero. We are really interested in these equations,

In[3]:=
```
MakeIndependentVotingIdeal[{1, 2, 3}] // Map[(# == 0) &, #] &
```

Out[3]=

$$\{P_\alpha P_{1,\alpha} P_{2,\alpha} P_{3,\alpha} + (1-P_\alpha)(1-P_{1,\beta})(1-P_{2,\beta})(1-P_{3,\beta}) - f_{\alpha,\alpha,\alpha} == 0,$$
$$P_\alpha P_{1,\alpha} P_{2,\alpha}(1-P_{3,\alpha}) + (1-P_\alpha)(1-P_{1,\beta})(1-P_{2,\beta}) P_{3,\beta} - f_{\alpha,\alpha,\beta} == 0,$$
$$P_\alpha P_{1,\alpha}(1-P_{2,\alpha}) P_{3,\alpha} + (1-P_\alpha)(1-P_{1,\beta}) P_{2,\beta}(1-P_{3,\beta}) - f_{\alpha,\beta,\alpha} == 0,$$
$$P_\alpha P_{1,\alpha}(1-P_{2,\alpha})(1-P_{3,\alpha}) + (1-P_\alpha)(1-P_{1,\beta}) P_{2,\beta} P_{3,\beta} - f_{\alpha,\beta,\beta} == 0,$$
$$P_\alpha(1-P_{1,\alpha}) P_{2,\alpha} P_{3,\alpha} + (1-P_\alpha) P_{1,\beta}(1-P_{2,\beta})(1-P_{3,\beta}) - f_{\beta,\alpha,\alpha} == 0,$$
$$P_\alpha(1-P_{1,\alpha}) P_{2,\alpha}(1-P_{3,\alpha}) + (1-P_\alpha) P_{1,\beta}(1-P_{2,\beta}) P_{3,\beta} - f_{\beta,\alpha,\beta} == 0,$$
$$P_\alpha(1-P_{1,\alpha})(1-P_{2,\alpha}) P_{3,\alpha} + (1-P_\alpha) P_{1,\beta} P_{2,\beta}(1-P_{3,\beta}) - f_{\beta,\beta,\alpha} == 0,$$
$$P_\alpha(1-P_{1,\alpha})(1-P_{2,\alpha})(1-P_{3,\alpha}) + (1-P_\alpha) P_{1,\beta} P_{2,\beta} P_{3,\beta} - f_{\beta,\beta,\beta} == 0\}$$

But it is a pain to carry all these equal to zero notation around. So we drop it, and prefer to work with,

In[4]:=
```
MakeIndependentVotingIdeal[{1, 2, 3}]
```

Out[4]=

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

Dropping the notation has no effect. Algebraic manipulations of the above set (multiplying them together, etc.) would be equivalent to polynomials of zero for points that satisfied the input set.

One “forest for the trees” note: the above ideal is one many possible ones. Algebraic evaluation is very much like data streaming algorithms. You are creating a sketch of the decisions by an ensemble of noisy judges. In the case of binary classification considered here, that “data sketch” is the frequency of their item-by-item voting patterns. Other polynomial systems are possible even for independent binary classifiers. We could be trying to evaluate sample statistics that look at how judges evaluated two different sample items, for example.

The evaluation variety of three error independent binary classifiers

The goal of algebraic evaluation is to obtain “grades” for the noisy judges on unlabeled data. We know the types of grades we want. We want their exact grades. Not spreads of where we think their grade is - no probabilistic solutions! That is not quite what we get in algebraic evaluation. The mathematical object you get as a grade is actually a geometric object in sample statistics space. The true grade lies on that geometric object. Mathematicians call the geometrical objects defined by polynomials - varieties. You will see that for error-independent binary classifiers that geometric object is almost what we want - a collection of two points. For correlated classifiers those points “bloom out”. It is an unsolved problem in algebraic evaluation to characterize the surface for correlated classifiers - but it does exist! The evaluation ideal for any set of correlated classifiers can be written down and it defines, by construction, an evaluation variety that is guaranteed to contain the true evaluation point for the classifiers. All that is left when you build the evaluation ideal is to figure out what that surface is and whether it would be useful to your AI safety task.

The ground truth for the performance of three noisy binary classifiers

Before continuing to present the formalism of algebraic evaluation, let’s do a quick end run to our goal - the grade for noisy binary classifiers. I’ll use the UCI Adult run that can be found in the Python code - AlgebraicEvaluation.py. The input to algebraic evaluation in our current application is the frequency of voting patterns by the classifiers. Here is how it looked for a single run of three binary classifiers on the UCI Adult dataset.

```
In[5]:= singleEvaluationUCIAdult =
  <|0 → <|{0, 0, 0} → 715, {0, 0, 1} → 161, {0, 1, 0} → 2406, {0, 1, 1} → 455,
      {1, 0, 0} → 290, {1, 0, 1} → 94, {1, 1, 0} → 1335, {1, 1, 1} → 231|>,
   1 → <|{0, 0, 0} → 271, {0, 0, 1} → 469, {0, 1, 0} → 3395, {0, 1, 1} → 7517,
      {1, 0, 0} → 272, {1, 0, 1} → 399, {1, 1, 0} → 6377, {1, 1, 1} → 12 455|>|>

Out[5]= <|0 → <|{0, 0, 0} → 715, {0, 0, 1} → 161, {0, 1, 0} → 2406, {0, 1, 1} → 455,
      {1, 0, 0} → 290, {1, 0, 1} → 94, {1, 1, 0} → 1335, {1, 1, 1} → 231|>,
   1 → <|{0, 0, 0} → 271, {0, 0, 1} → 469, {0, 1, 0} → 3395, {0, 1, 1} → 7517,
      {1, 0, 0} → 272, {1, 0, 1} → 399, {1, 1, 0} → 6377, {1, 1, 1} → 12 455|>|>
```

This is not a randomly selected run of binary classifiers on the UCI Adult dataset. It is being used for various reasons. It was engineered to be as close to error independence as possible. This will be discussed more later. For now, let's verify that, in fact, the classifiers are near error independence in this sample.

In[55]:=
```
evaluationGroundTruth = GTClassifiers[singleEvaluationUCIAdult]
```

Out[55]=

$$\left\langle\left| P_{\alpha} \to \frac{5687}{36\,842}, P_{1,\alpha} \to \frac{3737}{5687}, P_{2,\alpha} \to \frac{1260}{5687}, P_{3,\alpha} \to \frac{4746}{5687}, P_{1,\beta} \to \frac{6501}{10\,385}, P_{2,\beta} \to \frac{29\,744}{31\,155}, P_{3,\beta} \to \frac{4168}{6231}, \Gamma_{1,2,\alpha} \to \frac{273\,192}{32\,341\,969}, \Gamma_{1,3,\alpha} \to \frac{13\,325}{32\,341\,969}, \Gamma_{2,3,\alpha} \to -\frac{264\,525}{32\,341\,969}, \Gamma_{1,2,\beta} \to \frac{2\,204\,576}{323\,544\,675}, \Gamma_{1,3,\beta} \to -\frac{79\,682}{12\,941\,787}, \Gamma_{2,3,\beta} \to \frac{94\,508}{38\,825\,361}, \Gamma_{1,2,3,\alpha} \to \frac{452\,568\,508}{183\,928\,777\,703}, \Gamma_{1,2,3,\beta} \to -\frac{27\,265\,589}{134\,400\,457\,995} \right|\right\rangle$$

The evaluation ground truth is what we want. It was calculated above by cheating - we have the by-label counts so we can easily compute ALL the sample statistics required to explain exactly the frequency patterns we observe when we DO NOT have the knowledge of the true labels. Note that the evaluation ground truth are integer ratios. The exact sample statistics for evaluation are a subset of the real numbers. This is crucial. This allows algebraic evaluators to get closer to the true grades. Integer ratios are also in the field of algebraic numbers. But the field of algebraic numbers is less dense than reals!

This evaluation also reminds the reader of why evaluation is easier than training. There are no unknown unknowns. Evaluation computes sample statistics. The space of sample statistics required to explain observable voting patterns is finite and complete. You may not know what all these sample statistics are, but they are all you would need to know to describe the frequency of their observed decisions.

It is hard to compare integer ratios so let's get the floating point approximation to the evaluation to confirm the claim that these classifiers are nearly error independent.

In[56]:=
```
evaluationGroundTruth // N
```

Out[56]=

$$\left\langle\left| P_{\alpha} \to 0.154362, P_{1,\alpha} \to 0.657113, P_{2,\alpha} \to 0.221558, P_{3,\alpha} \to 0.834535, P_{1,\beta} \to 0.625999, P_{2,\beta} \to 0.95471, P_{3,\beta} \to 0.668913, \Gamma_{1,2,\alpha} \to 0.00844698, \Gamma_{1,3,\alpha} \to 0.000412003, \Gamma_{2,3,\alpha} \to -0.008179, \Gamma_{1,2,\beta} \to 0.00681382, \Gamma_{1,3,\beta} \to -0.00615695, \Gamma_{2,3,\beta} \to 0.00243418, \Gamma_{1,2,3,\alpha} \to 0.00246056, \Gamma_{1,2,3,\beta} \to -0.000202868 \right|\right\rangle$$

One can see that all the pair error correlation terms ($\Gamma_{i,j,label}$) are less that 1% absolute. This is encouraging. Since these classifiers are already so near error independence on the sample, will the using an evaluation ideal that assumes they are error-independent work well enough? Let's try it by using Mathematica's built-in algebraic geometry algorithms.

Evaluation with Mathematica's Solve function

Since we are trying to simulate evaluation on unlabeled data, we need to project the by-true-label counts into the counts that are observed when we have no knowledge of the true labels. This is easy. For binary classification, the observed counts for a voting pattern is the sum of the voting pattern counts when you know the true label. For example,

In[57]:=
```
sizeOfTestSet =
 singleEvaluationUCIAdult // Values // Map[Values, #] & // Flatten // Total
fα,β,α →
 Sum[singleEvaluationUCIAdult[label][{0, 0, 0}], {label, {0, 1}}] / sizeOfTestSet
```

Out[57]=

36 842

Out[58]=

$$f_{\alpha,\beta,\alpha} \to \frac{493}{18\,421}$$

So the “data sketch” for the evaluation of these three noisy binary classifiers is given by

In[60]:=
```
evaluationDataSketch =
 Transpose@{{fα,α,α, fα,α,β, fα,β,α, fα,β,β, fβ,α,α, fβ,α,β, fβ,β,α, fβ,β,β},
    Map[(Sum[singleEvaluationUCIAdult[label][#], {label, {0, 1}}] / sizeOfTestSet) &,
     {{0, 0, 0}, {0, 0, 1}, {0, 1, 0}, {0, 1, 1},
      {1, 0, 0}, {1, 0, 1}, {1, 1, 0}, {1, 1, 1}}]} //
   Rule @@ # & /@ # &
```

Out[60]=

$$\left\{f_{\alpha,\alpha,\alpha} \to \frac{493}{18\,421},\ f_{\alpha,\alpha,\beta} \to \frac{315}{18\,421},\ f_{\alpha,\beta,\alpha} \to \frac{5801}{36\,842},\ f_{\alpha,\beta,\beta} \to \frac{3986}{18\,421},\ f_{\beta,\alpha,\alpha} \to \frac{281}{18\,421},\ f_{\beta,\alpha,\beta} \to \frac{493}{36\,842},\ f_{\beta,\beta,\alpha} \to \frac{3856}{18\,421},\ f_{\beta,\beta,\beta} \to \frac{6343}{18\,421}\right\}$$

The goal of our current evaluation is to get estimates for the following sample statistics,

In[61]:=
```
evaluationVariables = MakeIndependentVotingIdeal[{1, 2, 3}] //
      Variables /@ # & // Flatten // DeleteDuplicates // Cases[#, Except[f__]] & // Sort
```

Out[61]=

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

Mathematica uses algebraic geometry under the hood of the Solve function to give us the grades for these classifiers.

In[64]:=
```
independentModelEvaluation = Solve[
   (MakeIndependentVotingIdeal[{1, 2, 3}] // Map[(# == 0) &, #] &) /.
    evaluationDataSketch,
   evaluationVariables] // Map[Association, #] &
```

Out[64]=

$$\Big\{\Big\langle\Big| P_\alpha \to \frac{61\,316\,911\,076\,911\,789 - 2\sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{122\,633\,822\,153\,823\,578},$$
$$P_{1,\alpha} \to \frac{197\,818\,302\,948\,040\,811 + 3\sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{375\,985\,460\,508\,686\,570},$$
$$P_{1,\beta} \to \frac{3\left(59\,389\,052\,520\,215\,253 + \sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}\right)}{375\,985\,460\,508\,686\,570},$$
$$P_{2,\alpha} \to \frac{23\,470\,130\,463\,167\,807 + \sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{136\,288\,278\,313\,807\,950},$$
$$P_{2,\beta} \to \frac{112\,818\,147\,850\,640\,143 + \sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{136\,288\,278\,313\,807\,950},$$
$$P_{3,\alpha} \to \frac{209\,373\,072\,434\,759\,059 + 3\sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{412\,438\,820\,078\,205\,386},$$
$$P_{3,\beta} \to \frac{203\,065\,747\,643\,446\,327 + 3\sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{412\,438\,820\,078\,205\,386}\Big|\Big\rangle,$$
$$\Big\langle\Big| P_\alpha \to \frac{61\,316\,911\,076\,911\,789 + 2\sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{122\,633\,822\,153\,823\,578},$$
$$P_{1,\alpha} \to \frac{197\,818\,302\,948\,040\,811 - 3\sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{375\,985\,460\,508\,686\,570},$$
$$P_{1,\beta} \to \frac{3\left(59\,389\,052\,520\,215\,253 - \sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}\right)}{375\,985\,460\,508\,686\,570},$$
$$P_{2,\alpha} \to \frac{23\,470\,130\,463\,167\,807 - \sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{136\,288\,278\,313\,807\,950},$$
$$P_{2,\beta} \to \frac{112\,818\,147\,850\,640\,143 - \sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{136\,288\,278\,313\,807\,950},$$
$$P_{3,\alpha} \to \frac{209\,373\,072\,434\,759\,059 - 3\sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{412\,438\,820\,078\,205\,386},$$
$$P_{3,\beta} \to \frac{203\,065\,747\,643\,446\,327 - 3\sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{412\,438\,820\,078\,205\,386}\Big|\Big\rangle\Big\}$$

Incredible! It is astonishing that more ML experts do not know about this. Consider what just happened. In essentially instantaneous time you are able evaluate these three noisy binary classifiers. Let’s confirm that by timing the evaluation Mathematica carries out for us.

In[63]:=
```
Timing[Solve[
   (MakeIndependentVotingIdeal[{1, 2, 3}] // Map[(# == 0) &, #] &) /.
    evaluationDataSketch,
   evaluationVariables];]
```
Out[63]=
```
{0.027575, Null}
```

And look at the information we are immediately getting on the quality of the evaluation. Remember that the exact grades for these classifiers are integer ratios. The evaluation we got assuming that they were error independent is not telling us that. Consider the algebraic evaluator's answer for the prevalence of the least likely label in the UCI Adult dataset - the alpha/0 label. We get two point answers for where the true prevalence must be,

In[65]:=
```
Map[#[Pα] &, independentModelEvaluation]
```
Out[65]=

$$\left\{\frac{61\,316\,911\,076\,911\,789-2\sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{122\,633\,822\,153\,823\,578},\ \frac{61\,316\,911\,076\,911\,789+2\sqrt{416\,629\,916\,124\,502\,529\,599\,755\,188\,035\,849}}{122\,633\,822\,153\,823\,578}\right\}$$

So the evaluation ideal for error independent binary classifiers is represented in sample statistics space by a geometrical object that consists of two point solutions to the evaluation statistics we are looking for. This is the geometrical representation of the decoding ambiguity of evaluation. In fact, it is never possible to know, with true certainty, absent any other outside knowledge, the ground truth values for the evaluation. You can take this computation to be the proof of that. If error independent judges cannot do it, no judges will ever do so either.

This decoding ambiguity freezes academic ML researchers. The fact that two, not one solution, is returned by the evaluator confuses people that are not familiar with another area that shows a deep connection between pure mathematics and engineering: error-correcting codes. If you understand how error-correcting codes also have decoding ambiguity and yet are ubiquitous in signal processing engineering, you understand how decoding ambiguity in algebraic evaluation is a non-problem for engineers. This point is hard for computer science academics to get. Later, I'll return to the practical significance that algebraic evaluation enables error-correcting algorithms. For now note that I have so far shown connections of Algebraic Evaluation with three other areas of research interest: algebraic geometry, data streaming algorithms and now error-correcting codes. I am not done. Algebraic evaluation has more goodies in store for us.

Algebraic evaluation connects mathematical field theory to AI safety. The number field you use to carry your AI safety computations matters. Algebraic numbers are more useful than real numbers. The evaluation above is a simple demonstration of this. The error independent evaluation model must be wrong. We can tell that it is wrong because it did not give us integer ratios. There are unresolved square roots in its output. This is gold in safety engineering contexts.

Knowing that you are flying blind is priceless

How good is the numerical estimate by the error independent evaluator?

So let’s see how well the error independent evaluator estimated the accuracy of the UCI Adult binary classifiers.

In[66]:=
```
N@independentModelEvaluation
```

Out[66]=

$$\{\langle| P_\alpha \to 0.167114, P_{1,\alpha} \to 0.688997, P_{1,\beta} \to 0.636731, P_{2,\alpha} \to 0.321977, P_{2,\beta} \to 0.977558, P_{3,\alpha} \to 0.656116, P_{3,\beta} \to 0.640823 |\rangle, \langle| P_\alpha \to 0.832886, P_{1,\alpha} \to 0.363269, P_{1,\beta} \to 0.311003, P_{2,\alpha} \to 0.0224423, P_{2,\beta} \to 0.678023, P_{3,\alpha} \to 0.359177, P_{3,\beta} \to 0.343884 |\rangle\}$$

Here is where we get to see why decoding is many times trivial in the real world. This is no different than decoding of error-correcting codes being trivial in the real world. You do here exactly what you would do for error correcting codes. When you decode in error-correcting codes you have multiple possible solutions if a detectable error has occurred. The default choice is ALWAYS least bit errors. Error-correcting codes are engineered so this is almost always true. It works for inter-planetary communications and your computer. Same thing in Algebraic Evaluation. Its engineering context usually has enough outside context to allow you to decode the right evaluation for noisy binary classifiers.

Let’s talk through one example of how this decoding choice could be done - you have very good knowledge of the prevalence of labels. For example, you could trying to evaluate DNA sequencers and you want to estimate their error rates. Since you are most likely handling Earth DNA, it would be a simple calibrating step to check which prevalence solution is closer to the known frequency distribution of DNA bases. That simple.

Another example applies to a business that is using AI to discover a rare, valuable thing. Like Google trying to find pages where users will click on ads. The click rate in the internet is about 1/1000 on a good website. The “it will be clicked” label will be rare in whatever ad campaign you run. This is the case in UCI Adult. The 0 label is the rare one. So we just choose the first solution.

In[71]:=
```
evaluationAlgebraicGuess = First@independentModelEvaluation;
N@evaluationAlgebraicGuess
```

Out[72]=

$$\langle| P_\alpha \to 0.167114, P_{1,\alpha} \to 0.688997, P_{1,\beta} \to 0.636731, P_{2,\alpha} \to 0.321977, P_{2,\beta} \to 0.977558, P_{3,\alpha} \to 0.656116, P_{3,\beta} \to 0.640823 |\rangle$$

Now let’s look at the ground truth for this single run of three binary classifiers on a UCI Adult test set.

In[67]:=
```
N@evaluationGroundTruth
```

Out[67]=

$\langle| P_{\alpha} \to 0.154362, P_{1,\alpha} \to 0.657113, P_{2,\alpha} \to 0.221558, P_{3,\alpha} \to 0.834535, P_{1,\beta} \to 0.625999, P_{2,\beta} \to 0.95471, P_{3,\beta} \to 0.668913, \Gamma_{1,2,\alpha} \to 0.00844698, \Gamma_{1,3,\alpha} \to 0.000412003, \Gamma_{2,3,\alpha} \to -0.008179, \Gamma_{1,2,\beta} \to 0.00681382, \Gamma_{1,3,\beta} \to -0.00615695, \Gamma_{2,3,\beta} \to 0.00243418, \Gamma_{1,2,3,\alpha} \to 0.00246056, \Gamma_{1,2,3,\beta} \to -0.000202868 |\rangle$

Wow! Look at the closeness of the independent model estimates. Let’s pretty print the comparison.

In[87]:=
```
Column[{"Algebraic evaluation,assuming error
    independendence, of three binary classifiers on UCI Adult",
  Grid[Prepend[Transpose@{evaluationVariables, Map[{evaluationGroundTruth@#,
       N@evaluationGroundTruth@#} &, evaluationVariables],
     Map[{evaluationAlgebraicGuess@#, N@evaluationAlgebraicGuess@#} &,
      evaluationVariables]}, {"Evaluation Statistic", "Correct", "Estimated"}],
   Dividers → All]}, Dividers → All, Alignment → Center]
```

Out[87]=

Algebraic evaluation,assuming error independendence, of three binary classifiers on UCI Adult

Evaluation Statistic	Correct	Estimated
P_{α}	$\left\{\frac{5687}{36842}, 0.154362\right\}$	$\left\{\left(61316911076911789 - 2\times\sqrt{416629916124502529599755188035849}\right)/122633822153823578, 0.167114\right\}$
$P_{1,\alpha}$	$\left\{\frac{3737}{5687}, 0.657113\right\}$	$\left\{\frac{197818302948040811+3\sqrt{416629916124502529599755188035849}}{375985460508686570}, 0.688997\right\}$
$P_{1,\beta}$	$\left\{\frac{6501}{10385}, 0.625999\right\}$	$\left\{\frac{3\left(59389052520215253+\sqrt{416629916124502529599755188035849}\right)}{375985460508686570}, 0.636731\right\}$
$P_{2,\alpha}$	$\left\{\frac{1260}{5687}, 0.221558\right\}$	$\left\{\frac{23470130463167807+\sqrt{416629916124502529599755188035849}}{136288278313807950}, 0.321977\right\}$
$P_{2,\beta}$	$\left\{\frac{29744}{31155}, 0.95471\right\}$	$\left\{\frac{112818147850640143+\sqrt{416629916124502529599755188035849}}{136288278313807950}, 0.977558\right\}$
$P_{3,\alpha}$	$\left\{\frac{4746}{5687}, 0.834535\right\}$	$\left\{\frac{209373072434759059+3\sqrt{416629916124502529599755188035849}}{412438820078205386}, 0.656116\right\}$
$P_{3,\beta}$	$\left\{\frac{4168}{6231}, 0.668913\right\}$	$\left\{\frac{203065747643446327+3\sqrt{416629916124502529599755188035849}}{412438820078205386}, 0.640823\right\}$

The above table illustrates why algebraic numbers are more useful in an evaluation context than real ones. All real numbers look the same. Here we see that the correct answers look very much like the algebraic evaluation output when you represent them as floating point numbers. The difference between the correct performance of the classifiers (always an integer ratio) and the output of the error independent evaluator are obvious when you express them as algebraic numbers.

The exact polynomial formulation of voting patterns for arbitrarily correlated classifiers

The main topic of this notebook is the error independent evaluator. It is the easiest algebraic evaluator you can build. But the reader should not think that algebraic evaluation is inexact when you have correlated classifiers. Exact polynomial formulations of arbitrarily correlated classifiers exist. This is significant for anyone that worries about AI safety. There are no unknown unknowns in evaluations of finite samples. Unlike the much harder task of training noisy judges, evaluation of these judges is much easier. The algebraic evaluator is dumb. It has no knowledge of the world or the experts. It just has to estimate sample statistics. But these statistics exist in a finite space that can be universally characterized for ALL evaluations. Estimating a sample statistic is not as hard as, say, making future predictions about that statistic. All the statistics needed for three correlated binary classifiers are:

In[89]:=
```
Keys@evaluationGroundTruth // Grid[{#}, Dividers → All] &
```

Out[89]=

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

Here is the polynomial set, based on these statistics, that generates the evaluation ideal for arbitrarily correlated classifiers,

In[328]:=

unknownSideOfEvaluationIdealCorrelatedBinaryClassifiers =

$$
\begin{aligned}
\{&P_\alpha \left(P_{1,\alpha} P_{2,\alpha} P_{3,\alpha} + P_{3,\alpha} \Gamma_{1,2,\alpha} + P_{2,\alpha} \Gamma_{1,3,\alpha} + P_{1,\alpha} \Gamma_{2,3,\alpha} + \Gamma_{1,2,3,\alpha}\right) + P_\beta \left(\left(1 - P_{1,\beta}\right) \left(1 - P_{2,\beta}\right)\right. \\
&\quad \left.\left(1 - P_{3,\beta}\right) + \left(1 - P_{3,\beta}\right) \Gamma_{1,2,\beta} + \left(1 - P_{2,\beta}\right) \Gamma_{1,3,\beta} + \left(1 - P_{1,\beta}\right) \Gamma_{2,3,\beta} - \Gamma_{1,2,3,\beta}\right), \\
&P_\alpha \left(P_{1,\alpha} P_{2,\alpha} \left(1 - P_{3,\alpha}\right) + \left(1 - P_{3,\alpha}\right) \Gamma_{1,2,\alpha} - P_{2,\alpha} \Gamma_{1,3,\alpha} - P_{1,\alpha} \Gamma_{2,3,\alpha} - \Gamma_{1,2,3,\alpha}\right) + \\
&\quad P_\beta \left(\left(1 - P_{1,\beta}\right) \left(1 - P_{2,\beta}\right) P_{3,\beta} + P_{3,\beta} \Gamma_{1,2,\beta} - \left(1 - P_{2,\beta}\right) \Gamma_{1,3,\beta} - \left(1 - P_{1,\beta}\right) \Gamma_{2,3,\beta} + \Gamma_{1,2,3,\beta}\right), \\
&P_\alpha \left(P_{1,\alpha} \left(1 - P_{2,\alpha}\right) P_{3,\alpha} - P_{3,\alpha} \Gamma_{1,2,\alpha} + \left(1 - P_{2,\alpha}\right) \Gamma_{1,3,\alpha} - P_{1,\alpha} \Gamma_{2,3,\alpha} - \Gamma_{1,2,3,\alpha}\right) + \\
&\quad P_\beta \left(\left(1 - P_{1,\beta}\right) P_{2,\beta} \left(1 - P_{3,\beta}\right) - \left(1 - P_{3,\beta}\right) \Gamma_{1,2,\beta} + P_{2,\beta} \Gamma_{1,3,\beta} - \left(1 - P_{1,\beta}\right) \Gamma_{2,3,\beta} + \Gamma_{1,2,3,\beta}\right), \\
&P_\alpha \left(P_{1,\alpha} \left(1 - P_{2,\alpha}\right) \left(1 - P_{3,\alpha}\right) - \left(1 - P_{3,\alpha}\right) \Gamma_{1,2,\alpha} - \left(1 - P_{2,\alpha}\right) \Gamma_{1,3,\alpha} + P_{1,\alpha} \Gamma_{2,3,\alpha} + \Gamma_{1,2,3,\alpha}\right) + \\
&\quad P_\beta \left(\left(1 - P_{1,\beta}\right) P_{2,\beta} P_{3,\beta} - P_{3,\beta} \Gamma_{1,2,\beta} - P_{2,\beta} \Gamma_{1,3,\beta} + \left(1 - P_{1,\beta}\right) \Gamma_{2,3,\beta} - \Gamma_{1,2,3,\beta}\right), \\
&P_\alpha \left(\left(1 - P_{1,\alpha}\right) P_{2,\alpha} P_{3,\alpha} - P_{3,\alpha} \Gamma_{1,2,\alpha} - P_{2,\alpha} \Gamma_{1,3,\alpha} + \left(1 - P_{1,\alpha}\right) \Gamma_{2,3,\alpha} - \Gamma_{1,2,3,\alpha}\right) + \\
&\quad P_\beta \left(P_{1,\beta} \left(1 - P_{2,\beta}\right) \left(1 - P_{3,\beta}\right) - \left(1 - P_{3,\beta}\right) \Gamma_{1,2,\beta} - \left(1 - P_{2,\beta}\right) \Gamma_{1,3,\beta} + P_{1,\beta} \Gamma_{2,3,\beta} + \Gamma_{1,2,3,\beta}\right), \\
&P_\alpha \left(\left(1 - P_{1,\alpha}\right) P_{2,\alpha} \left(1 - P_{3,\alpha}\right) - \left(1 - P_{3,\alpha}\right) \Gamma_{1,2,\alpha} + P_{2,\alpha} \Gamma_{1,3,\alpha} - \left(1 - P_{1,\alpha}\right) \Gamma_{2,3,\alpha} + \Gamma_{1,2,3,\alpha}\right) + \\
&\quad P_\beta \left(P_{1,\beta} \left(1 - P_{2,\beta}\right) P_{3,\beta} - P_{3,\beta} \Gamma_{1,2,\beta} + \left(1 - P_{2,\beta}\right) \Gamma_{1,3,\beta} - P_{1,\beta} \Gamma_{2,3,\beta} - \Gamma_{1,2,3,\beta}\right), \\
&P_\alpha \left(\left(1 - P_{1,\alpha}\right) \left(1 - P_{2,\alpha}\right) P_{3,\alpha} + P_{3,\alpha} \Gamma_{1,2,\alpha} - \left(1 - P_{2,\alpha}\right) \Gamma_{1,3,\alpha} - \left(1 - P_{1,\alpha}\right) \Gamma_{2,3,\alpha} + \Gamma_{1,2,3,\alpha}\right) + \\
&\quad P_\beta \left(P_{1,\beta} P_{2,\beta} \left(1 - P_{3,\beta}\right) + \left(1 - P_{3,\beta}\right) \Gamma_{1,2,\beta} - P_{2,\beta} \Gamma_{1,3,\beta} - P_{1,\beta} \Gamma_{2,3,\beta} - \Gamma_{1,2,3,\beta}\right), \\
&P_\alpha \left(\left(1 - P_{1,\alpha}\right) \left(1 - P_{2,\alpha}\right) \left(1 - P_{3,\alpha}\right) + \left(1 - P_{3,\alpha}\right) \Gamma_{1,2,\alpha} + \left(1 - P_{2,\alpha}\right) \Gamma_{1,3,\alpha} + \left(1 - P_{1,\alpha}\right) \Gamma_{2,3,\alpha} - \right. \\
&\quad \left.\Gamma_{1,2,3,\alpha}\right) + P_\beta \left(P_{1,\beta} P_{2,\beta} P_{3,\beta} + P_{3,\beta} \Gamma_{1,2,\beta} + P_{2,\beta} \Gamma_{1,3,\beta} + P_{1,\beta} \Gamma_{2,3,\beta} + \Gamma_{1,2,3,\beta}\right)\};
\end{aligned}
$$

We kept P_α and P_β to simplify the math. These are related by $P_\alpha + P_\beta = 1$ so we'll get rid of the P_β variable when we do our computation. We have this unknown side completely written in a gibberish of sample statistics. What is the value of all these polynomials when we plug in the true evaluation values for our working UCI Adult evaluation run?

In[329]:=

```
(unknownSideOfEvaluationIdealCorrelatedBinaryClassifiers /. {Pβ → (1 - Pα)}) /.
 evaluationGroundTruth
```

Out[329]=

$$\left\{\frac{493}{18421}, \frac{315}{18421}, \frac{5801}{36842}, \frac{3986}{18421}, \frac{281}{18421}, \frac{493}{36842}, \frac{3856}{18421}, \frac{6343}{18421}\right\}$$

This is encouraging. We know all the voting pattern frequencies are integer ratios. Do these polynomials get the exact right answer for these integer ratios? Yes. That is the claim that in Evaluation Land all evaluation ideals are exact.

In[330]:=

```
evaluationDataSketch
```

Out[330]=

$$\left\{f_{\alpha,\alpha,\alpha} \to \frac{493}{18421}, f_{\alpha,\alpha,\beta} \to \frac{315}{18421}, f_{\alpha,\beta,\alpha} \to \frac{5801}{36842}, f_{\alpha,\beta,\beta} \to \frac{3986}{18421}, f_{\beta,\alpha,\alpha} \to \frac{281}{18421}, f_{\beta,\alpha,\beta} \to \frac{493}{36842}, f_{\beta,\beta,\alpha} \to \frac{3856}{18421}, f_{\beta,\beta,\beta} \to \frac{6343}{18421}\right\}$$

Nothing like this exists in Training Land. It is impossible to devise training algorithms based on probability theory that are exact representations of all possible future data processed by AI agents. Not so in Evaluation Land. And for a trivial reason - methods of moments are always possible with finite statistics. Why is the AI community unaware of this trivial fact? Why are these exact polynomial representations not part of any textbook that claims to explain Machine Learning Theory? Evaluation is the forgotten twin of Learning. Learning is training + evaluation.

Consider what this means theoretically. We have the exact algebraic object that explains ALL evaluations of arbitrarily correlated classifiers. The unresolved problems in Algebraic Evaluation are not here. Exact representations will always be possible in the same way that moment expansions are always possible when describing sample statistics. The unresolved problems in Algebraic Evaluation lie in understanding the evaluation variety - the surface in sample statistics space that is universally guaranteed to contain the true evaluation values.

Computing the evaluation variety corresponding to the three error-independent evaluation ideal

We now come to the hard part of this notebook. One full of the math and terminology of algebraic geometry. This is a mathematical topic that few in the AI community know. Likewise in the statistics community. But there is an exception in Statistics Land. There is a well defined field concerned with the use of algebraic geometry - algebraic statistics. Algebraic evaluation is a sister to algebraic statistics. The former deals with the algebraic geometry of evaluating noisy judges on finite samples, the later, with the algebraic geometry of the mathematics of infinite samples. Given their affinity, in the future Algebraic Statistics books will contain chapters on the topics discussed here if they are concerned with the algebraic geometry of evaluating noisy judges.

Our goal is an algebraic formulation of the variety that allows us to understand its geometrical structure. The way to do is to compute the Groebner basis for the evaluation ideal. So let's do it. As you'll see, it is going to be crazy long.

In[332]:=

```
gb = GroebnerBasis[MakeIndependentVotingIdeal[{1, 2, 3}], evaluationVariables]
```

Out[332]=

$$\{-1 + f_{\alpha,\alpha,\alpha} + f_{\alpha,\alpha,\beta} + f_{\alpha,\beta,\alpha} + f_{\alpha,\beta,\beta} + f_{\beta,\alpha,\alpha} + f_{\beta,\alpha,\beta} + f_{\beta,\beta,\alpha} + f_{\beta,\beta,\beta},$$
$$\cdots 39 \cdots, -P_\alpha + P_\alpha P_{1,\alpha} - P_{1,\beta} + P_\alpha P_{1,\beta} + f_{\beta,\alpha,\alpha} + f_{\beta,\alpha,\beta} + f_{\beta,\beta,\alpha} + f_{\beta,\beta,\beta}\}$$

Size in memory: 1 MB | + Show more | Show all | Iconize | Store full expression in notebook

Understanding the Groebner basis for three error-independent binary classifiers

It is going to take some work to figure out the algebraic mess we get from the Groebner basis computation. This is the hard work that will result in simple algebraic expressions making possible the computation of the independent algebraic evaluator in Python code. Let's start simple and look at the very first equation returned by Mathematica's computation,

In[333]:=

```
First@gb
```

Out[333]=

$$-1 + f_{\alpha,\alpha,\alpha} + f_{\alpha,\alpha,\beta} + f_{\alpha,\beta,\alpha} + f_{\alpha,\beta,\beta} + f_{\beta,\alpha,\alpha} + f_{\beta,\alpha,\beta} + f_{\beta,\beta,\alpha} + f_{\beta,\beta,\beta}$$

When we constructed the evaluation ideal for the classifiers, we never told Mathematica what those voting pattern frequencies meant. The algebraic computation has no semantic meaning except its own algebra. It is us who are applying that mathematics in a specific scientific context - the evaluation of noisy judges on unlabeled test data. Because those frequencies were constructed from the voting patterns of the judges, they must sum to one. This first equation is essentially asserting that fact as an essential mathematical requirement for there to actually exist ANY surface in sample parameter space. At the same time, the mathematics does not require that any of the f's be positive. This is an additional constraint that relates to the application context of this algebra - counts of observed events cannot be negative.

Let's take a peek at all the equations at once by making a survey of the sample statistics involved in each equation in the computed Groebner basis.

In[334]:=

```
Map[Variables, gb] //
  Map[Cases[#, Except[f__]] &, #] & //
 Column[#, Dividers → All] &
```

Out[334]=

{}
$\{P_{3,\beta}\}$

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

$\boxed{\{P_{\alpha},\ P_{1,\alpha},\ P_{1,\beta}\}}$

Interesting, the 2nd equation involves $P_{3,\beta}$ alone. This means that there is a polynomial for it that we can use to see how $P_{3,\beta}$ varies along the evaluation variety. Here is the polynomial.

In[335]:=

```
gb[[2]]
```

Out[335]=

$$P_{3,\beta}^2 f_{\alpha,\beta,\alpha} f_{\beta,\alpha,\alpha} - P_{3,\beta} f_{\alpha,\beta,\beta} f_{\beta,\alpha,\alpha} + P_{3,\beta}^2 f_{\alpha,\beta,\beta} f_{\beta,\alpha,\alpha} - P_{3,\beta} f_{\alpha,\beta,\alpha} f_{\beta,\alpha,\beta} + P_{3,\beta}^2 f_{\alpha,\beta,\alpha} f_{\beta,\alpha,\beta} + f_{\alpha,\beta,\beta} f_{\beta,\alpha,\beta} - 2 P_{3,\beta} f_{\alpha,\beta,\beta} f_{\beta,\alpha,\beta} + P_{3,\beta}^2 f_{\alpha,\beta,\beta} f_{\beta,\alpha,\beta} - P_{3,\beta}^2 f_{\beta,\beta,\alpha} + P_{3,\beta} f_{\alpha,\alpha,\beta} f_{\beta,\beta,\alpha} + P_{3,\beta}^2 f_{\alpha,\beta,\alpha} f_{\beta,\beta,\alpha} + P_{3,\beta}^2 f_{\alpha,\beta,\beta} f_{\beta,\beta,\alpha} + P_{3,\beta}^2 f_{\beta,\alpha,\alpha} f_{\beta,\beta,\alpha} + P_{3,\beta}^2 f_{\beta,\alpha,\beta} f_{\beta,\beta,\alpha} + P_{3,\beta}^2 f_{\beta,\beta,\alpha}^2 + P_{3,\beta} f_{\beta,\beta,\beta} - P_{3,\beta}^2 f_{\beta,\beta,\beta} - f_{\alpha,\alpha,\beta} f_{\beta,\beta,\beta} + P_{3,\beta} f_{\alpha,\alpha,\beta} f_{\beta,\beta,\beta} - P_{3,\beta} f_{\alpha,\beta,\alpha} f_{\beta,\beta,\beta} + P_{3,\beta}^2 f_{\alpha,\beta,\alpha} f_{\beta,\beta,\beta} - P_{3,\beta} f_{\alpha,\beta,\beta} f_{\beta,\beta,\beta} + P_{3,\beta}^2 f_{\alpha,\beta,\beta} f_{\beta,\beta,\beta} - P_{3,\beta} f_{\beta,\alpha,\alpha} f_{\beta,\beta,\beta} + P_{3,\beta}^2 f_{\beta,\alpha,\alpha} f_{\beta,\beta,\beta} - P_{3,\beta} f_{\beta,\alpha,\beta} f_{\beta,\beta,\beta} + P_{3,\beta}^2 f_{\beta,\alpha,\beta} f_{\beta,\beta,\beta} - P_{3,\beta} f_{\beta,\beta,\alpha} f_{\beta,\beta,\beta} + 2 P_{3,\beta}^2 f_{\beta,\beta,\alpha} f_{\beta,\beta,\beta} - P_{3,\beta} f_{\beta,\beta,\beta}^2 + P_{3,\beta}^2 f_{\beta,\beta,\beta}^2$$

It looks like a quadratic. Let’s collect the terms using $P_{3,\beta}$

In[337]:=

```
Collect[gb[[2]], P3,β, Factor]
```

Out[337]=

$$f_{\alpha,\beta,\beta} f_{\beta,\alpha,\beta} - f_{\alpha,\alpha,\beta} f_{\beta,\beta,\beta} + P_{3,\beta} \left(- f_{\alpha,\beta,\beta} f_{\beta,\alpha,\alpha} - f_{\alpha,\beta,\alpha} f_{\beta,\alpha,\beta} - 2 f_{\alpha,\beta,\beta} f_{\beta,\alpha,\beta} + f_{\alpha,\alpha,\beta} f_{\beta,\beta,\alpha} + f_{\beta,\beta,\beta} + f_{\alpha,\alpha,\beta} f_{\beta,\beta,\beta} - f_{\alpha,\beta,\alpha} f_{\beta,\beta,\beta} - f_{\alpha,\beta,\beta} f_{\beta,\beta,\beta} - f_{\beta,\alpha,\alpha} f_{\beta,\beta,\beta} - f_{\beta,\alpha,\beta} f_{\beta,\beta,\beta} - f_{\beta,\beta,\alpha} f_{\beta,\beta,\beta} - f_{\beta,\beta,\beta}^2\right) + P_{3,\beta}^2 \left(f_{\alpha,\beta,\alpha} f_{\beta,\alpha,\alpha} + f_{\alpha,\beta,\beta} f_{\beta,\alpha,\alpha} + f_{\alpha,\beta,\alpha} f_{\beta,\alpha,\beta} + f_{\alpha,\beta,\beta} f_{\beta,\alpha,\beta} - f_{\beta,\beta,\alpha} + f_{\alpha,\beta,\alpha} f_{\beta,\beta,\alpha} + f_{\alpha,\beta,\beta} f_{\beta,\beta,\alpha} + f_{\beta,\alpha,\alpha} f_{\beta,\beta,\alpha} + f_{\beta,\alpha,\beta} f_{\beta,\beta,\alpha} + f_{\beta,\beta,\alpha}^2 - f_{\beta,\beta,\beta} + f_{\alpha,\beta,\alpha} f_{\beta,\beta,\beta} + f_{\alpha,\beta,\beta} f_{\beta,\beta,\beta} + f_{\beta,\alpha,\alpha} f_{\beta,\beta,\beta} + f_{\beta,\alpha,\beta} f_{\beta,\beta,\beta} + 2 f_{\beta,\beta,\alpha} f_{\beta,\beta,\beta} + f_{\beta,\beta,\beta}^2\right)$$

The appearance of quadratics like this for a single of the evaluation statistics is the beginning of the proof that the evaluation variety connected to the independent algebraic evaluator consists of two points in the 7 dimensional space of these evaluation statistics. There is a solution ladder, if you will, the elimination ladder, that starts using this quadratic and then using the solutions for $P_{3,\beta}$ to get solutions for the other statistics. We want to climb that elimination ladder but all this algebra still looks crazy. Let’s start by forgetting about this Groebner basis and see if we can get another one for our evaluation ideal that plays nicer. It should be possible to get a quadratic for P_{α}