Bayesian Method for Calculating Efficiency Uncertainty

Xiaoning Wang

Calculating Efficiency Uncertainty

- Problem: use regular propagation of errors cannot account for that the total number of events N and number of passed events k are correlated.
- Two commonly used methods:

- Binomial errors $\frac{\sqrt{k(\frac{1-k}{N})}}{N}$

$$\begin{aligned}\sigma_k &= \sqrt{\mathrm{var}(k)} \\ &= \sqrt{\epsilon(1-\epsilon)N}.\end{aligned}$$

$$\delta\epsilon' = (1/N)\sqrt{k(1-k/N)}$$

$\varepsilon' = k/N$: estimate of true efficiency
$\delta\varepsilon' = \sigma_k/N$: uncertainty of estimated efficiency

Problem: gives symmetric uncertainty, but k=N can have nonzero lower uncertainty, and k=0 can have nonzero upper uncertainty.
Source:
https://home.fnal.gov/~paterno/images/effic.pdf

- Bayesian errors

$$P(\epsilon|k;N) = \frac{1}{norm} \times P(k|\epsilon;N) \times Prior(\epsilon)$$

$$P(k|\epsilon;N) = Binomial(N,k) \times \epsilon^k \times (1-\epsilon)^{N-k} \ldots \text{ binomial distribution}$$

$$Prior(\epsilon) = \frac{1}{B(\alpha,\beta)} \times \epsilon^{\alpha-1} \times (1-\epsilon)^{\beta-1} \equiv Beta(\epsilon;\alpha,\beta)$$

$$\Rightarrow P(\epsilon|k;N) = \frac{1}{norm'} \times \epsilon^{k+\alpha-1} \times (1-\epsilon)^{N-k+\beta-1} \equiv Beta(\epsilon;k+\alpha,N-k+\beta)$$

https://root.cern.ch/doc/master/classTEfficiency.html
Step by step explanation see next slide

Bayesian Method

$$P(\epsilon|k;N) = \frac{1}{norm} \times P(k|\epsilon;N) \times Prior(\epsilon)$$

Bayes' Theorem

$$P(k|\epsilon;N) = Binomial(N,k) \times \epsilon^k \times (1-\epsilon)^{N-k} \ldots \ binomial\ distribution$$

$$Prior(\epsilon) = \frac{1}{B(\alpha,\beta)} \times \epsilon^{\alpha-1} \times (1-\epsilon)^{\beta-1} \equiv Beta(\epsilon;\alpha,\beta)$$

In our case where we have no prior knowledge of true efficiency, α and β are set to 1, that is uniform distribution between 0 and 1. And we set it 0 probability outside of this range for normalization.

$$\Rightarrow P(\epsilon|k;N) = \frac{1}{norm'} \times \epsilon^{k+\alpha-1} \times (1-\epsilon)^{N-k+\beta-1} \equiv Beta(\epsilon;k+\alpha,N-k+\beta)$$

- ROOT code used:

```
TGraphAsymmErrors *new_eff2 = new TGraphAsymmErrors(hMatched, hTotal,"cl=0.683 b(1,1) mode");
```

Confidence level: 0.683

b(1,1): prior probability distribution of ϵ $Prior(\epsilon)$ is set as uniform from 0 to 1 $Beta(\epsilon; 1,1)$.

mode: Use mode of the posterior probability distribution of ϵ $P(\epsilon|k;N)$ for estimate of centroid ϵ value.

Mode of a beta distribution $Beta(\varepsilon;x,y)$ for x, y >1 is given by the formula $\frac{x-1}{x+y-2}$ (https://en.wikipedia.org/wiki/Beta_distribution#Definitions). And in this case it is $\hat{\varepsilon} = \frac{k+\alpha-1}{N+\alpha+\beta-2}$

- With the probability distribution given by last step, ROOT numerically finds the shortest interval boundaries that integrates to the confidence level and subtract the mode value as upper and lower efficiency uncertainty.

- Source code implemented in TGraphAsymmErrors and TEfficiency
- https://root.cern.ch/doc/master/TEfficiency_8cxx_source.html#l01249

How uncertainty in hMatched and hTotal is used (Part 1)

```
tw = total->GetBinContent(b);
tw2 = (total->GetSumw2()->fN > 0) ? total->GetSumw2()->At(b) : tw;
pw = pass->GetBinContent(b);
pw2 = (pass->GetSumw2()->fN > 0) ? pass->GetSumw2()->At(b) : pw;
```

tw: number of total events.

pw: number of passed/matched events.

tw2 and pw2 as squares of weight for total and passed/matched.

Note:

In case of unweighted events where sumW2 are set before filling, tw2 = tw = bin content.

In case the error bar of binning is set manually instead of by filling, ***tw2 = error^2***

The latter case was used in case of extracting passed and total using fitting to data, where error is the error of fitting.

How uncertainty in hMatched and hTotal is used (Part 2)

```
if ((bEffective && !bPoissonRatio) && tw2 <= 0) {
   // case of bins with zero errors
   eff = pw/tw;
   low = eff; upper = eff;
}
```

In case of tw2 being 0, that is, error bars are manually set to be 0, efficiency also has no error bar.
bEffective is the true and bPoissonRatio is false when we are trying to calculate efficiency instead of ratio of two Poisson means (option "pois").

```
if (bEffective && !bPoissonRatio) {
   // tw/tw2 re-normalize the weights
   double norm = tw/tw2;  // case of tw2 = 0 is treated above
   aa =  pw * norm + alpha;
   bb =  (tw - pw) * norm + beta;
}
if (usePosteriorMode)
   eff = TEfficiency::BetaMode(aa,bb);

if (useShortestInterval) {
   TEfficiency::BetaShortestInterval(conf,aa,bb,low,upper);
}
```

alpha and beta is given by "b(alpha, beta)" in the option

usePosteriorMode is set by "mode" in the option
userShortestInterval is by default set to true when using "mode" option, see slide 9 for the author (Marc Paterno)'s reason for recommending using shortest confidence interval method.

$$P(\epsilon|k;N) = \frac{1}{norm'} \times \epsilon^{k+\alpha-1} \times (1-\epsilon)^{N-k+\beta-1} \equiv Beta(\epsilon; k+\alpha, N-k+\beta)$$

Recall this is our formula for posterior probability distribution

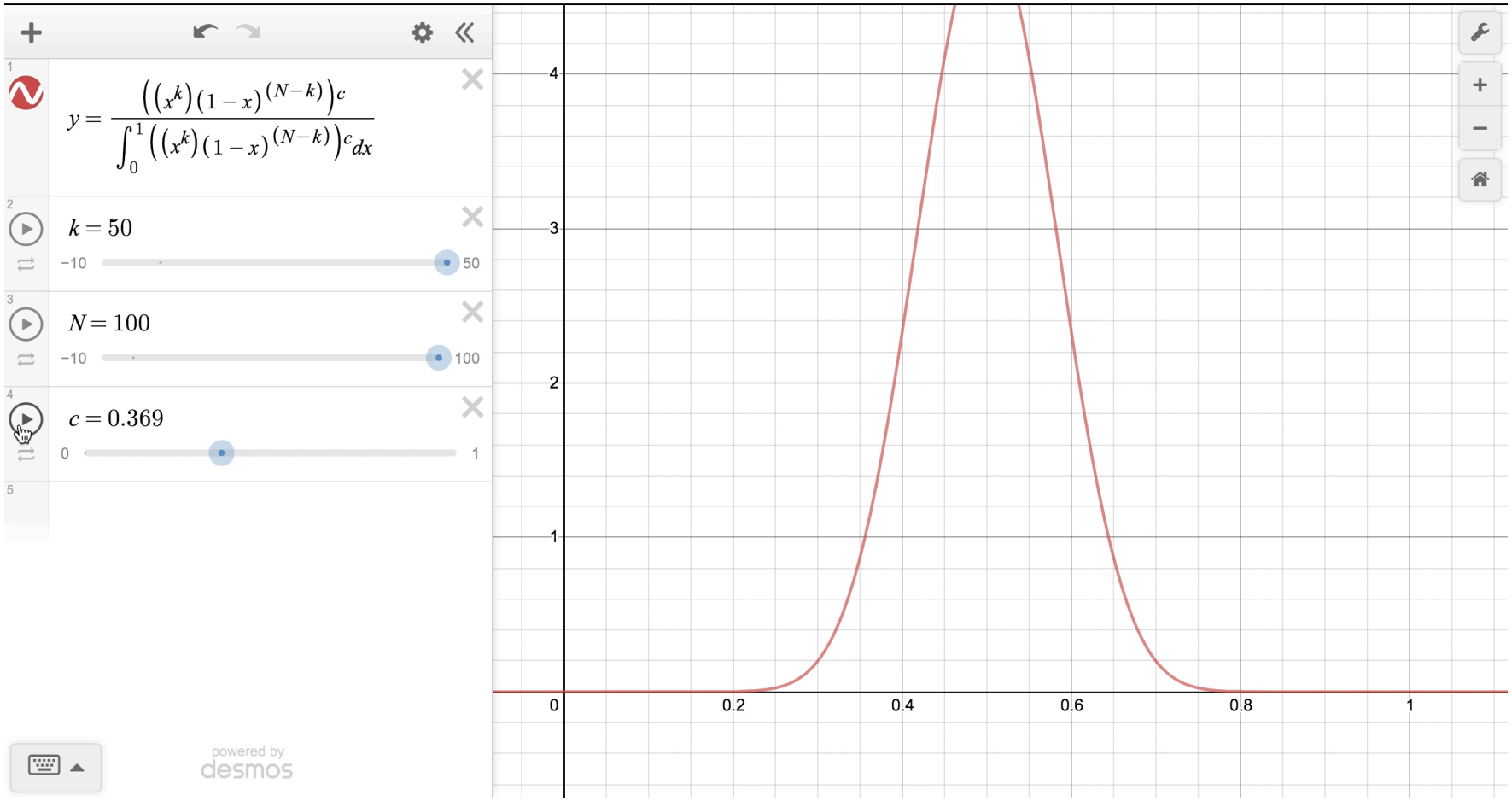
In case of tw2 being set manually, a re-normalization factor (tw/tw2), $\frac{N}{\sigma_N^2}$, is applied to k and $N-k$.

$k \to k \times \frac{N}{\sigma_N^2}$, $(N-k) \to (N-k) \times \frac{N}{\sigma_N^2}$

Centroid value is not affected by this renormalization for $\alpha = \beta = 1$, for its calculated using $\hat{\varepsilon} = \frac{k+\alpha-1}{N+\alpha+\beta-2}$

How uncertainty in hMatched and hTotal is used (Part 3)

- The renormalized posterior distribution $P' \propto P^{\frac{N}{\sigma_N^2}}$, with larger σ_N^2, the distribution becomes flatter and efficiency uncertainty increases.



C = $\frac{N}{\sigma_N^2}$, as c becomes smaller (fitting uncertainty of hTotal increases), the distribution is flatter.

As seen in the formula, this error is not affected by fitting errors in hMatched, and the following simple code verified the case.

```
void matched_noerror(){
    TFile* f = TFile::Open("output/files/eff_ID_ECpt_datapp_HI_dr0.050_exp.root");
    //hMacthed, histogram of number of ID tracks with a matched MS track, uncertainty is set to be fitting uncertainty
    //hTotal, histogram of number of ID tracks in total, uncertainty is set to be fitting uncertainty
    //new_eff2, efficiency graph calculated vs pT, using,
    //TGraphAsymmErrors *new_eff2 = new TGraphAsymmErrors(hMatched, hTotal,"cl=0.683 b(1,1) mode");
    TGraphAsymmErrors* new_eff2 = (TGraphAsymmErrors*)f->Get("new_eff2");
    TH1F* hMatched = (TH1F*)f->Get("hMatched");
    TH1F* hTotal = (TH1F*)f->Get("hTotal");
    //define a new hMatched, and set its error bar to 0
    TH1F* hMatched_n = (TH1F*)hMatched->Clone();
    for (int i = 0; i < 6; i++){
        hMatched_n->SetBinError(i+1,0);
    }
    TGraphAsymmErrors* new_eff3 = new TGraphAsymmErrors(hMatched_n, hTotal,"cl=0.683 b(1,1) mode");
    cout << "With error bars on hMatched, the eff errors are:" << endl;
    cout << "Lower errors: " << new_eff2->GetErrorYlow(0) << ", " << new_eff2->GetErrorYlow(1) << ", " << new_eff2->GetErrorYlow(2) << "
        , " <<new_eff2->GetErrorYlow(3) << ", " <<new_eff2->GetErrorYlow(4) << ", " <<new_eff2->GetErrorYlow(5) <<  endl;
    cout << "Upper errors: " << new_eff2->GetErrorYhigh(0) << ", " << new_eff2->GetErrorYhigh(1) << ", " << new_eff2->GetErrorYhigh(2) <<
         ", " <<new_eff2->GetErrorYhigh(3) << ", " <<new_eff2->GetErrorYhigh(4) << ", " <<new_eff2->GetErrorYhigh(5) << endl;

    cout << "Without error bars on hMatched, the eff errors are:" << endl;
    cout << "Lower errors: " << new_eff3->GetErrorYlow(0) << ", " << new_eff3->GetErrorYlow(1) << ", " << new_eff3->GetErrorYlow(2) << "
        , " <<new_eff3->GetErrorYlow(3) << ", " <<new_eff3->GetErrorYlow(4) << ", " <<new_eff3->GetErrorYlow(5) << endl;
    cout << "Upper errors: " << new_eff3->GetErrorYhigh(0) << ", " << new_eff3->GetErrorYhigh(1) << ", " << new_eff3->GetErrorYhigh(2) <<
         ", " <<new_eff3->GetErrorYhigh(3) << ", " <<new_eff3->GetErrorYhigh(4) << ", " <<new_eff3->GetErrorYhigh(5) << endl;

}
```

```
With error bars on hMatched, the eff errors are:
Lower errors: 0.000859964, 0.00106934, 0.000827385, 0.000374413, 0.000840199, 0.000661177
Upper errors: 1.14341e-06, 0.000766622, 0.000653977, 2.91142e-08, 0.000714811, 0.000555899
Without error bars on hMatched, the eff errors are:
Lower errors: 0.000859964, 0.00106934, 0.000827385, 0.000374413, 0.000840199, 0.000661177
Upper errors: 1.14341e-06, 0.000766622, 0.000653977, 2.91142e-08, 0.000714811, 0.000555899
```

Summary

- The option "cl=0.683 b(1,1) mode”, which is a replacement for the old method BayesDivide in dividing two histograms for efficiency uses Bayesian method, the code was written by Marc Paterno, and the description of method is available here, https://home.fnal.gov/~paterno/images/effic.pdf .
- This method utilizes only uncertainty in fitting total events but not that of the matched events.
  - The fitting is a simultaneous fitting, and number of total events and its uncertainty is obtained by simultaneously fitting into matched and unmatched data.

Back up (reasons for using shortest confidence interval by the author)

I recommend using the *shortest 68.3% confidence interval* as the measure of the uncertainty in the efficiency measurement. It has two attractive features. First, it has a known probability content, one chosen to be the same as a "1 σ" Gaussian error. Therefore such error intervals will behave as we most often expect. Second, it is the most constrained region which has this probability content, so that we present our measurement in the fashion that most constrains the range in which we believe the true value exists.

References

- https://home.fnal.gov/~paterno/images/effic.pdf Marc Paterno
- https://root.cern/root/html524/TGraphAsymmErrors.html#TGraphAsymmErrors:BayesDivide
- https://root.cern.ch/doc/master/classTEfficiency.html, Section IV: Statistic Options