TEfficiency: A ROOT class for calculating efficiencies

Christian Gumpert

CERN PH–SFT, IKTP

28. October 2010

Outline

Efficiencies

Applications

- expressing trigger performance
- describing detector effects
- used in cut – based analysis

Example – Selection efficiency

We use cuts to select electrons. For a further analysis we need to know the selection efficiency as a function of p_T. The efficiency can be estimated by:

$$\varepsilon(p_T) = \frac{\#\text{ electrons with given } p_T \text{ which pass the cuts}}{\#\text{ electrons with given } p_T}$$

Efficiencies

Applications

- expressing trigger performance
- describing detector effects
- used in cut – based analysis

Example – Selection efficiency

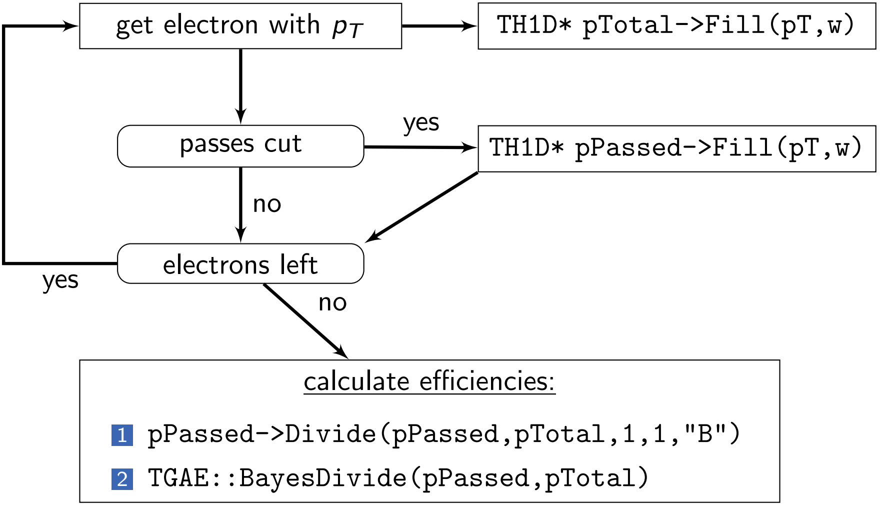
We use cuts to select electrons. For a further analysis we need to know the selection efficiency as a function of p_T. The efficiency can be estimated by:

$$\varepsilon(p_T) = \frac{\#\text{ electrons with given } p_T \text{ which pass the cuts}}{\#\text{ electrons with given } p_T}$$

But what is the uncertainty of ε?

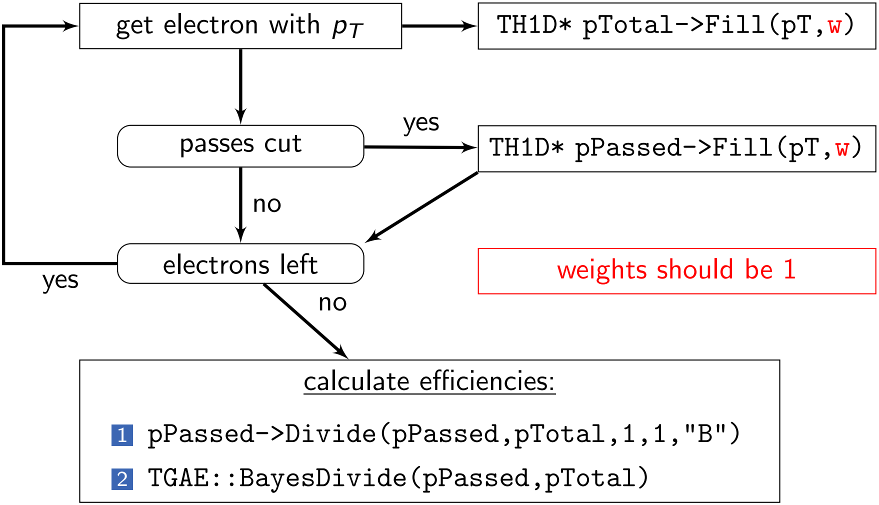
How is it done at the moment?

Example – Selection efficiency



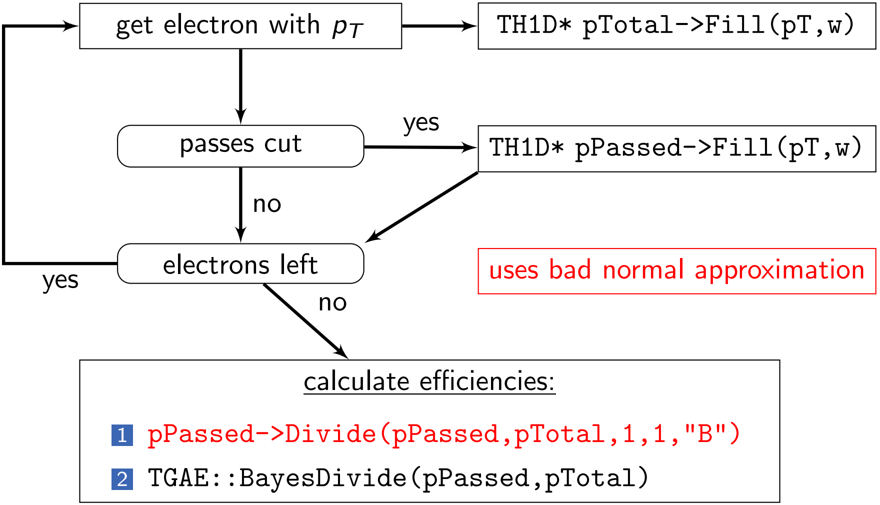
What are the problems?

Example – Selection efficiency



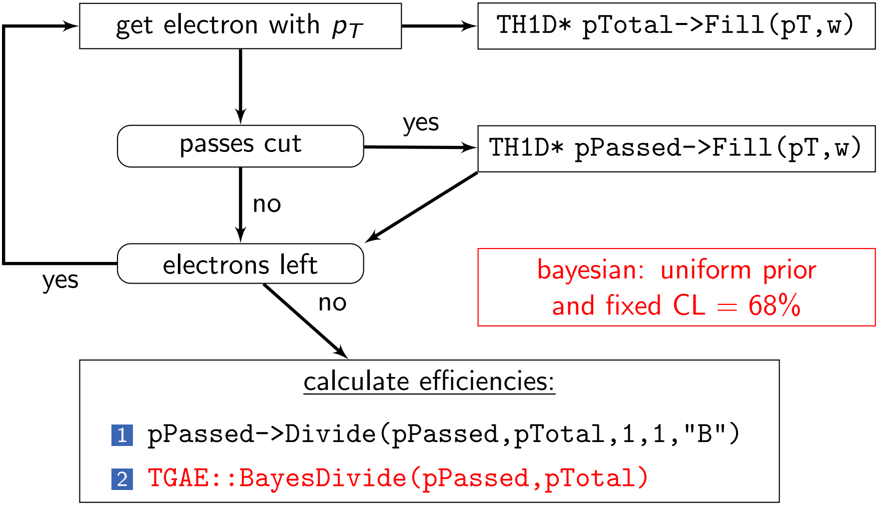
What are the problems?

Example – Selection efficiency



What are the problems?

Example – Selection efficiency



Drawbacks

- `TH1::Divide` uses normal approximation which fails for $\varepsilon \longrightarrow 1$ or 0
  $\longrightarrow$ confidence intervals have bad coverage
- only one (bayesian) method for a proper error calculation is supported
- no reasonable results for weighted histograms
- external fitting routine (`TBinomialEfficiencyFitter`)
- efficiencies as `TGraphAsymmErrors` contain less information
  $\longrightarrow$ no merging/combining of different efficiencies possible

Requirements on `TEfficiency`

- provide statistically correct error calculation for frequentist and bayesian approaches
- handle weights in a proper way

Efficiencies from a statistical point of view

Interlude

- efficiency ε = probability of a positive outcome of a *Bernoulli* trial
- binomial distribution = probability of finding k successes in a sequence of N independent *Bernoulli* trials, each with a success probability of ε

$$P(k; \varepsilon, N) = \binom{N}{k} \varepsilon^k (1-\varepsilon)^{N-k}$$

with the following properties:

$$\langle k \rangle = N\varepsilon$$
$$\sigma_k^2 = N\varepsilon(1-\varepsilon)$$

Estimate efficiencies (frequentist)

Estimation

Observing k successes out of N trials the efficiency can be estimated as:

$$\hat{\varepsilon} = \frac{k}{N} \quad (\hat{\varepsilon} \equiv 0 \text{ if } N = 0)$$

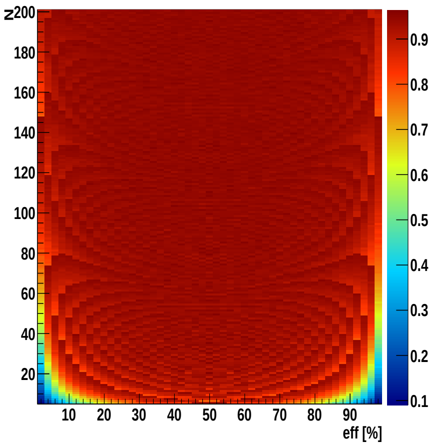
Confidence intervals for confidence level $1 - \alpha$

Notation: $\kappa = \Phi^{-1}(1 - \frac{\alpha}{2})$ ...quantile of normal distribution

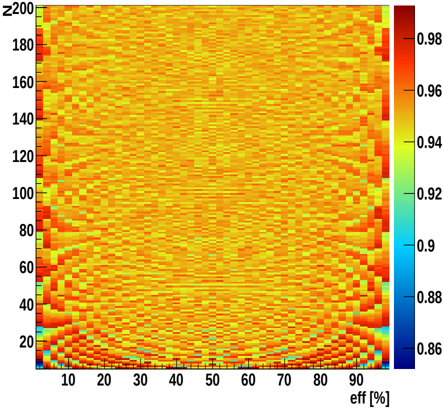
- normal approximation: $\varepsilon \in \hat{\varepsilon} \pm \kappa \sqrt{\frac{\hat{\varepsilon}(1-\hat{\varepsilon})}{N}}$
- *Wilson* interval: $\tilde{\varepsilon} = \frac{k + \frac{\kappa^2}{2}}{N + \kappa^2}, \quad \varepsilon \in \tilde{\varepsilon} \pm \frac{\kappa}{N+\kappa^2} \sqrt{\hat{\varepsilon}(1 - \hat{\varepsilon})N + \frac{\kappa^2}{4}}$
- *Agresti – Coull* interval: $\varepsilon \in \tilde{\varepsilon} \pm \kappa \sqrt{\frac{\tilde{\varepsilon}(1-\tilde{\varepsilon})}{N+\kappa^2}}$
- *Clopper – Pearson*: $P(X \geq k; \varepsilon, N) = \frac{\alpha}{2}$ and $P(X \leq k; \varepsilon, N) = \frac{\alpha}{2}$

Actual coverage of frequentist intervals (CL = 95%)

normal approximation

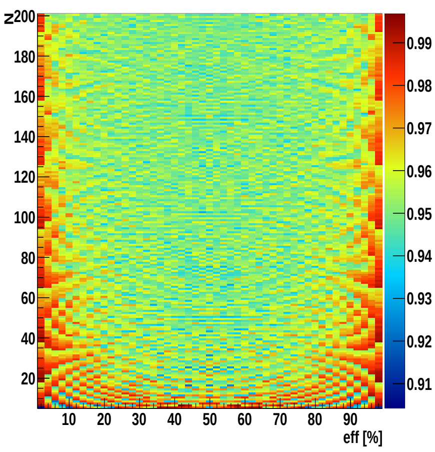


Wilson interval

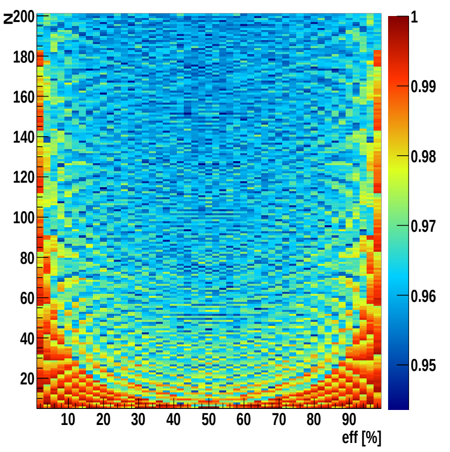


Actual coverage of frequentist intervals (CL = 95%)

Agresti-Coull interval



Clopper-Pearson interval



Estimate efficiencies (bayesian)

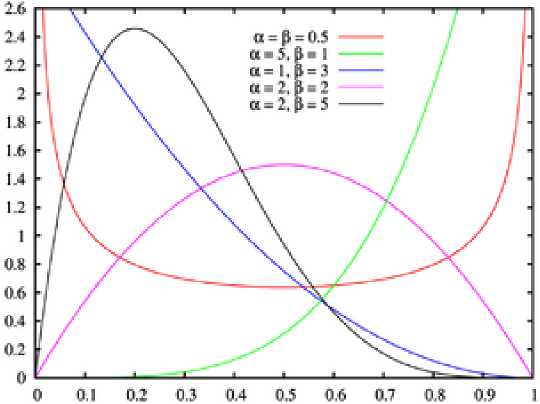
Estimation

- using likelihood function: $\mathcal{L}(\varepsilon; k, N) \propto \binom{N}{k} \varepsilon^k (1-\varepsilon)^{N-k} \cdot \mathrm{Prior}(\varepsilon)$
- supported prior probability:
  $\mathrm{Prior}(\varepsilon) = \mathrm{Beta}(\varepsilon; \alpha, \beta) \propto \varepsilon^{\alpha-1}(1-\varepsilon)^{\beta-1}$
- posterior probability: $P(\varepsilon; k, N) \propto \binom{N}{k} \varepsilon^{k+\alpha-1}(1-\varepsilon)^{N-k+\beta-1}$

$\Rightarrow$ expectation value: $\hat{\varepsilon} = \frac{k+\alpha}{N+\alpha+\beta}$ $\quad (\hat{\varepsilon} \equiv 0$ if $N+\alpha+\beta = 0)$

Estimate efficiencies (bayesian)

Beta distributions



Estimate efficiencies (bayesian)

Estimation

- using likelihood function: $\mathcal{L}(\varepsilon; k, N) \propto \binom{N}{k} \varepsilon^k (1-\varepsilon)^{N-k} \cdot \mathrm{Prior}(\varepsilon)$
- supported prior probability:
  $\mathrm{Prior}(\varepsilon) = \mathrm{Beta}(\varepsilon; \alpha, \beta) \propto \varepsilon^{\alpha-1}(1-\varepsilon)^{\beta-1}$
- posterior probability: $P(\varepsilon; k, N) \propto \binom{N}{k} \varepsilon^{k+\alpha-1}(1-\varepsilon)^{N-k+\beta-1}$

$\Rightarrow$ expectation value: $\hat{\varepsilon} = \frac{k+\alpha}{N+\alpha+\beta}$ $\quad (\hat{\varepsilon} \equiv 0 \text{ if } N+\alpha+\beta = 0)$

Confidence intervals for confidence level $1-\alpha$

- cumulative distribution (regularized incomplete beta function):

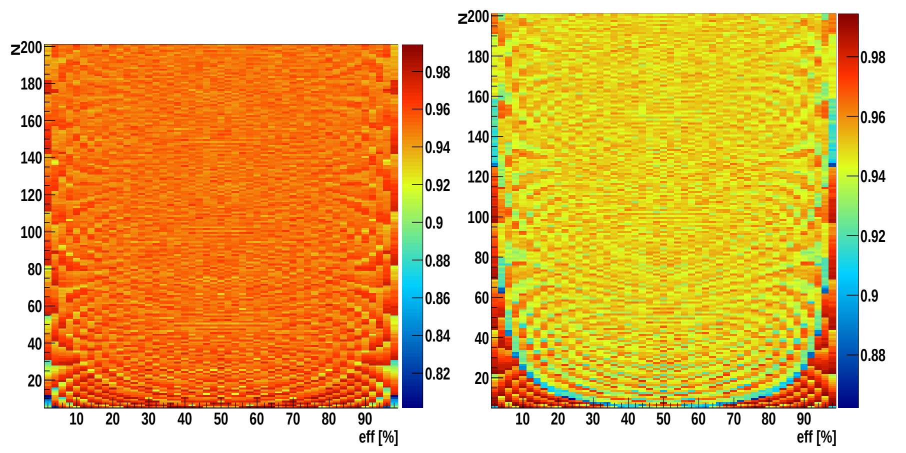
$$F(\varepsilon; k, N, \alpha, \beta) = \frac{1}{B(k+\alpha, N-k+\beta)} \int_0^{\varepsilon} t^{k+\alpha-1}(1-t)^{N-k+\beta-1} dt$$

- confidence interval: $F^{-1}(\frac{\alpha}{2}) \leq \varepsilon \leq F^{-1}(1-\frac{\alpha}{2})$

Actual coverage of bayesian intervals (CL = 95%)

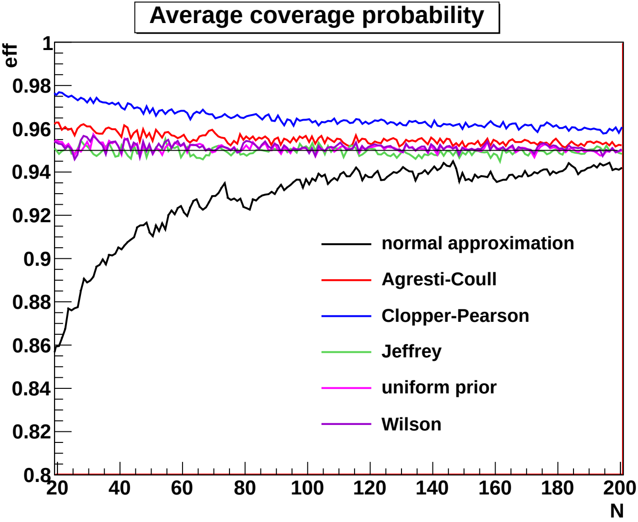
uniform prior

Jeffrey prior



Average coverage probability of confidence intervals

Nominal confidence level 95%



Concept of TEfficiency

Type declarations

```
class TEfficiency : public TObject {

 public:
   enum EStatOpt {soFCP,...};    //statistic option

 ...

};
```

- enumeration for all supported statistic options
  soF* . . . frequentist ones
  soB* . . . bayesian ones
- implementation realised by function pointer

Concept of TEfficiency

Data members

```
class THEfficiency :  public TObject {
 private:
   TH1* fTotalHistogram;    //containing all events
   TH1* fPassedHistogram;   //containing only passed events
   Double_t fConfLevel;     //0 < confidence level < 1
   EStatOpt fStatisticOption;
   Double_t fBeta_alpha;    //shape parameter prior > 0
   Double_t fBeta_beta;     //shape parameter prior > 0
   Double_t (*fBoundary);   //return confidence limits
   Double_t fWeight;        //global weight > 0

};
```

Concept of THEfficiency

Public methods

- Get-/Set-methods for histograms, confidence level, prior parameters, weight and statistic option
- `void Fill(Bool_t bPassedCut,Double_t x,Double_t y=0,Double_t z=0)`
- `Double_t GetEfficiency(Int_t bin) const`
- `Double_t GetEfficiencyErrorUp/Low(Int_t bin) const`
- `void Draw(Option_t*)` using `TGraphAsymmErrors` or `TH2` class
- `Int_t Fit(TF1*,Option_t*)` using internally the `TBinomialEfficiencyFitter` class (maximum likelihood fit)
- methods for merging (`Add`,`Merge`,`+=`,`+`) and combining (`Combine`)
- complete documentation: `http://root.cern.ch/root/html/TEfficiency.html`

What does a weight represent?

a weight is usually given by

$$w = \frac{\sigma_i L}{N_{\text{gen}} \epsilon_{\text{trig}}}$$

with

- σ_i …cross-section for a given process i
- L …integrated luminosity
- N_{gen} …generated Monte – Carlo events, sample size
- ϵ_{trig} …(known) trigger efficiency

$\Rightarrow$ weights represent different processes, sample sizes, luminosities, trigger efficiencies or any combination of them

What does a weight represent?

a weight is usually given by

$$w = \frac{\sigma_i L}{N_{\text{gen}} \epsilon_{\text{trig}}}$$

with

- σ_i . . . cross-section for a given process i
- L . . . integrated luminosity
- N_{gen} . . . generated Monte – Carlo events, sample size
- ϵ_{trig} . . . (known) trigger efficiency

$\Rightarrow$ weights represent different processes, sample sizes, luminosities, trigger efficiencies or any combination of them

$\Rightarrow$ treat samples with different weights as distinct subgroups
$\longrightarrow$ one `TEfficiency` object for each weight!

Merging efficiencies

Same process in different samples

for determining the selection efficiency of my cut I processed:

- yesterday: a sample with N_1 events $\rightarrow \varepsilon_1 = \frac{k_1}{N_1}$
- today: another sample with N_2 events of the same process $\rightarrow \varepsilon_2 = \frac{k_2}{N_2}$
- suppose same integrated luminosity and trigger efficiency

$\Rightarrow$ $w_1 = \frac{\sigma L}{N_1 \epsilon}$ and $w_2 = \frac{\sigma L}{N_2 \epsilon}$

!!! but different weights are totally artificial

$\Rightarrow$ should obtain same total selection efficiency as using the merged sample: $\varepsilon = \frac{k_1 + k_2}{N_1 + N_2}$ with new weight $w = \frac{\sigma L}{(N_1 + N_2)\epsilon}$

$\Rightarrow$ use: `Add`, `Merge` method or `+`, `+=` operators

Combining efficiencies (the way I implemented it)

Different processes

electrons can originate from two different processes

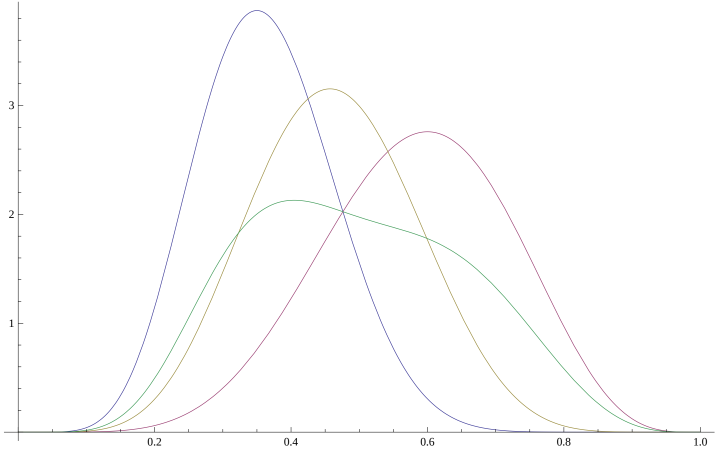
- process 1: efficiency ε_1 and weight $w_1 = \frac{\sigma_1 L}{N_1 \epsilon}$
  $\rightarrow$ posterior probability: $P_1(\varepsilon_1; k_1, N_1)$
- process 2: efficiency ε_2 and weight $w_2 = \frac{\sigma_2 L}{N_2 \epsilon}$
  $\rightarrow$ posterior probability: $P_2(\varepsilon_2; k_2, N_2)$

$\Rightarrow$ overall posterior is the weighted average of the individual posteriors: $P_{\text{total}} = \sum_i p_i \cdot P(\varepsilon_i; k_i, N_i)$

- probability p_i is the probability that an electron originates from process i: $p_i = \frac{\sigma_i}{\sum_k \sigma_k} = \frac{w_i N_i}{\sum_k w_k N_k}$

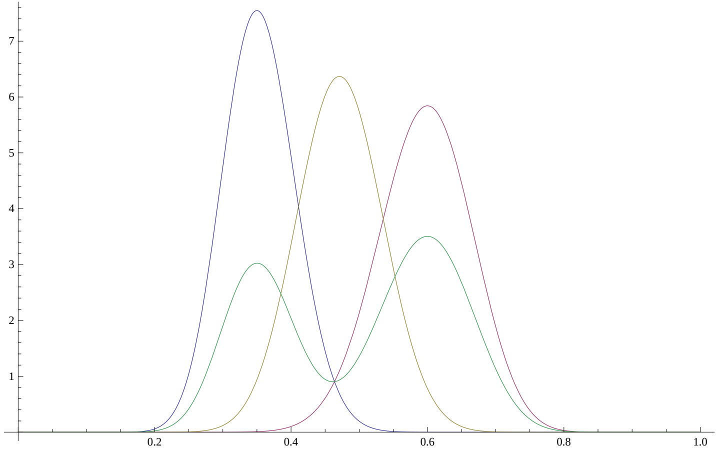
$\Rightarrow$ use `Combine` method

Example for combining two posteriors



red: $P_1(\varepsilon; 7, 20)$; blue: $P_2(\varepsilon; 6, 10)$; green: combined posterior for $p_1 = 0.4, p_2 = 0.6$; yellow: $P_{old}(\varepsilon; 6.4, 14)$

Example for combining two posteriors



red: $P_1(\varepsilon; 28, 80)$; blue: $P_2(\varepsilon; 30, 50)$; green: combined posterior for $p_1 = 0.4, p_2 = 0.6$; yellow: $P_{old}(\varepsilon; 29.2, 62)$

Combining how it is implemented right now

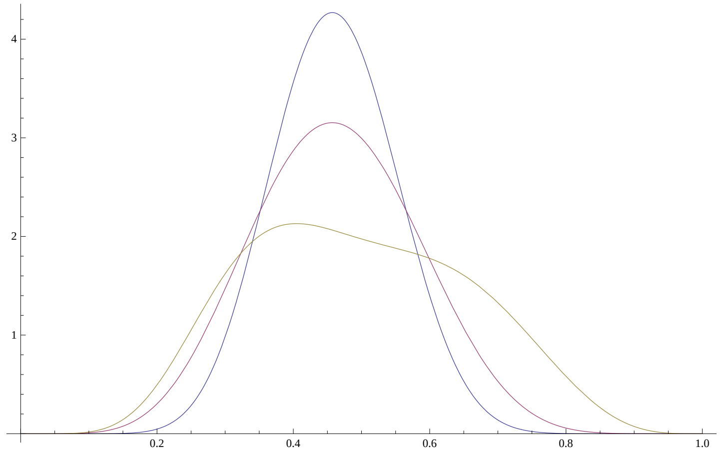
Using *effective entries*

The posterior distribution for calculating the confidence interval is constructed in the following way:

1. $\tilde{k} = \sum_i w_i k_i \quad \text{and} \quad \tilde{N} = \sum_i w_i N_i$
2. $c = \frac{\sum_i w_i}{\sum_i w_i^2}$ (so called *effective entries*)

$\Rightarrow$ posterior: $P(\varepsilon, c\tilde{k}, c\tilde{N})$ is used to determine the quantiles

resulting combined posterior



red: $P_{old}(\varepsilon; 6.4, 14)$, blue: $P_{new\ combined}$, yellow: $P_{my\ combined}$

Advantages of `TEfficiency`

- provides several methods for calculating statistically correct confidence intervals for frequentist and bayesian statistics
- encapsulates all information needed for calculating efficiencies
  $\longrightarrow$ possible to combine and merge different `TEfficiency` objects
- global weight $\longrightarrow$ reusing of the same object for different weights possible (e.g. normalising to different luminosities)
- does the bookkeeping of histograms
- clean handling of the fit routine (e.g. attaching the fitted function)

Outlook

Possible future developments

- include in `TTree::Draw` method to generate automatically efficiency graphs
- solve problem of combining efficiencies (hopefully even for frequentist approaches)
- simplify `Draw` method (needs asymmetric errors in histograms which is going to be introduced soon)
- whatever you request or suggest