Software to set upper limits using a Bayesian technique.

(Poisson process, nuisance parameters on epsilon/background)

- Aim was to make a tool to quickly explore properties of priors.
- 2 month undergraduate project in 2005 supervised by Louis Lyons, recently re-written in Root (C++).
- Validated by repeating results of CDF "Interval estimation in the presence of nuisance parameters" (Heinrich et al.) and other sources.

Index of slides

- Mathematical statement of problem
- Software features and settings
- Supported priors
- Method for solving problem
- Performance/precision
- Example plots (3)
- Conclusion

Mathematically, need to solve:

$$\int_0^{s_u}\left[\int_0^{\infty}\int_0^{\infty}\left\{\frac{e^{-(\varepsilon s+b)}(\varepsilon s+b)^n}{n!}\right\}\pi_2\pi_3\,d\varepsilon\,db\right]\pi_1\,ds=c$$

where	$\pi_1=\pi_1(s)$	prior for signal, s
	$\pi_2=\pi_2(\varepsilon)$	prior for epsilon, ε
	$\pi_3=\pi_3(b)$	prior for background, b
	s_u	upper limit on signal, s

Solve for s_u with choice of priors and credibility level, c.

Features of software:

- Generate upper limits for a given c (credibility), for a range of n.
- Plot posterior PDF's (as a graph in Root)
- Produce coverage plots, including case of nuisance on epsilon (using technique in CDF note.)

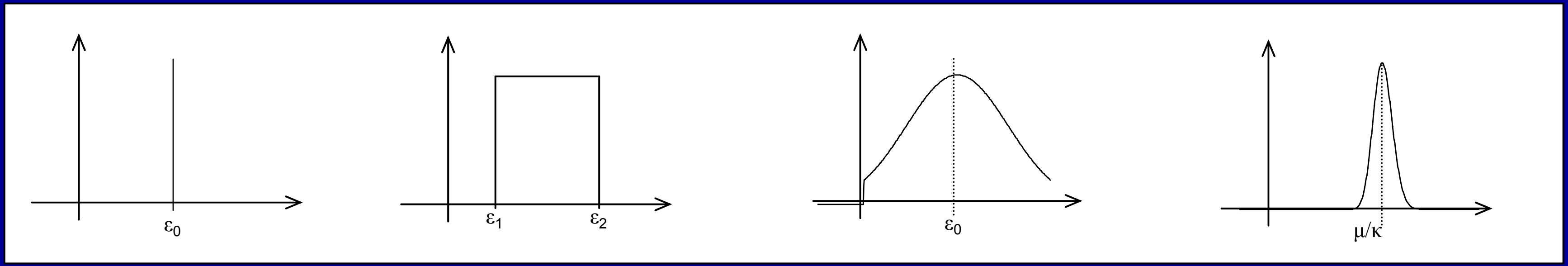
Input settings:

1) Turn on the features from above list.
2) Set n-range for each feature.
3) Set credibility level (c).
4) Choose priors on s, ε and b.

(there are many other options related to internal working that end user probably won't want to touch)

Current priors supported

For background and epsilon:



delta (i.e. no prior) | "box" | normal (choice of how to truncate) | gamma

For signal: "box" prior (implementation of $s^{\alpha-1}$ so far problematic)

Any prior separable in s, ε and b is straightforward to add.

(Coverage plots only for delta, normal and gamma, so far.)

Re-write problem, but don't worry about constants:

$$\int_0^{s_u} p(s \mid n) \cdot ds = c \cdot \int_0^{\infty} p(s \mid n) \cdot ds$$

And apply Newton-Raphson method:

$$s_u^{\text{improved}} = s_u - \frac{\int_0^{s_u} p(s \mid n) \cdot ds - c \cdot \int_0^{\infty} p(s \mid n) \cdot ds}{p(s \mid n)}$$

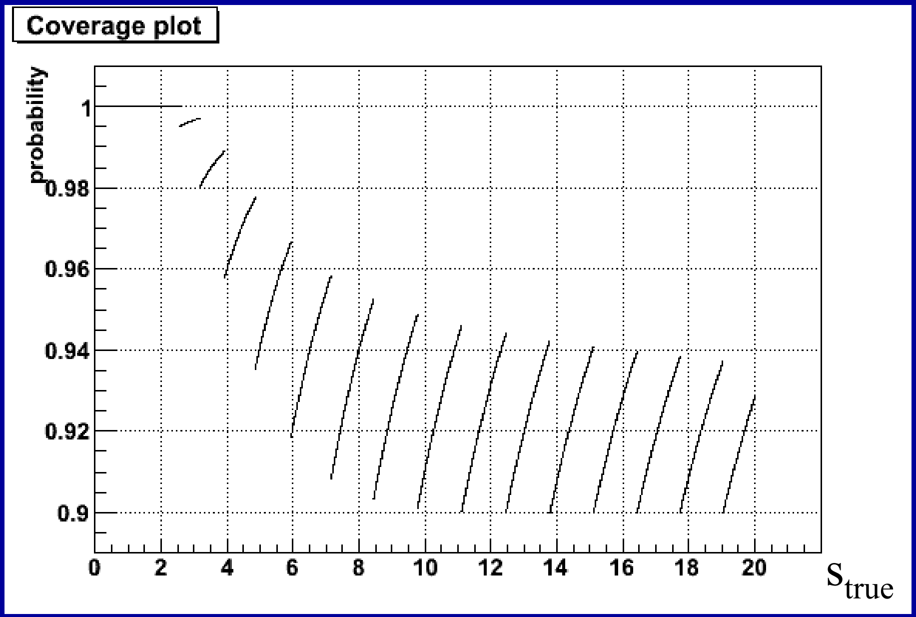
- Constants drop out of integrals (and therefore out of priors).
- $p(s|n)$ is still an integral over ε and b.
- All integrals then carried out using Gaussian integration. (i.e. completely numerical)
- Convergence in ~4 iterations.

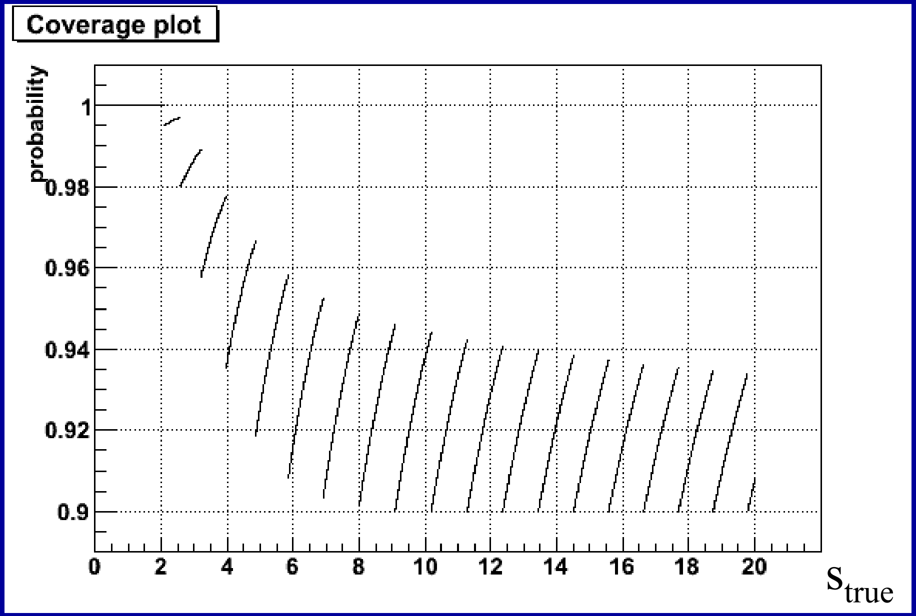
- Precision beyond 4d.p. when compared to analytic.
- No speed optimisation … yet (anticipate factor 4 reduction by compiling and tuning settings for precision, numerical integration etc.)

Total times to compute upper limits for n = 0 to 20:	
No nuisance parameters on ε or b	< 1 second
Nuisance parameter on ε (±10%)	~ 2 seconds
Nuisance on both ε and b (both ±10%)	~ 18 seconds
Total times to compute coverage (10,000 points):	
$\varepsilon = 1.0 \pm 0.1$, $b = 3.0$	~12 minutes
$\varepsilon = 1.0 \pm 0.3$, $b = 3.0$	~19 minutes
(for additional ε_{true}, computation is much faster)	

All times are running on a single ~3GHz CPU on the TWIST experiment's cluster (at Triumf).

Example Plots (1)



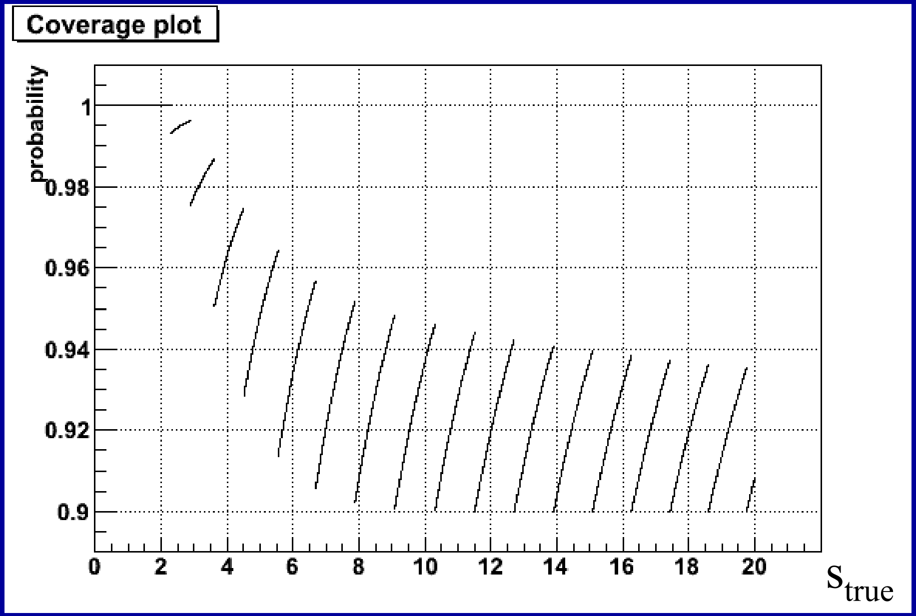


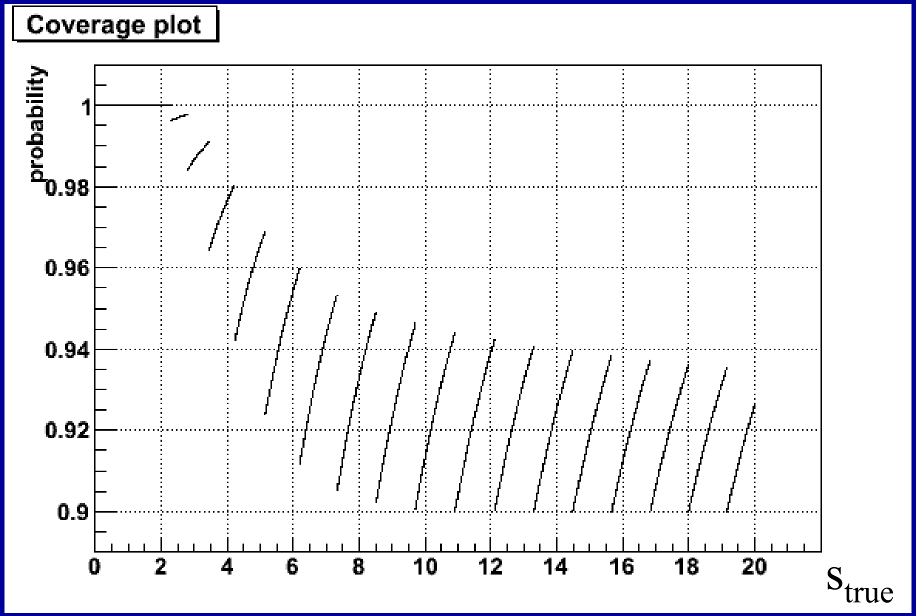
$\varepsilon = 0.9$, $b = 3.0$

$\varepsilon = 1.1$, $b = 3.0$

For both plots $c = 0.90$, uniform signal prior.

Example Plots (2)

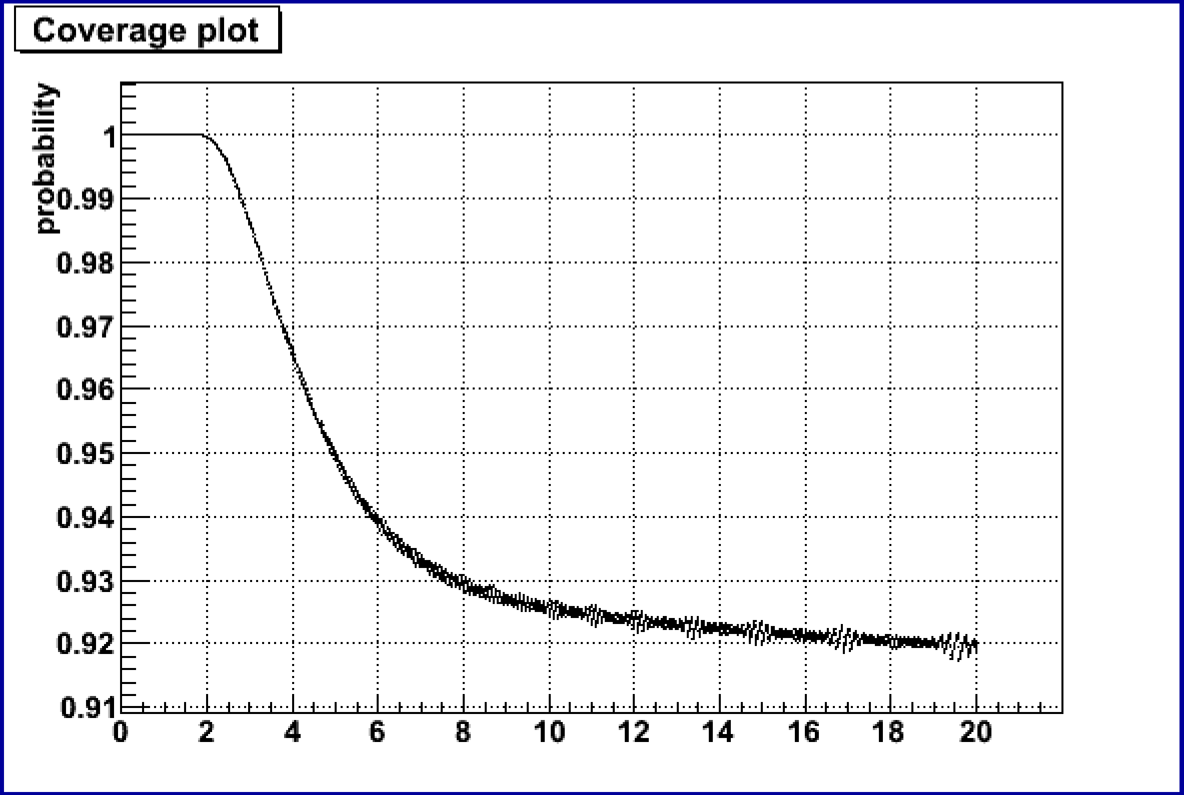




$\varepsilon = 0.9$, $b = 2.7$

$\varepsilon = 1.1$, $b = 3.3$

For both plots $c = 0.90$, uniform signal prior.



$\varepsilon = 1.0 \pm 0.1$ (Gamma), $b = 3.0$

$\varepsilon_{true} = 1.0$, $c = 0.90$

- Easy for user to change settings.
- Flexibility in priors while maintaining speed/precision.
- Inclusion of nuisance parameters on both ε and b.
- Can quickly explore properties like varying mean/s.d. of background etc.