Nuisance Parameters and Other Issues in Searching for Signals in High-Energy Physics Experiments

Radford M. Neal, University of Toronto

28 June 2007

Outline of the Talk

I'll look at two versions of a problem that I hope resemble ones of interest to people here:

1) A signal detection problem without nuisance parameters
   - Role of the likelihood function
   - Need for reducing dimensionality
   - Building classifiers as a way to reduce dimensionality
   - Robustness

2) Introducing nuisance parameters for uncertain physics and detector properties
   - The revised likelihood function
   - Is there any feasible way to compute this likelihood?
   - How should we reduce dimensionality now?
   - How can we do the computations needed?

My discussion of solutions will have much in common with present practice, but I hope to clarify the problems, for both physicists and statisticians.

Signal Detection With No Nuisance Parameters

We will observe O events, indexed by $i = 1, \ldots, O$, that are described by PID variables, v_i.

Events can either be from the "background" or (if it exists) from the "signal".

Simulation programs for background and signal events exist, which stochastically generate the PID variables from either a background distribution, which has probability density function $p_0(v)$, or a signal distribution, which has probability density function $p_1(v)$.

Difficulty: No explicit formulas for $p_0(v)$ and $p_1(v)$ exist.

The real events come from a *mixture* of signal and background distributions, with proportion f of signal. We may be most interested in whether or not f is zero.

The Likelihood Function

The *likelihood function* is the probability of the observed data, seen as a function of the model parameter(s).

The likelihood function for this problem is

$$L(f) \;=\; \prod_{i=1}^{O} \Big[f p_1(v_i) \;+\; (1-f) p_0(v_i) \Big]$$

The likelihood function is defined only up to an arbitrary constant factor — ie, only ratios of likelihood for different values of f are meaningful.

The Role of the Likelihood

According to the *likelihood principle* — widely, though not universally, accepted by statisticians — the likelihood function contains all the information from the experiment that is relevant to inference for the parameters(s). So inference (except checks of the appropriateness of the model that the likelihood is based on) should depend only on the likelihood function.

Frequentist confidence intervals and p-values (other than those for model checking) usually violate this principal.

Example: In the (in)famous Poisson signal+background problem, any method that produces different inferences from an observed count of zero depending on the expected background violates the likelihood principle, and in my view cannot be justified.

Aside: Even if you're not willing to abandon frequentist ideas of confidence, at least abandon confidence *intervals*, in favour of upper and lower confidence *limits*, constructed separately. Letting your upper limit be liberal because your lower limit is conservative makes no sense, even if the resulting interval has "correct" coverage.

Unfortunately, We Can't Compute the Likelihood

Back to our model, with likelihood $L(f) \;=\; \prod_{i=1}^{O} \Big[f p_1(v_i) \;+\; (1-f) p_0(v_i) \Big]$.

This simple mixture model, with only the mixing proportion unknown, would usually not be difficult (eg, easy to find the maximum likelihood estimate for f).

But here, we don't know how to compute the likelihood! It involves p_0 and p_1, which are known only through simulation programs.

If the v_i are low-dimensional, we can generate many points from p_0 and p_1, and use them to get good estimates for these density functions.

But if the dimensionality if greater than ≈ 4, this may not work well.

Fortunately, there's a fairly good solution...

We Only Really Need to Compute $p_1(v)/p_0(v)$

Since constant factors in the likelihood can be ignored, we can reduce the likelihood as follows:

$$
\begin{aligned}
L(f) &= \prod_{i=1}^{O} \Big[f p_1(v_i) \;+\; (1-f) p_0(v_i) \Big] \\
&= \prod_{i=1}^{O} p_0(v_i) \Big[f \, \frac{p_1(v_i)}{p_0(v_i)} \;+\; (1-f) \Big] \\
&= \Big[\prod_{i=1}^{O} p_0(v_i) \Big] \cdot \prod_{i=1}^{O} \Big[f \, \frac{p_1(v_i)}{p_0(v_i)} \;+\; (1-f) \Big] \\
&\propto \prod_{i=1}^{O} \Big[f \, \frac{p_1(v_i)}{p_0(v_i)} \;+\; (1-f) \Big]
\end{aligned}
$$

So it's enough to be able to compute $p_1(v)/p_0(v)$, without having to compute $p_0(v)$ and $p_1(v)$.

Training a Classifier to Distinguish Signal and Background

Classifiers based on neural networks, decisions trees, or whatever can be used to solve this problem.

Suppose we simulate many events from the background and signal distributions — say equal numbers, for simplicity. With enough events, a good classification method should be able to output something very close to the actual probability that a simulated event is signal, based on the PID variables.

This probability is

$$P(\text{signal}\,|\,v) \;=\; \frac{p_1(v)}{p_0(v)+p_1(v)}$$

So once we have this classifier, we can find the desired ratios as follows:

$$\frac{p_1(v)}{p_0(v)} \;=\; \Big[P(\text{signal}\,|\,v)^{-1} - 1\Big]^{-1}$$

If we really trust our classifier, we can now compute the likelihood function for f, and present a plot of $L(f)$ as the result of the experiment. Note that a plot of $L(f)$ has more information than *any* sort of interval. Only $L(f)$ is an adequate summary of the experimental results.

Robustness to Flaws in the Classifier

If we don't totally trust our classifier, we can still use it to get good results. We just treat it as a way of reducing the dimensionality of the data — from the multidimensional v_i to the scalar $r_i \approx p_1(v_i)/p_0(v_i)$ produced using the classifier.

If the classifier were perfect, this reduction does not lose any useful information. If it's not perfect, it will throw away a bit of information, but the reduction to a scalar allows us to easily estimate $p_0(r)$ and $p_1(r)$ from simulation data, and use them to compute the likelihood given the r_i. The results will be valid, if not quite as precise as with a perfect classifier.

One could reduce the data further by binning the r_i values, but this loses information. (But using a fairly large number of bins may be OK, if it loses little information, and makes estimating the probabilities easier.)

It's not so easy to get robustness to flaws in the simulators for background and signal events. That brings us to nuisance parameters...

Handling Uncertainty in the Physics and Detector Behaviour

In practice, we don't know p_0 and p_1 exactly. The simulators for generating from these distributions have some parameters — relating either to the physics or to the behaviour of the detector — whose values are not known precisely. Call these parameters ϕ.

(We can assume ϕ is the same for simulating p_0 and p_1, though some components of ϕ may be used by only one of these simulators.)

We have to assume that these ϕ parameters are known to some degree, or there's no hope. I'll assume that based on theory or previous experiments, a prior distribution for ϕ is available, with density $p(\phi)$. This prior might well be flawed however — eg, it might assume independence of components of ϕ when it really ought not to.

Note that ϕ is a "nuisance" parameter, since our only real interest is in f. The fact that ϕ is unknown is an annoyance. (Well, maybe someone is interested in ϕ itself, but I'll assume not here.)

The Likelihood and Marginal Likelihood With Nuisance Parameters

Here's our likelihood function once there are ϕ parameters:

$$L(f,\phi) \;\;=\;\; \prod_{i=1}^{O} \Big[f p_1(v_i|\phi) \;+\; (1-f) p_0(v|\phi) \Big]$$

where $p_0(v|\phi)$ and $p_1(v|\phi)$ denote probability densities for generating v from the background and signal simulators with parameters set to ϕ.

This is a high dimensional function (since ϕ is typically high dimensional), and hence will be difficult to visualize. Just plotting $L(f,\phi)$ will *not* be a feasible way of presenting the results of the experiment.

We can integrate $L(f,\phi)$ with respect to the prior for ϕ, however, to obtain a *marginal likelihood function* for f alone, which we would be able to plot (if we could compute it):

$$\underline{L}(f) \;\;=\;\; \int L(f,\phi)\, p(\phi)\, d\phi$$

We could compute this fairly easy by simple Monte Carlo (sampling from the prior for ϕ), if we could compute $L(f,\phi)$...

Can We Still Reduce Dimensionality Using a Classifier?

Here's what happens if we rewrite the likelihood as we did before:

$$\begin{aligned} L(f,\phi) &= \prod_{i=1}^{O} \Big[f p_1(v_i|\phi) \;+\; (1-f) p_0(v_i|\phi) \Big] \\ &= \Big[\prod_{i=1}^{O} p_0(v_i|\phi) \Big] \cdot \prod_{i=1}^{O} \Big[f \,\frac{p_1(v_i|\phi)}{p_0(v_i|\phi)} \;+\; (1-f) \Big] \end{aligned}$$

Unlike before, we can't ignore the first factor, since it now depends on the parameter ϕ. Properties of events that are irrelevant for classifying them as signal versus background may still be relevant for inferring ϕ, and hence indirectly f.

Two options:

- Figure out how to compute $p_0(v|\phi)$ and $p_1(v|\phi)$. It's probably possible, using methods similar to free energy computations in statistical physics, but the time required per event is likely prohibitive, given the large number of events.

- Reduce dimensionality some way or other, even if we lose some information. Perhaps the inferences can still be valid, even if they're not as precise as they could be.

Reducing Dimensionality With Nuisance Parameters

Option A: Train a classifier using background and signal events generated using a single value for the nuisance parameters (eg, the prior mean). Reduce the data from v_i to $P(\text{signal}|v_i)$, as approximated by this classifier.

Option B: Generate events with many values of ϕ, drawn from the prior (a different ϕ_i for every v_i). Train a classifier with v_i as inputs on all of this data, and use it to reduce the data as above.

Option C: Generate events using a fairly small number, H, of values for ϕ (perhaps carefully chosen to be representative of the whole prior). Use this data to train a classifier that takes both v and ϕ as inputs — ie, one that learns to classify events given both the PID variables and the nuisance parameters. Reduce the data from v_i to the H-dimensional vector with components $P(\text{signal}|v_i, \phi_k)$, for $k = 1, \ldots, H$. One could also try reducing this data further (eg, using PCA).

As before, it might be convenient to bin the reduced data, but the number of bins should be large enough that not much information is lost by doing this. The final reduced data derived from v_i will be denoted r_i (either a fairly low dimensional quantity, or the index of a bin).

The Reduced Data Likelihood

We can define a likelihood function based on the reduced data:

$$L_r(f,\phi) \;=\; \prod_{i=1}^{O} \Big[f p_1(r_i|\phi) \;+\; (1-f) p_0(r_i|\phi) \Big]$$

Since we have likely lost information by going from v_i to r_i, this is not the same function as $L(f,\phi)$. But it can be used to make valid (if inefficient) inferences.

We can again define a marginal likelihood, integrating over the prior for ϕ:

$$\underline{L}_r(f) \;=\; \int L_r(f,\phi)\, p(\phi)\, d\phi$$

If we can compute this, plotting it will display the results of the experiment, as well as possible given our computational limits.

Computing the Marginal Reduced Likelihood

We need to compute the marginal reduced likelihood,

$$
\begin{aligned}
\underline{L}_r(f) &= \int L_r(f,\phi)\,p(\phi)\,d\phi \\
&= \int \prod_{i=1}^{O} \Big[f p_1(r_i|\phi) \;+\; (1-f)p_0(r_i|\phi) \Big]\, p(\phi)\,d\phi
\end{aligned}
$$

If the r_i are fairly low dimensional, this seems possible.

We chose some moderate number of ϕ values, $\phi_1, \ldots, \phi_K$, from the prior, randomly or by some quasi-Monte Carlo scheme. We then average $L_r(f, \phi_k)$ over these K values to approximate the integral above.

Computing $L_r(f, \phi_k)$ will require simulating many events from the background and signal distributions with parameters ϕ_k, and then using these to estimate the probability densities $p_0(r|\phi_k)$ and $p_1(r|\phi_k)$ (or the bin probabilities, if the r_i were binned).

Concluding Questions and Remarks

- How does this relate to Poisson models of signal + background?

  If we reduce all the way to r_i being binary (and f is small), then $L(f, \phi)$ becomes a Poisson likelihood, and uncertainty in ϕ shows up as uncertainty in the Poisson mean.

- How do we account for Monte Carlo error in estimating $p_0(r|\phi)$ and $p_1(r|\phi)$ from simulation runs?

- How do we decide on the number of different nuisance parameter values for reducing dimensionality (H) and computing the marginal likelihood (K)?

- Aren't we ignoring relevant information not required by classifiers for any ϕ?

  Remember, we're very limited in the dimensionality of r, if we're to get good estimates for $p_0(r|\phi)$ and $p_1(r|\phi)$. But we could try to directly reduce the dimensionality of v (eg, with PCA), and add a small number of variables found that way to r. Perhaps better would be to build a model for $p(\phi|v)$ and then do PCA on the predictions of this model.

- Can we be robust to flaws in the prior for ϕ?