Handling Systematic Errors

Radford M. Neal, University of Toronto

BIRS, 19 July 2006

The Problem

Events, i, described by PID variables, v_i, can be either "background" or "signal".

Simulation programs for background and signal events exist, with unknown "systematic" parameters, ϕ (assumed, w.l.g., to be the same for both programs). These simulation programs define distributions for signal and background that depend on ϕ.

A prior for ϕ exists, which is based on some real information, but which is probably too simple — eg, Gaussian, perhaps with diagonal covariance matrix.

The real events come from a mixture of signal and background, with proportion f of signal. We may be most interested in testing whether f is zero. We want the inference to be both sensitive and valid for any likely value of ϕ.

Three-Phase Solution

Phase I: Simulate both signal and background events, and use them to train an event classifier to distinguish signal from background. Given v_i for an event, the classifier outputs the probability, p_i, that the event is from the signal distribution.

Phase II: Fix one or more cut points for p_i, used to put each event into one of two or more bins. With one cutpoint, these would be a background bin (p_i low) and a signal bin. We could divide up the signal bin using more cutpoints.

Estimate the probability that an event that is actually background will be put into the signal bin (or such probabilities for several signal bins), using a new set of events produced by the background simulator.

Multiplying by the number of background events that will be collected gives an estimate of the expected number, b_j, of background events that will land in bin j.

Phase III: Gather real events, use the event classifier to put them in bins, and see whether the the number of events in the signal bin(s) exceeds the expected background by enough to be declared significant.

Handling Systematic Errors — Phase I

In Phase I, what values for the systematic parameters, ϕ, should we use when generating signal and background events used to train the event classifier?

Option A: Generate all events using a single value, ϕ_*, for the systematic parameters (eg, the prior mean). Train a classifier with v_i as inputs on this data.

Option B: Generate events with many values of ϕ, drawn from the prior. Probably every event, i, would be generated with a different ϕ_i. Train a classifier with v_i as inputs on all of this data. It learns to "robustly" classify an event regardless of the value of ϕ_i used to generate it.

Option C: Same as Option B, except that the classifier takes both v_i and ϕ_i as inputs — ie, it learns to classify events given both the PID variables and the systematic parameters.

Option C looks interesting, but doesn't produce a single classifier for v_i. Let's look only at A and B for the moment. The choice between A and B, and the choice of classifier, affects sensitivity, but not the validity of the final conclusion.

Handling Systematic Errors — Phase II

We fix an event classifier obtained in Phase I, and fix a set of cutpoints (which define bins). We need to estimate the expected number of background events that will be put in each bin, using a new sample of simulated background events.

Option A: Generate N background events using a single set of systematic parameters, ϕ_*. Produces one set of estimates, $\hat{b}_j$, with known Poisson uncertainty, but with no indication of how much the b_j vary as ϕ varies.

Option B: For each of K values, ϕ_k, for the systematic parameters, generate M background events. To keep computation time the same as Option A, $KM = N$. Estimate expected numbers of background events for each ϕ_k, with the estimate for bin j being $\hat{b}_{j,k}$. Deconvolve the Poisson variation to estimate the distribution of the true b_j as ϕ varies. Needs M to be large enough for reasonable estimates $\hat{b}_{j,k}$, and hence K will be fairly small.

Option C: Let $M = 1$ and hence $K = N$ in Option B. Estimating $b_{j,k}$ for each ϕ_k separately is impossible. Instead, train a classifier that takes ϕ_k as input and tries to predict which bin event k is put in. When this classifier is given input ϕ, it produces bin probabilities that when multiplied by the total number of events give estimates of b_j for that ϕ. Using the prior on ϕ, we get a distribution for the b_j.

Discussion of Phase II Options

Option A ignores systematic error — the final inference may not be valid.

Option B accounts for systematic error using a fairly small number of values for ϕ. If the prior for ϕ is Gaussian, **and** the effects of ϕ on bin probabilities are linear, the resulting distribution for b_j will be Gaussian, and will be estimated reasonably well with a small sample. It's important to deconvolve, not use the distribution of $\hat{b}_j$ as if it were the distribution of b_j — simplest version: estimate the mean of b_j by the mean of the $\hat{b}_{j,k}$, and the variance of b_j by the variance of the $\hat{b}_{j,k}$ **minus** $\hat{b}_j/K$ (from Poisson variance).

But if there are non-linear effects, the distribution of b_j might be non-Gaussian, perhaps with heavy tails. To see the extent of the tails, a large sample of ϕ values is needed.

Option C uses as large a sample of ϕ values as possible, but relies on a classifier to estimate the b_j for a given ϕ. This implicitly assumes that b_j varies (somewhat) smoothly with ϕ. The classifier's parameters will probably not be well determined by the data. If it's a Bayesian classifier, one could sample from the posterior to see the combined uncertainty in b_j from this and from the prior on ϕ.

Handling Systematic Errors — Phase III

The distribution for b_j found in Phase II could be taken as a prior to be used in Phase III. (Or one can visualize it as a result of an imaginary experiment if one doesn't like priors.)

Option A: Assume this prior is Gaussian.

Option B: Like A, but we make the Gaussian assumption only if we fail to reject it from the data. Unfortunately, one needs quite a bit of data for a reasonable test.

Option C: Do something that handles non-Gaussian priors. Needed for B if the test rejects.

Maybe the whole approach is wrong, though...

Isn't the Actual Data Informative About ϕ?

So shouldn't we use a distribution of ϕ that is updated from the prior on the basis of the actual data?

A direct approach would be based on the full likelihood:

$$L(\phi, f) \;=\; \prod_i p(v_i|\phi, f)$$

where $p(v_i|\phi, f)$ is implicitly defined by the signal and background simulators. Integrating $L(\phi, f)$ with respect to the prior for ϕ would give the marginal likelihood for f (the signal fraction).

Unfortunately, we have no expression for $p(v_i|\phi, f)$, so the direct approach is out.

A Possible Approach?

Use the Phase I simulation data (with many ϕ values) to train a model for predicting ϕ_i from v_i.

Use Option C, so that we also train a classifier to separate signal and background given v_i and ϕ_i.

In Phase II, define cutpoints for the outputs of both these Phase I models, and use additional simulation data to find the probabilities of events falling in these bins as a function of ϕ.

From real events, we get a likelihood, $L(\phi, f)$, based on these bin counts.