A Way to Get Good Evaluation Functions from Fewer Data

Michael Frank

1.0 The Problem

We're making a game-playing program. We have a game search algorithm that can do the job, if it's given a fast source of a certain kind of probability information. For example, given a game position, we may want to quickly find our personal probability $\wp(Win)$ that we would win if we were in that game position. Or, if it's a game with several possible outcomes $o_1, \ldots, o_n$, we may want a probability distribution $\wp(o_i)$ over these outcomes. Or, for control of search, we may want, for each o_i, some approximate representation of a continuous distribution over the amount that our $\wp(o_i)$ for the position will change after some more computation. In any of these cases, the problem is the same: how do we do a quick, yet high-quality assessment of these probabilities?

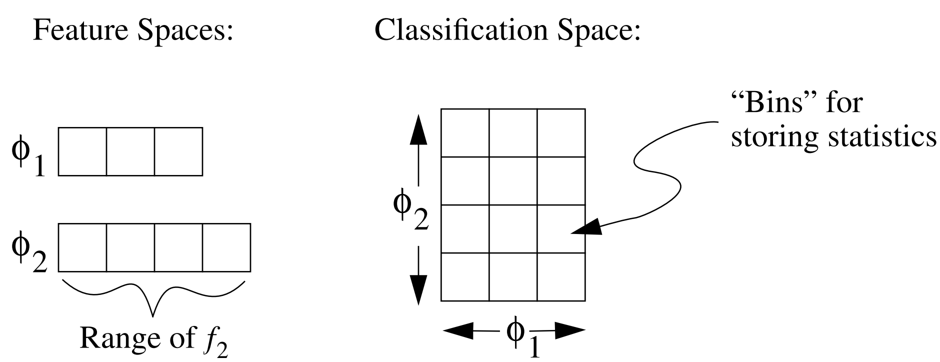
One good answer is to get these probabilities by collecting statistics. Unfortunately, in real games there are far too many positions to feasibly collect statistics for each of them individually. The standard solution to this problem is to collect statistics for whole classes of positions at once, without remembering the statistics for the individual positions within a class. The *classification space* for positions is often defined as the cross-product of a number of *feature spaces* $\phi_1, \ldots, \phi_k$, each of which is the range of a *feature* f_i, which is some easily-computed function of game positions (see Figure 1, "Classification Spaces," on page 2). We generally try to choose a classification space that we think will provide a lot of information about the variable we're interested in (the game outcome, or our future probability assessment).

Once we've defined our classification space, or set of bins, we can begin gathering statistics from our data. Each data point is assumed to be in the form of a game position P, together with the value of the variable y we are interested in. We classify P, look up a data storage bin for its class, and add y to the data stored in that bin. We might also associate y with some additional information about P that know relates to y in a certain special way; for example, if y is numeric, we might expect its value to be correlated with the values of certain numeric features $x_1, \ldots, x_k$; these features would then be left out of the feature set used to define the classification space, and instead their values would be stored along with the value of y. (A la Baum and Smith.)

Then, for each bin, we form a model of the data in that bin, in the form of a probability distribution over y (or, in the case where additional information is stored in the bin, a function from $x_1, \ldots, x_k$ to distributions over y). We might get the distribution by choosing the maximum-likelihood distribution given the data, out of some simple space of approximate representations of distributions.

So then we're done; when we see a game position, we simply look up its bin and use the stored probability distribution as our quick first assessment of our belief about which outcome will actually occur.

FIGURE 1. Classification Spaces



Unfortunately, we often find that it's hard to come up with one simple feature that does at well at predicting the outcome. Even combinations of 2 or 3 simple features might not be enough. We generally find that we need a lot of features in order for the statistics to discriminate the different outcomes well. However, this leads to another problem: the more features we have, the larger the classification space becomes (exponentially), and so a huge amount of data is required in order to get a statistically significant sample in each class.

The point of this paper is to briefly describe one possible solution to this problem. This is not original with me, but it's been a long time since I worked on this and I don't remember the references; I'm in the process of tracking them down.

2.0 The Proposed Solution

The basic idea is this: even though it may be true that no single feature, or combination of just a few features, discriminates outcomes very well, it may be the case that lots of these small classification spaces do *okay*. If only we could combine the recommendations of these small classification spaces in a principled way, we might get a good overall answer without having to resort to collecting statistics on the exponentially-large classification space that combines all the features.

For example, we could form a classification space for each combination of 1, 2, or 3 different features. Each data point can be classified within each space, so we only need to collect enough data to get a good sample in each bin of the largest such space.

But when it comes time to use the results, how do we combine the probability distributions produced by a position's classification in all the different spaces? We don't want to just add them together, because some of them will be much better predictors of the outcome than others; we want them to contribute to the answer proportionately. Is there any principled way to weight the distributions?

Well, here is one possibility. Each classification space, with probability distributions in all of its bins, can be thought of as a probabilistic model of the data, i.e., as a hypothesis that a position's outcome is determined by selecting a value from a its class's distribution. Thought of in this way, we can say that each classification space, each hypothesis, has some probability of being the "correct" one. If we could only compute this probability, it would make sense to use *that* as the space's weight. (Assuming that all these hypotheses are mutally exclusive—the damage of assuming this in practice is hoped to be small.)

We can compute model probabilities using Bayes' rule. Since these are probabilistic models, it's trivial to compute the likelihood of the data given the model. So the problem is reduced to one of finding sensible priors for all the models. This is not too critical since, with a good statistical sample in all of a model's bins, the differences in the data-likelihood between the different spaces ought to be the deciding factor.

In any case, the prior for a classification-space model is just the product of the priors for the distributions in each of its bins, since the classification-space can be thought of as a conjunctive hypothesis of the form "in this bin, the distribution is this; and in this other bin, it's this; etc. ."

How we compute a prior for the distributions in the bins depends on how we represent the distributions, and whether we have additional stored features (Baum and Smith's $x_1, \ldots, x_k$) complicating our model. However, one answer in the simple case is that the prior ought to be just e^H, where H is the entropy of the probability distribution in the bin; a high-entropy distribution carries less information and so should be *a priori* more probable.

2.1 Application of the Technique

A version of the above-described technique has been tried in God, a simple card-playing game; the results were very successful.

In God, one player is the "god," and makes up a rule determining what stacks of cards are "allowed." For example, the rule might say that a stack is allowed if and only if the top card has a different suit and than either of the next two cards. The other player starts with all the cards, and he attempts to play them one at a time on a growing stack; after each attempt the god says "yes," or "no;" if "no," the player must take back the card and try again. The player's goal is to get rid of the cards as quickly as possible. Assuming a greedy strategy, his goal on each play is to play the card that forms a stack having the highest probability of being allowed, based on the answers received so far. (Other rules prevent the God from making the rule too restrictive or complicated; I won't go into those here.)

One thing to note is that the data is very scant; in a well-played game the player receives only a few bits per card. Thus my collaborator Mark Torrance and I required a technique that used the data very efficiently. We defined about ten features of card stacks, for example,whether the top two cards were the same suit. We then formed classification spaces for every combination of one or two of these features, and derived predictions from past data as described above. The technique proved very effective in generating good predictions, our player was the best out of about a half-dozen God-playing programs written for a class.

2.2 Problems With this Approach

I already mentioned the problem that several classification spaces (with bins filled in) might turn out to embody the same probabilistic model of the data, and thus it will be improper to count them as mutually-exclusive hypotheses. This can happen, for instance, with the spaces ϕ_i and $\phi_i \times \phi_j$, if the outcome is totally independent of $f_j(P)$ given $f_i(P)$, because $\phi_i \times \phi_j$ will then represent the same hypothesis as ϕ_i. Thus ϕ_i's contribution to the total distribution will be unfairly magnified. This particular case should not be a serious problem, since $\phi_i \times \phi_j$ will have a negligible probability compared to ϕ_i, as a result of it being a much larger space and so having a much smaller prior. But if two identical features are included by accident, their contribution will be counted twice; and any two features that are not independent will degrade the prediction quality, to a degree that is still unclear.

Another problem is that we are considering only models that contain maximum-likelihood distributions for the data. What if we also considered, in each space, a continuum of nearby models that contained distributions that were slightly different? Might that yield different weights for our classification spaces?

To avoid this problem, we need to make sure that our distributions are not just "most likely" distributions but rather express the sum of all possible distributions, weighted by their probability given the data. If we do this, then we can be sure that when we calculate the probability of a classification space, we are truly calculating the probability of just the hypothesis that it's the right space, not the hypothesis that it's the right space *and* that the distribution in each node is such-and-such particular distribution.[1]

If the variable y has a small, discrete range (e.g., $\{\text{Win}, \text{Lose}, \text{Draw}\}$), and we accept a normal flat prior over the space of possible distributions over y, then there is no problem; a standard formula for the exact distribution is given by:

$$\wp(y_i) = \frac{m_i + 1}{M + n_y}^2 \qquad \text{(EQ 1)}$$

where m_i is the number of data points where $y = y_i$, M is the total number of data points $\sum_{j=1}^{n_y} m_j$, and n_y is the number of possible values of the variable y. This formula works well even on small amounts of data. Note that when there is no data, we get a flat distribution; this is the correct weighted sum of all possible distributions if all are equally likely. If

1. But if the hypothesis is just that the space is correct, shouldn't the space's prior depend only on the space itself, and not on the distributions stored in its bins? Maybe its probability should just be proportional to $e^{-\#\text{bins}}$. This needs more work.

2. This is the mean of the beta-multinomial distribution, for a uniform prior. The beta-multinomial distribution generalizes the binomial distribution to more than 2 outcomes.

there are two outcomes $\{\text{Win}, \text{Lose}\}$ and the data has 1 win and no losses in the bin, when we get $\wp(\text{Win}) = \frac{2}{3}$.

However, if the range of y is continuous (or just too large to store all the m_i's), as is the case if y represents a future probability assessment, then we cannot use (EQ 1) directly. In any case, we probably don't want our prior over distributions over such a large space to be flat; smooth, simple distributions seem more likely *a priori* than complex, jagged ones with high spatial frequencies.

One simple method that ignores high spatial frequencies but still allows multimodal distributions is to discretize y into evenly-spaced sections, and use (EQ 1) on those. However, this loses the distinction between a needle-sharp peak and a slightly smoother one that's still all in one discrete section.

I have thought of a few more complex ways to try to deal with this issue, but I'll save them until they're better developed.

3.0 A More Refined Version

I recently thought of a refinement of this technique that seems like it would produce even more accurate results; however, this new method has not been examined in detail or tested.

In the previous proposal, we weighted a classification space's prediction of a game-position's outcome by the probability that the entire classification space was the "right" model of the data. However, each game-position lies in only one bin in each classification space. So perhaps, the probability of that indivual bin, rather than that of the entire space, would be the more appropriate quantity for weighting the bin's prediction.

However, this raises a couple of new problems. It complicates the question of whether two alternative hypotheses are independent. Also, it complicates the comparison of the probability of the two hypotheses, since in general they will have been based on different data.

Further work remains to be done on this alternative.