Converting Utilities Into Probabilities

Kenneth W. Regan*
University at Buffalo

April 16, 2012

Abstract

Given events E_t each with some number $n_t \geq 2$ of different outcomes, fixed data D_t on the possible outcomes, the index i_t of the observed outcome, and overall parameter(s) Z, we have a function $P(Z; D_t) = \{ p_{t,j} : 1 \leq j \leq n_t \}$ giving projected probabilities for each outcome of each event. The aim is to fit Z to make the projected probabilities $P(Z, -)$ imply projected frequencies as close as possible to the observations. We envision the D_t to be simple, such as vectors of hindsight utility values $u_{t,j}$ for each outcome. Hence rather than learning how to select the best outcome, this application is to model fallible stochastic agents whose past choices are recorded by the i_t.

When all n_t and D_t are equal, so that $p_{t,j} = p_j$ depends only on j, one should clearly fit Z to minimize distance between the p_j and the observed frequencies of the outcomes j, and various fitting methods have good agreement. When the events are heterogeneous, however, the goal is less clear and the methods can diverge substantially. The "Percentile Fitting" method introduced in [Regan and Haworth, AAAI 2011] is developed further, and shown to out-perform maximum-likelihood and several other frequentist methods in the same application, namely where the outcomes are possible moves m_j in chess positions E_t and the utilities $u_{t,j}$ are values given for m_j by strong chess programs. Consequences for the expected-value statistic $\sum_j p_{t,j} u_{t,j}$, and conformance of experimental results to theoretical error bars for the new technique, are also studied.

Keywords. Computer games, chess, sequential decision making, probabilistic inference, machine learning, data mining, statistics.

1 Introduction

A running controversy in statistical modeling pits *frequentist* against *Bayesian* approaches, with *maximum likelihood* (ML) sometimes being considered separate from either.[1] This paper gives an application that provides both a wide variety of approaches for fitting its parameters, and a way to judge their performance on large data. We fully define the frequentist method introduced as "Percentile Fitting" (PF) in [RH11], show theoretically that it employs cumulative-distribution

*Department of CSE, University at Buffalo, Amherst, NY 14260 USA; (716) 645-4738; regan@buffalo.edu

[1] For one instance, see [Pet07].

information that Bayesian and ML methods do not, and demonstrate empirically that it gives better fits in this application. The data sets for our experiments comes from computer analysis of chess games, but there is no dependence on chess beyond obtaining the utility values from the computer program of various options (moves) available in a given situation (chess position).

The application samples *decision events* E in which choices m_i are made by fallible agents A among competing options $m_1, m_2, \ldots, m_n$. In terms of fixed data $D_1, \ldots, D_n$ about those options and variable parameters Z calibrating the skill of A, the goal is to infer probabilities $p_1, \ldots, p_n$ of choosing each option, say by an independent agent A' known only to have the same Z values as A. The training data comprises multiple such events, which we index as E_t from $t = 1$ to $t = T$. The events are assumed to be globally independent throughout this paper.

In the simplest case, every event has the same *arity* n and data $D_1, \ldots, D_n$. Then one defines the frequencies $f_j = |\{\, t : i_t = j \,\}|/T$ for $j = 1$ to n. Since the data are all alike, the projected probabilities $p_1, \ldots, p_n$ should be the same for all events. In case of complete freedom to project them, the simple frequentist prescription is to take $p_j = f_j$ for all j. The point of this paper is to analyze two ways in which the modeling can become less simple:

1. The function $P(Z) = (p_1, \ldots, p_n)$ has range only a restrictive subset of the probability space.

2. The events E_t are *heterogeneous*, by which we mean that the arity n_t and data $D_{t,1}, \ldots, D_{t,n_t}$ may vary widely.

We define several particular fitting methods, observe that they all agree in the simple case, show that the former change is already enough to distinguish them, and show marked effects of the latter change in a natural application with substantial data size.

These results all hold under specializing the data items $D_{t,i}$ to be real numbers $u_{t,i}$ representing *utility values*, ordering the indexing of outcomes to give $u_{t,1} \geq u_{t,2} \geq \ldots \geq u_{t,n_t}$ for all t, and restricting attention to projection functions P that give $p_{t,1} \geq p_{t,2} \geq \ldots \geq p_{t,n_t}$ for all t. That is, the analysis respects the condition that the model be *monotone* in the sense that higher utility for an outcome never lowers its projected probability. This also makes clear that the objective is *not* to find the outcome with greatest utility, but rather to model the probabilistic behavior of agents by inference from their past choices recorded in training data.

In the utility case we can make the arities equal by taking n to be the maximum n_t and *padding* all shorter tuples with nonce options having effectively negative-infinite utility. Heterogeneity still prevents regarding $m_{t,j}$ and $m_{u,j}$ for $u \neq t$ as "the same outcome j." Making all n_t equal is just a formal convenience, and the frequentist methods do not assume it. It does simplify saying that our applications have the general rubric of converting utility vectors $(u_1, \ldots, u_n)$ into probability vectors $(p_1, \ldots, p_n)$.

We describe the fitting methods studied here in a general way that applies to all cases, except that the final method is defined with respect to a fixed ordering of the outcomes for each event—intendedly but not necessarily the utility ordering.

2 Fitting Methods

In all cases we are given training data consisting of values n_t and $D_{t,i}$ for T-many events E_t, which in the utility case consists of a set U of vectors $(u_{t,1}, \ldots, u_{t,n_t})$ for each t, which we also call *tuples*.

Since the data are fixed, we can regard the probabilities $p_{t,j}$ as functions of Z alone, and suppress Z too where convenient.

2.1 ML and Bayes

The *Maximum Likelihood* method is now to fit Z to maximize the probability of the selected options i_t in the training data. By global independence this means to maximize

$$\prod_t p_{t,i_t}$$

which is equivalent to minimizing the log-sum

$$\sum_t \ln(1/p_{t,i_t}).$$

The notation and goal are unperturbed by the tuples being heterogeneous. We write z_{ML} for some value of Z that minimizes this logsum, and call $P(z_{ML})$ the *max-likelihood probabilities*.

Broadly speaking, *Bayesian iteration* in this setting satisfies generally-known conditions under which it gives convergence to the ML estimator [good reference needed]. For completeness and later use, we derive this. Let $A(z)$ denote the event that the agent in the training data with chosen options $\vec{i} = i_1, i_2, \ldots$ arises from $Z = z$. By Bayes' Theorem, assuming the space Z is finite,

$$\begin{aligned}
\Pr(A(z) \mid \vec{i}) &= \frac{\Pr(\vec{i} \mid A(z)) \Pr(A(z))}{\Pr(\vec{i})} \\
&= \frac{\Pr(\vec{i} \mid A(z)) \Pr(A(z))}{\sum_z \Pr(A(z)) \Pr(\vec{i} \mid A(z))} \\
&= \frac{\prod_t \Pr(i_t \mid A(z)) \Pr(A(z))}{\sum_z \Pr(A(z)) \prod_t \Pr(i_t \mid A(z))} \\
&= \frac{\prod_t p_{t,i_t}(z) \Pr(A(z))}{\sum_z \prod_t p_{t,i_t}(z) \Pr(A(z))}.
\end{aligned}$$

The standard "know-nothing prior" assumption $\Pr(A(z)) = 1/|Z|$ lets us simplify this even further to

$$\Pr(A(z) \mid \vec{i}) = \frac{\prod_t p_{t,i_t}(z)}{\sum_z \prod_t p_{t,i_t}(z)}.$$

Note that the global independence assumption not only creates a simple product over t but also makes the value independent of the order of presentation of the data for each t. Thus the Bayesian probability of $Z = z$ is just the normalized likelihood function.

Write N_z for $\prod_t p_{t,i_t}(z)$. Upon iterating the data d times, we get

$$\Pr(A(z) \mid \vec{i}^d) = \frac{N_z^d}{\sum_z N_z^d}.$$

Because $a^d = o(b^d)$ whenever $a < b$ and the vector of values N_z is finite, the right-hand side as $d \longrightarrow \infty$ converges pointwise to the Dirac delta function for the value z maximizing N_z, which is just $z = z_{ML}$ as before. (This also holds true under any fixed prior $A(z)$.)

Thus the peaks of the Bayesian probability curves approach the ML estimators. Large homogeneous training sets can be expected to behave like d-fold iterations of a smaller training set. In this paper we emphasize large-data situations, for which this convergence applies. Thus in this applicationm we regard ML as already capturing the objective of Bayesian iteration, and hence from here we do not distinguish the methods.

Note that in both cases, only the probabilities of the selected options m_{t,i_t} are involved in the formulas. The basis for the critique in this paper is that the ML and Bayesian approaches are not using all available information.

Thus we move on to frequentist approaches. We define $d(x, y)$ to be (the square of) a distance function, not necessarily supposing $d(x, y) = (x - y)^2$ for use with least-squares estimation. Since there is no notion of "same outcome," the first issue becomes how best to preserve the intuition of building frequencies for the outcomes. A second issue is scedasticity: some "bins" will have many more data points. We attempt to solve both issues by imposing a uniform percentile grid on the tabulations.

2.2 Percentile Fitting

The "Percentile Fitting" method of [RH11] attempts to avoid these choices and weighting issues. The method is to minimize a distance integral of the form

$$\int_{q=0}^{q=1} d(q, f_q(z))$$

where $f_q(z)$ is the *hit score for percentile* q defined as follows. The hit score is the average of the hit scores for each tuple t, so suppressing t we need only define $f_q(z)$ for one vector of projected probabilities $P(z) = (p_1(z), p_2(z), \ldots, p_n(z))$. Here is where the fixed ordering of outcomes is used. Let $i = i_t$ be the selected outcome for that tuple. Define

$$p = \sum_{j=1}^{i-1} p_j(z) \tag{1}$$

$$r = p + p_i(z) \tag{2}$$

and then define

$$f_q(z) = \begin{cases} 1 & \text{if } q \geq r \\ \frac{q-p}{r-p} & \text{if } p \leq q \leq r \\ 0 & \text{if } q \leq p. \end{cases} \tag{3}$$

Here is the frequentist intuition. Consider any fixed value of q, say $q = 0.60$, and consider any projected tuple $(p_1(z), p_2(z), \ldots, p_n(z))$. The parameter(s) z represent a way of stretching or compressing sub-intervals of width $p_k(z)$ in the unit interval. Let us suppose first that q is exactly at the upper end of interval p_k, meaning that $p_1(z)+p_2(z)+\cdots+p_k(z) = q$. Then we interpret z as representing the assertion that the probability of one of the first k options being chosen is exactly

q. That is to say, if $i \leq k$ then we call the tuple a "hit," else it is a "miss." So this particular z is asserting that the probability of a hit is q, and that is the z that we wish to find.

If q sits midway inside interval p_k, then we must consider how to score z in the case $i = k$. To interpolate correctly, let b be the ratio of the real-number distance of q from the left end of the interval to its width p_k. Then score this case as b of a hit. Thus z and q represent the assertion that the expected hit score for the tuple *at* percentile q is none other than q itself.

For each z and q, this prescription defines a criterion for scoring a hit for each tuple, and asserts that this expectation is q. Since the expectation is the same for each tuple, we have intuiitvely achieved the effect of the simple-frequency case, and can aggregate over the tuples. The frequency function $f_q(z)$ defined above tabulates the actual hit scores from the data. The degree of fit given by z for percentile q is then quantified as the distance between q itself and $f_q(z)$.

Treating q itself as a continuous parameter leads to minimizing the above integral. The one arbitrary choice we see is whether this should be weighted in terms of q. Minimizing

$$\int_{q=0}^{q=1} H(q)d(q, f_q(z))$$

instead is natural because having $H(0) = H(1) = 0$ reinforces the idea that the hit percentage projections are automatically correct at the endpoints $q = 0$ and $q = 1$. Apart from this, our intuitive point of using percentiles is that they skirt issues of skedasticity. Our experiments approximated this integral by summing over q in steps of 0.02, and in comparing the bins method, used simple bins of width 0.02. We abbreviate this method as PF.

2.3 Fitting the Derived Utility Statistic

In case the data are utility vectors $(u_1, \ldots, u_n)$, the projected probabilities also yield a *projected utility* from the tuple:

$$u(z) = \sum_j u_j p_j(z).$$

This can readily be summed or averaged over all tuples. Thus one can also fit z by equating the projected $u(z)$ with the actual utility u achievedin the training data. In cases where the objective is to see how often the agent makes the optimal choice, as well as modeling its average (falloff in) utility (from optimal), one can write two equations in the parameters Z. When Z comprises just two parameters, one can fit by solving two equations in two unknowns, equating $u = u(z)$ and the first-choice hit frequency $h_1 = |\{\, t : i_t = 1 \,\}|/T$ with the average of $p_{t,1}(z)$. This hybrid fitting method bypasses all of the above options, and hence acts as a helpful check on them. We abbreviate it FF for "first-choice and falloff."

3 Agreement and Differences in the Simple Homogeneous Tuple Case

.

In case the parameter space for Z allows a dense set of probability vectors, the simple case of repeated data (or equal utility vectors) allows exact fitting, and gives the same optimal z under any method.

Percentile Fitting agrees with Maximum Likelihood for homogeneous data and free parameterization of probabilities. Under suitable choices of bins, so does probability binning, and the optimum makes $h_1 = avg_t p_{t,1}(z)$ and $u = u(z)$.

Proof. Take $f_1, \ldots, f_n$ to be the frequencies from the data. The logsum minimand for ML then becomes

$$e(z) = f_1 \ln(1/p_1(z)) + f_2 \ln(1/p_2(z)) + \cdots f_n \ln(1/p_n(z)).$$

This is known to be minimized by setting $p_j(z) = f_j$ for each j, in accordance with the basic frequency idea, and the freedom assumption on z allows this to be realized. It remains to show that PF achieves the same minimum z.

For this z, let q be any percentile. If q falls on the endpoint r of any interval $p_k = p_k(z)$, then as $r = p_1 + p_2 + \cdots p_k = f_1 + f_2 + \cdots f_k$, the training data gives $f_q(z) = r = q$. Since other values of q occupy the same medial position in the same interval over all of the equal tuples, the interpolation gives $f_q(z) = q$ for all q, so the PF minimand is zero.

Also $h_1 = f_1 = p_1(z)$ and $u = \sum_j u_j f_j = \sum_j u_j p_j(z) = u(z)$, so the FF equations hold. Finally define bins with a_k being the average over probabilities in bin k. Since every tuple yields the same probability projections, every probability in the bin occurs T times, counting cases of $p_i = p_j$ for $i \neq j$ separately. It follows that a_k is also the overall frequency of hits in the bin, so the PB minimand is also zero. □

3.1 Differences Among ML/Bayes and

Now we give an example showing how far the equivalence can be disturbed by constraining the parameters. Let each tuple have outcomes m_1, m_2, m_3, and let the probabilities be given by $p_1(z) = z$, $p_2(z) = p_3(z) = (1 - z)/2$ for one numerical parameter $z \in [0, 1]$. Consider training data with $t = 5$ equal tuples, in which m_1 is chosen twice, m_2 twice, and m_3 once.

The ML logsum minimand is $2\ln(z) + 3\ln((1 - z)/2)$. Note that this is in fact independent of the frequencies of m_2 and m_3 provided they sum to $3/5$. A little calculus puts $z = 0.40$ as the optimum.

The PF minimand is a three-piece integral. The first piece integrates $d(q, \frac{2}{5}\frac{q}{z})$ from $q = 0$ to $q = z$. When $z = 0.40$, the reasoning of the proof of Theorem 3 shows that this first piece gives zero. The remaining two pieces do not, however, with the $2/5$ hit rate of m_2 and $1/5$ hit rate of m_3 both differing by 0.10 from their allocated widths of 0.30.

References

[Pet07] J. Petersen. Notes of frequentist, maximum likelihood, and bayesian statistics. http://genome-lab.ucdavis.edu/Links/ECL0_Petersen.pdf, 2007. Lecture notes of ECL 290 at U.C. Davis.

[RH11] K. Regan and G.M^{c}C. Haworth. Intrinsic chess ratings. In *Proceedings of AAAI 2011, San Francisco*, 2011.