The Traveling Algorithm Problem

(A Modicum of Mathematics
for Microscropic Middlemen)

Peter Cotton, Stony Brook, Sep 18, 2020

Outline

1. Hayek's defense of middlemen.
2. Examples of "microprediction" middlepeople.
3. Roaming algorithms
4. Two frameworks that help them choose which problems to address, only one of which is new.
5. Solving an optimization problem with 999,999 free variables.
6. Remarks on other uses.

Motivating the traveling algorithm problem

The big problem. How can bespoke statistics and optimization be efficiently produced and distributed to humanity?[1]

Some prodding from the economists

> What is the problem we seek to solve when we set out to establish a rational economic order?

Hayek *The Use of Knowledge in Society* [2]

[1] Just to be clear, humanity includes companies with revenues less than one billion per year

What's the problem?

> The peculiar character of the problem of a rational economic order is determined precisely by the fact that the knowledge of the circumstances of which we must make use never exists in concentrated or integrated form but solely as the dispersed bits of incomplete and frequently contradictory knowledge which all the separate individuals possess.

(And my wife thinks that *I* don't use enough commas)

A special economic good

Now suppose there is only one good in the economy.

The good is called *microprediction*.

Which is to say short term (minutes, hours), repeated prediction (more than $10,000$ observations, say)

Loosely, microprediction is the hard part of control, and thus all business optimization.[2]

The electricity powering intelligent applications.

A little more Hayek (five commas this time) with this in mind...

[2] Control and Reinforcement Learning are thinly veiled micro-prediction (of value functions or other intermediate results).

An ode to middlemen

> To know of and put to use a machine not fully employed, or somebody's skill which could be better utilized, or to be aware of a surplus stock which can be drawn upon during an interruption of supplies, is socially quite as useful as the knowledge of better alternative techniques. And the shipper who earns his living from using otherwise empty or half-filled journeys of tramp-steamers, or the estate agent whose whole knowledge is almost exclusively one of temporary opportunities, or the arbitrageur who gains from local differences of commodity prices, are all performing eminently useful functions based on special knowledge of circumstances of the fleeting moment not known to others.

Part I: Examples of middlepeople

Who are the middlepersons?

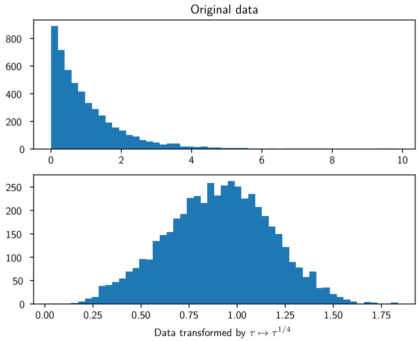
Who will take advantage of the *special knowledge of circumstances of the fleeting moment not known to others*?

Statistical algorithms!

(Otherwise it is too expensive)

What microprediction middlemen do. Example 1

Yang et al. [9]

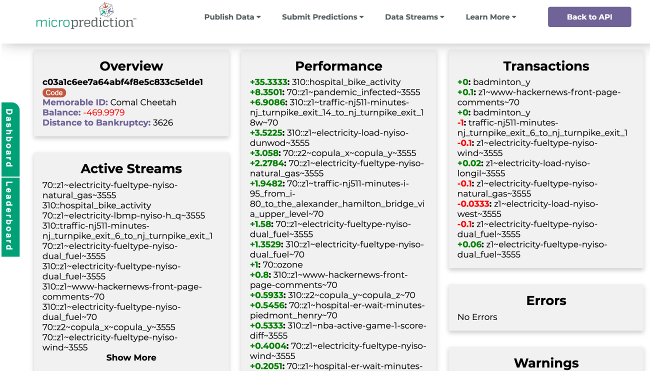


Exponential to normally via fourth root (or 0.2654)

Not impressed? Maybe you don't like housing brokers either. Overcome your prejudice against middlepeople.

What microprediction middlemen do. Example 2

An API matching problems to algorithms.



An algorithm that searches for time series to predict.

What microprediction middlemen do. Example 3

(An errors-in-variables inspired middleman).

The coefficient $\hat{b}$ is intended to represent an economic agent performing a cost-aware regression.

$$\hat{x}_j^{(1)} = x_j^{(1)} + \eta_j^{(1)} \qquad \eta_j^{(1)} \sim N(0, \sigma_1^2)$$

$$\hat{y}_j \quad \text{———} \quad \hat{b}$$

$$\hat{x}_j^{(2)} = x_j^{(2)} + \eta_j^{(2)} \qquad \eta_j^{(2)} \sim N(0, \sigma_2^2)$$

Buying precision to sell it?

The agent can reduce σ_1^2 or σ_2^2 by buying more precision.

The agent is compensated upstream depending on the accuracy of it's estimate $\hat{y}$ of y.

It is a repeated game.

Assume the truth ....

$$y_j = b_0 + \sum_{i=1}^{n} b_i x_j^{(i)} \tag{1}$$

True coefficients $b = (b_0, b_1, \ldots, b_n)$.

Estimate $\hat{b}_i$ will be attenuated due to the noise in $\hat{x}_i$.

Assume parent asymptotically learns the true coefficients b, irrespective of the child precision $\{p_i\}_{i=1}^n$.[3] Introducing notation γ let us say $\hat{b}_i \to \gamma_i b_i$ where $\gamma_1, \gamma_2, \ldots$ are shrinking parameters. This is because the parent uses the estimate

$$\hat{y}_j = \hat{b}_0 + \sum_{i=1}^n \hat{b}_i \hat{x}_j^{(i)} \to \gamma_0 b_0 + \sum_{i=1}^n \gamma_i b_i \hat{x}_j^{(i)}$$

with attenuated coefficients in order to avoid bias in $\hat{y}_j$.

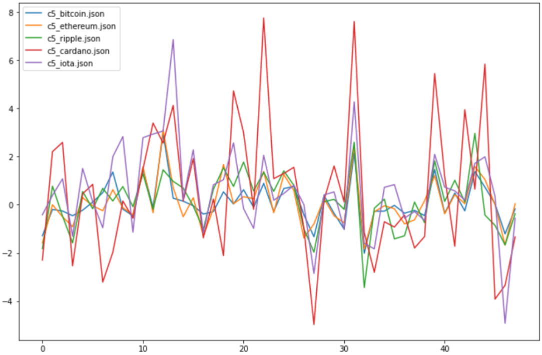
Subtracting this from the previous equation, we see that there will be error in $\hat{y}_j$ not purely attributed to weighted sums of errors in the ingredients $x_j^{(i)}$.

Thus the strategy for buying precision and reselling it is non-trivial due to attenuation. There is incentive to pay to discover more about the true coefficients in the short term, for instance.

[3] Identification is a lurking issue, as discussed in Bekker [1].

Microprediction middleperson. Example 4

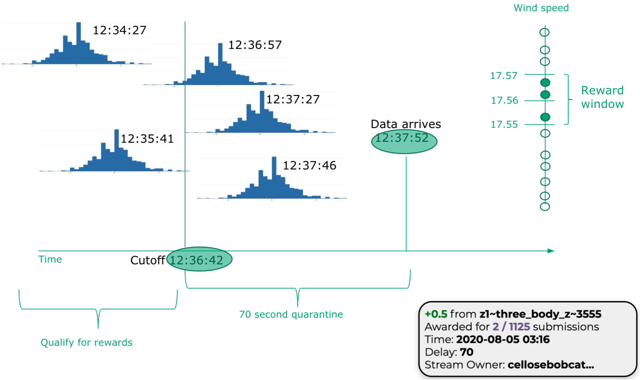
A “judge” receives submissions from algorithms that are predicting price changes in crytocurrencies.



Algorithms each supply 225 guesses of the next data point *after a fixed time horizon*.

Microprediction middleperson. Example 4 continued

Distributional predictions are quarantined.

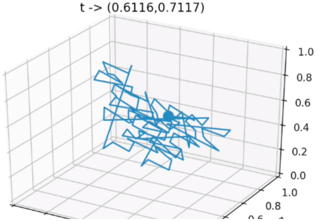


Then rewarded if close.

Microprediction middleperson. Example 4 continued

Implied percentiles $p_1, \ldots, p_5$ are computed for each cryptocurrency, for each prediction horizon (i.e. quarantine periods).

Groups of two and three percentiles are mapped back down to $(0, 1)$ via a space filling curve.

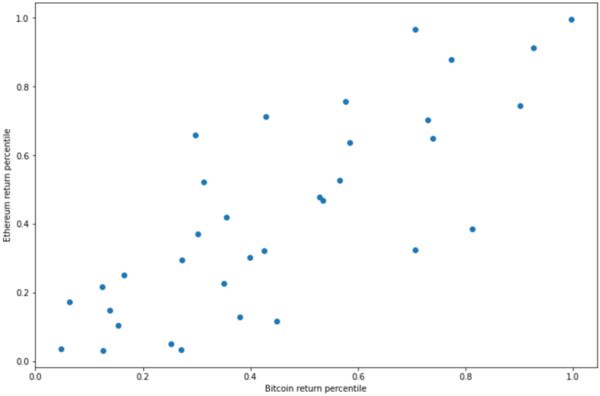


So that the process can repeat.

Microprediction middleperson. Example 4 continued

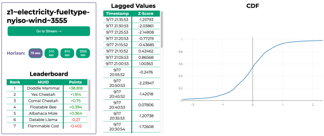
Further down the supply chain...

An algorithm examines the *implied copula* generated by the concensus percentiles p_1, p_2 representing bitcoin and ethereum.



Microprediction middleperson. Example 4

Elsewhere, another collection of algorithms fight to see if $\Phi^{-1}(p_1)$ is actually normally distributed



A micro-economy is performing multi-level residual analysis. Hoorah for middle-algorithms!

Part II: Release the hounds!

How algorithms travel today

What's the problem? Cost.

Time to lose the human chaperone.[4]

[4]Yes that's Graeme Garden

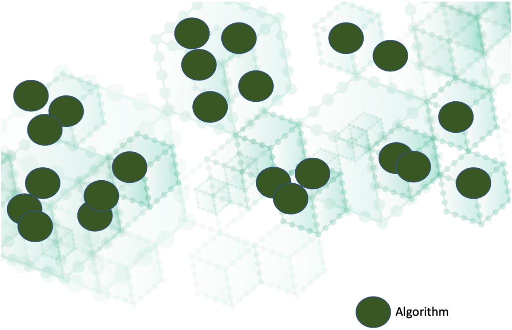
Why haven't we helped algorithms travel on their own?

Answers:

- Our dark hearts have frosted over
- We don't care about the disabled
- We like having moats around the problems that require generalized intelligence to cross because deep down we are bad, bad people
- Only large firms pay for statistical consulting
- Pretending that its hard to assess Machine Learning algorithms creates the illusion of value creation.[5]

[5] When a simple score suffices. We're talking microprediction, not prediction.

What would helping traveling algorithms look like?



1. Standing up the Stony Brook server
2. Publishing live data
3. Providing a starting port of call for algorithms that might travel to New Zealand and optimize the dairy supply chain.

Part III: Algorithms helping themselves

Assume mean spirited humans stop hiding problems from algorithms.

Assume algorithms face thousands or millions of potential opportunities to add value, but that each is competitive. How will they decide where to put their effort?

The traveling contestant problem

1. Model as all-pay auctions (review of key results)
2. Compute shadow prices of precision (sort of new stuff)

Approach 1: All pay auctions

A standard way to model all-pay auctions is to assume that participants place a subjective value on winning, then decide how much to bid. We label the players in decreasing order of the values they place on the item, namely $v_1, v_2, \ldots, v_n$.

Translating an auction into a contest, we'd say that the participant that puts in the most effort is declared the winner. Participants incur a cost per unit effort which is common to all players. They seek to maximize

$$utility = expected\ \overbrace{subjective\ reward}^{v_i} - effort$$

Equivalent formulation

Equivalently, we can assume that each player values the prize the same (and that value can be set to unity), but that players have different abilities (which multiply the effort they put in).

$$utility = expected\ \overbrace{reward}^{v=1} - effort \times \overbrace{production\ cost}^{inverse\ skill}$$

Two players

Nash equilibrium assumptions.

- Player 1 knows the strategy of player 2
- Player 2 knows the strategy of player 1

Both strategies are stochastic.[6]

The behaviour of both players is devised in such a way that the other, even with this knowledge, cannot take advantage.

[6] A deterministic strategy cannot be best. To follow such a strategy would allow another player to easily improve by investing just a little more effort. Therefore all players choose their efforts randomly.

It can be shown that the player who values the prize the most will choose effort randomly and uniformly between zero and the value v_2 that the second player ascribes to the prize.[7]

Conversely, the second player needs to keep the first player as honest as possible, and so will choose a random effort between zero and v_2 (to invest more would incur an obvious winner's curse).

It can be shown that the second player will make no effort at all with probability $1 - \frac{v_2}{v_1}$. Otherwise, the second player will draw uniformly in $(0, v_2)$.

[7] This is somewhat intuitive. The first player knows that the second player will never invest more effort than the value v_2 they place on the prize, but both players need to keep the other honest. For instance if the second player never invested more than $\frac{1}{2}v_2$ the first player could take advantage by investing just a little more.

Many players

1. The two players most desirous of winning will crowd out everyone else.

2. They will deploy the same strategy as if they only faced each other

3. All the other players will give up and not invest any effort.

Oddly efficient. No need for champion/challenger et cetera.

Approach II: Explicit shadow price computation

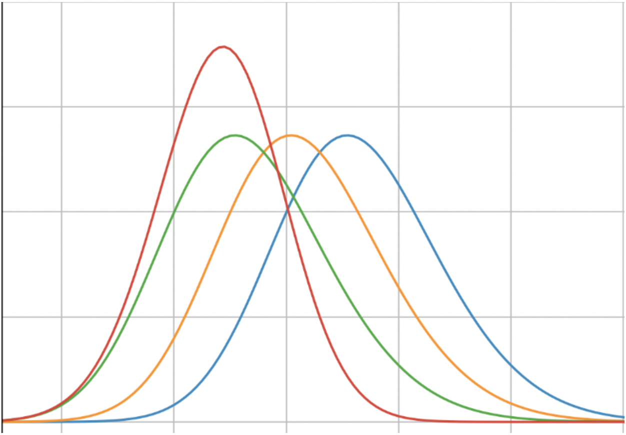
Auction theory is helpful when participation is viewed at a high level of abstraction. However, what if we know more about the relationship between effort and compensation? Remarks:

- Assume periodic, epoch based prizes. Winning is important because auction theory tells us that winner-take all all-pay auctions are the most efficient, from the perspective of the purse holder.
- Algorithms observe which other algorithms win, and how often.
- Possibly, algorithms' probabilities of winning are themselves the object of prediction.

We view winning as the result of ability, work but also performance noise.

Marginal effort calculations

Green (wins 1/2 the time), yellow (wins 1/3) and blue (1/6) score distributions. Distribution of best in red.



Question: how hard should yellow try?

Marginal effort calculations (cont)

Evidence available to yellow:

1. Cost of “performance”
2. Frequency of winning.
3. Competitors entering and leaving.

Costs are associated with moving the performance location parameter. Examples:

1. Fixed cost of gathering and storing independent samples implies linear cost of precision.
2. Cost of hyper-parameter search.
3. Cost of buying additional precision from a “child” supplying ingredient data feed (middle man role).

Aside: Luce's Axiom of choice

What happens when a competitor leaves?

(Seemingly a large literature that is all wrong, wrong, wrong)

Marginal effort calculations (cont)

Approach:

1. Calibrate effort scale.
2. Calibrate relative location parameters of all players to contest winning frequency.
3. Compute marginal prize-money versus shadow price of performance improvement.
4. Quit sometimes.

Only (2) presents any genuine numerical challenge in the presence of a large number of competing algorithms and/or a large number of contests in which one might choose to participate.

We are suggesting that one solution to the traveling contestant problem is provided the the *horse race calibration problem*...

Outline of the rest of my talk

- Review the horse race problem (briefly, as I have talked about it here before)

- New numerical results which are quite startling - I hope you think so anyway.

- Brief remarks on other applications of the traveling contestant/horse race problem (search, e-commerce etc)

The Horse Race Problem - recap

The continuous horse race problem assumes a density $f()$ with distribution $F()$ and seeks offsets $(a_1, \ldots, a_n)$ in order to satisfy

$$p_i = \int_{-\infty}^{\infty} f^{\rightarrow a_i}(x) \Pi_{j \neq i}^{n} \left(1 - F^{\rightarrow a_j}(x)\right) dx$$

for some specified winning probabilities p_i. Here $f^{\rightarrow a_i}$ is the density translated by a constant a_i.

Some special cases have attracted attention ...

Henery: normally distributed performance

The case of normal $f()$ is considered by [4] [3] and approximate analytical results derived by Taylor expansion:

$$a_i = \frac{(n-1)\phi\left(\Phi^{-1}(\frac{1}{n})\right)\left(\Phi^{-1}(p_i) - \Phi^{-1}(\frac{1}{n})\right)}{\Phi^{-1}\left(\frac{i-\frac{3}{8}}{n+\frac{3}{4}}\right)}$$

where ϕ and Φ are the standard normal density and cumulative distribution respectively.

Pythagorean formula

Bill James' Pythagorean formula for baseball

$$\overbrace{p}^{win\ probability} = \frac{\overbrace{RS^2}^{season\ runs}}{RS^2 + \underbrace{RA^2}_{season\ runs\ against}} \tag{2}$$

is precisely correct if runs scored in a game follows the Weibull distribution.[6]

Great for two horse races ... maybe.

More analytical possibilities ...

A theoretical comparison between Harville (exponential performance) and Henery's approach is made in [5]. An ad-hoc attempt to improve Harville by replacing p_i with p_i raised to a power β (then normalized across horses) is suggested. Other suggestions are made in [8] who previously noted the tractibility of the case of Gamma distributed X_i in [7].

Drawbacks to all of these: they don't really work. They aren't general.

The Discrete Horse Race Problem

Let $X_1, ..., X_n$ be discrete univariate contestant scoresassumed to take values on a lattice of equally spaced points. Let $X^{(k)}$ denote the k'th order statistic and in particular let $X^{(1)}$ denote the winning minimum score.

We define the implied state price for each contestant i as follows.

$$p_i = E\left[\frac{\iota_{X_i = X^{(1)}}}{\sum_k \iota_{X_k = X^{(1)}}}\right] \tag{3}$$

where ι is the indicator function. The price p_i is the expected payout in a game where we get

$$payoff = \begin{cases} 1 & if\ horse\ i\ wins \\ \frac{1}{2} & if\ tied\ with\ one\ other \\ \frac{1}{3} & if\ tied\ with\ two\ others \\ \dots & \end{cases}$$

It reduces to the probability of winning if there are no ties.

Approximate translation operator

For any $f : \mathbb{N} \to \mathbb{R}$ and any $a \in \mathbb{R}$ we define the shifted distribution $f^{\to a}(\cdot)$.

$$f^{\to a} = (1-r) f^{\to \lfloor a \rfloor} + r f^{\to \lfloor a \rfloor + 1} \tag{4}$$

where $r = a - \lfloor a \rfloor$ is the fractional part of the shift a obtained by subtracting the floor. This operation takes a density of X to one that approximates the density of $X + a$.

Remarks:

1. Exact translation if integer a
2. The same mean as $X + a$

The discrete horse race (calibration) problem

Given a distribution $f(\cdot)$ on the integers and a vector $\{p_i\}_{i=1}^n$ of state prices summing to unity, find a vector of offsets $(a_1, \ldots, a_n)$ such that the following holds for every i when the distribution of the i'th score X_i is given by $f^{\rightarrow a_i}$.

$$p_i = E\left[\frac{\iota_{X_i = X^{(1)}}}{\sum_k \iota_{X_k = X^{(1)}}}\right]$$

Remarks:

1. Can set $a_0 = 0$ w.l.o.g.
2. Don't really need $f()$ common across all horses.
3. A "mere" optimization ... but try it for $n = 300$

Solution to the discrete problem

It will be convenient to define

$$S_i(j) = Prob(X_i > j) = 1 - F_i(j) \tag{5}$$

as the i'th survival function.

Define the (conditional) multiplicity to be the expected number of variables that tie for the lowest value, assuming the lowest value is precisely j:

$$m(j) = E\left[\sum_{i=1}^{n} \iota_{X_i=j} | X^{(1)} = j\right] \tag{6}$$

Tie multiplicity calculus

Now suppose $X_1, ..., X_n$ represent the minimums of groups of (non-overlapping) variables, with respective multiplicities $m_1, ..., m_n$ respectively.

Take the union of the first two groups. The multiplicity is:

$$\begin{aligned} m^{(1)}_{1,2}(j) &\approx \frac{numer}{denom} \quad (7) \\ numer &= m_1(j)f_1(j)S_2(j) + (m_1(j) \\ &\quad + m_2(j))f_1(j)f_2(j) + m_2(j)f_2(j)S_1(j) \\ denom &= f_1(j)S_2(j) + f_1(j)f_2(j) + f_2(j)S_1(j) \end{aligned}$$

by careful accounting and some luck. See the paper for a more careful analysis.

Algorithm 1: Density and multiplicity

Input: Discrete densities $f_i : \mathbb{N} \to \mathbb{R}$, multiplicities $m_i : \mathbb{N} \to \mathbb{R}$ for $i \in \{1, \ldots, n\}$;

Initialize $S = S_1$, $f = f_1$, $m = m_1$;

for $i = 2$ *to* n **do**

  $S \to 1 - (1 - S)(1 - S_i)$;

  $f(j) = S(j) - S(j - 1)$ for all j;

  Assign $m(j)$ for all j using eqn 7 with f, m, S taking the role of group 1 and f_i, m_i, S_i the role of group 2.

end

Solution to the discrete horse race problem - strategy

The idea of our algorithm is to estimate the "best of the rest" density assuming some given offsets a_i, then adjust, then re-estimate and so forth. After each iteration and choice of a_i we compute the survival function S and multiplicity m for the set of all horses in the race.

Pricing against the rest

Recalling the implied state price:

$$p_i = E\left[\frac{\iota_{X_i=X^{(1)}}}{\sum_k \iota_{X_k=X^{(1)}}}\right]$$

Compute by conditioning on the winning score j:

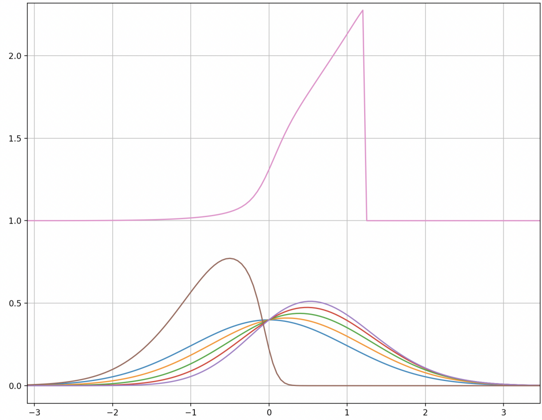
$$p_i = \sum_j \frac{f_{\hat{i}}(j) f_i(j)}{1 + m_{\hat{i}}(j)} \tag{8}$$

where $f_{\hat{i}}(j)$ and $m_{\hat{i}}$ are the density and multiplicity for the first order statistic in the complement of $\{j\}$ (i.e. all the competitors except horse i).

Multiplicity inversion

The multiplicity calculus can be inverted to determine the multiplicity for all-but-one of the horses.

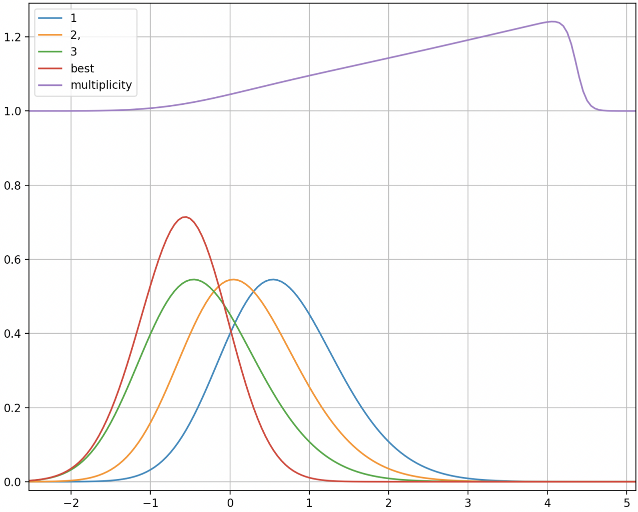
$$
\begin{aligned}
m_{\hat{i}}(j) &= \frac{numer}{denom} && (9) \\
numer &= m(j)f_1(j)S_{\hat{i}}(j) + m(j)f_1(j)f_{\hat{i}}(j) \\
&\quad + m(j)f_{\hat{i}}(j)S_1(j) - m_1(j)f_1(j)S_{\hat{i}}(j) \\
&\quad - m_1(j)f_1(j)f_{\hat{i}}(j) \\
denom &= f_{\hat{i}}(f_1 + S_1)
\end{aligned}
$$



Example of tie multiplicity (conditional expected number of horses tying for first place, conditioned on a fixed performance) for a larger race with 25 contestants. An evenly spaced lattice with 500 points supports the scoring distributions.

Algorithm 2: Iterative Solution to Discrete Horse Race Problem

input: Win probability p_i, discrete density f ;
Initialize $a_i = 0$ for all i. ;
while $\tilde{p}_i \not\approx p_i$ *for any* i **do**
  Apply Algorithm 1 to compute S, m, f from the collection of densities $f_i := f^{\to a_i}()$;
  for $i = 1$ ***to*** n **do**
    Compute $S_{\hat{i}}$, $m_{\hat{i}}$ using eqn 9 ;
    Compute implied state prices $\tilde{p}_i$ for all i using eqn 8 ;
  If any $\tilde{p}_i - p_i$ is too large assign new a_i for all i by assuming $a \to \tilde{p}$ is linear ;
output: a_i



Solution to our three horse example.

Summary: Horse Race Solution

We have exhibited a fast algorithm for inferring the location parameters of variables X_i when partial information is available:

1. The probability[8] that X_i is least among $X_1, \ldots, X_n$
2. The distribution of X_i up to a translation.

In contrast to previous work the distribution $f()$ is not assumed to fall into a known family, but kept general.

[8] Technically the state price in the discrete case

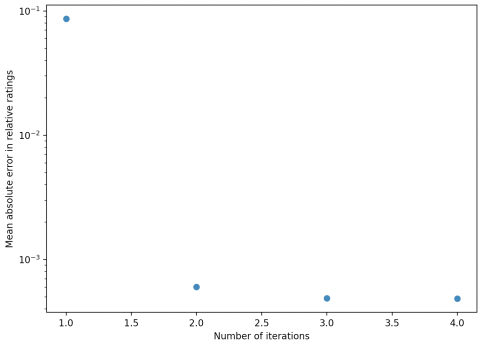
The Discrete Horse Race Solution - Numerical Results

Summary:

1. Convergence takes only a few iterations
2. A $100,000$ dimensional calibration problem can be solved in reasonable time

Convergence to ability (location parameter)

Worst result first :-)



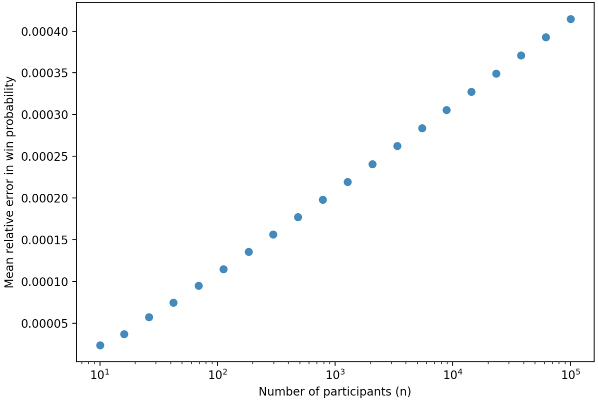
The relative error in implied relative location parameters for a synthetic race with 100 participants and a lattice of size 250.

Interpretation: Roughly 1 part in 1000.

Why not perfect?

Convergence to probability

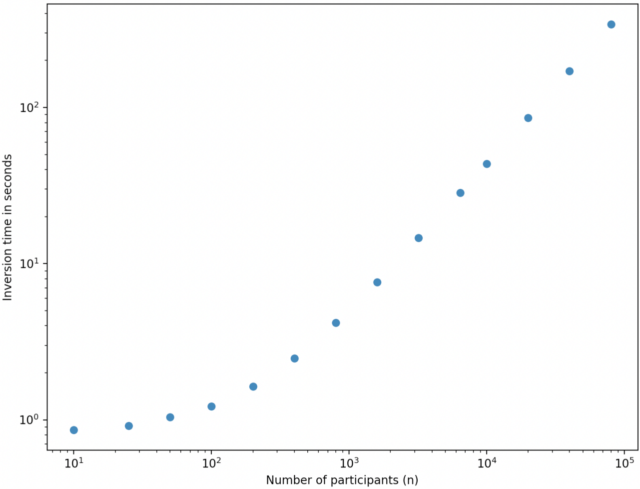
Relative error in implied probability post-calibration.



Interpretation: Even long-shots calibrate near-perfectly, even for a race with $10,000$ participants.

Computation time

Calibration time as a function of number of participants



Interpretation: Eventually sub-linear

New applications and remarks

From last time

In a stylized version of trade each dealer participates in a sealed bid auction, and the customer will take the best price offered.

Customer chooses who to call by optimizing

$$V^* = argmin_{V \subset S} \left\{ xE \left[\min_{i \in V} m_i \right] + I(V;x) \right\} \quad (10)$$

Can be shown to be a sub-modular minimization. Yeh!

Web search

In web search a horse race occurs every time a search phrase is entered. The winner of the race is the link that is clicked on. Risk neutral win probability ($\{p_i\}$) may not exist in quite the same fashion at the racetrack, but the vast number of searches that occur (at least for common phrases) leads to a precisely defined set of $\{p_i\}$ nonetheless.

The position of a link on the page strongly influences the user's decision (not completely dissimilar to horseracing where barrier position also matters) but we shall assume we are in the experimental phase and that a random permutation is applied so as not to bias any particular link. The user might in theory scroll or page down many times, so the size of this particular horse race might well run into the hundreds or more.

Placement, preference and Luce

An analogous situation occurs in e-commerce, where there may be sufficient volume to accurately imply a probability that a product wins out over others in a given category.

Services that try to estimate which image of a house or clothing item a person might click, when presented with numerous possibilities arrayed randomly.

These are examples of contests occurring with high velocity. The problem is not the estimation of $\{p_i\}$'s from a surfeit of historical data, but rather, inferring what probabilities will apply when a new very similar search is performed, or when some results are removed (in analogy to the scratching of a horse from a race).

References

[1] Paul A. Bekker. Comment on Identification in the Linear Errors in Variables Model. *Econometrica*, 2006.

[2] F Hayek. The Use of Knowledge in Society FA Hayek. *American Economic Review*, 1945.

[3] R. J. Henery. Permutation Probabilities as Models for Horse Races. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1981.

[4] R. J. Henery. Place probabilities in normal order statistics models for horse races. *Journal of Applied Probability*, 1981.

[5] Victor S. Y. Lo and John Bacon-Shone. A Comparison Between Two Models for Predicting Ordering Probabilities in Multiple-Entry Competitions. *The Statistician*, 2006.

[6] Steven J Miller. The pythagorean won-loss formula and hockey: A statistical justification for using the classic baseball formula as an evaluative tool in hockey. *arXiv preprint arXiv:1208.1725*, 2012.

[7] Hal Stern. Models for distributions on permutations. *Journal of the American Statistical Association*, 1990.

[8] Hal S. Stern. Estimating the Probabilities of the Outcomes of a Horse Race (Alternatives to the Harville Formulas). 2010.

[9] Zhenlin Yang and Min Xie. Process monitoring of exponentially distributed characteristics through an optimal normalizing transformation. *Journal of Applied Statistics*, 2002.