CSE702 Spring 2025 Week 4: Predictive Analytics

A *predictive analytic model* :

1. addresses a series of *situations*, each of which involves a set of outcomes $m_1, m_2, \dots, m_\ell$.
2. generates projected probabilities $p_1, p_2, \dots, p_\ell$ for the respective outcomes. And:
3. also generates confidence intervals $[a_1 < p_1 < b_1], [a_2 < p_2 < b_2]$,..., $[a_\ell < p_\ell < b_\ell]$ for these probabilities.

In my usage, point 3 distinguishes "predictive analytics" from mere "analytics." But what on earth does it mean to speak of a "95% probability interval" for one of your own projected probabilities? An outcome m_i either happens or it doesn't.

The point comes more into focus if we imagine analyzing a physical coin. Suppose the coin has a raised head and a slightly concave tail relative to its rim. Then we may estimate the probability p of tails at 0.51. Moreover, we want to be able to assert 95% confidence that the true physical probability $\check{p}$ is between 0.505 and 0.515. What does this *mean*? Basically this:

- Say that a "trial run" r flips the coin $1{,}000$ times and records the proportion t_r of tails.
- The assertion says that if we do 1,000 trial runs, then at least **950** of them will have $0.505 \ \leq\ t_r\ \leq\ 0.515$.

We've had to flip the coin a million times total to explain the concept. But what we did was not just estimate the coin, we tested and verified the *claimed precision* as well as accuracy of our estimate. That is to say, we did analytics of the prediction itself. Thus: **predictive analytics**.

(By the way, note this article about dependence on how the coin faces initially.)

The point about confidence intervals becomes more concrete if we add a fourth point to the definition: *A predictive analytic model*

4. projects risk/reward quantities v_i associated to the outcomes m_i.

Understood simply, the projected loss/value in the single situation is $E[v] = \sum_{i=1}^{\ell} p_i v_i$. But with repeated situations $t = 1, \dots, T$ we can project in that dimension too. If outcome m_1 at each time t, which we can label $m_{1,t}$, is the costly one, then the projected total loss is $\sum_{t=1}^{T} p_{1,t} v_{1,t}$. Or for total value/loss, we sum over both dimensions to get the expectation: $\sum_{t=1}^{T} \sum_{i=1}^{\ell_t} p_{i,t} v_{i,t}$.

Then the confidence intervals around $p_{1,t}$, or around all the projections $p_{i,t}$, translate into confidence intervals for these **aggregate statistics**. It is **not** as simple as saying that they are weighted sums of

$a_{i,t}v_{i,t}$ and $b_{i,t}v_{i,t}$. Consider $E[v]$ in the case $\ell = 2$ of just two outcomes m_1 and m_2, which are exhaustive and mutually exclusive. Further suppose $v_1 = -v_2$, $p_1 = p_2 = 0.5$, and $a_1 = a_2$, $b_1 = b_2$. If you expected the confidence interval to be $[a_1v_1 + a_2v_2,\ b_1v_1 + b_2v_2]$, then surprise! that's $[0, 0]$. If the true $\check{p}_1$ is at the bottom a_1 of its envelope, then we must have $\check{p}_2 = (1 - \check{p}_1) = (1 - a_1) = b_2$. Note that $b_2 = 1 - a_1$ and $a_2 = 1 - b_1$ must be true in general. Then $[a_1v_1 + (1 - a_1)v_2,\ b_1v_1 + (1 - b_1)v_2]$ is true in general, and when $v_2 = -v_1$ it becomes $[(2a_1 - 1)v_1, (2b_1 - 1)v_1]$. But now try the case $\ell = 3$...

For the aggregates over t, the simple sums $\sum_{t=1}^{T} a_{1,t}v_{1,t}$ and $\sum_{t=1}^{T}\sum_{i=1}^{\ell_t} a_{i,t}v_{i,t}$ put the bottom of the envelope far too low when the events for different t are **independent**. What's needed instead is to compute the **variance** var_t for each t. **If** the situations for different t are independent, then we get the overall variance as $\sum_t var_t$ by the rule that variances of independent events add. Taking the square root then gives an overall **standard deviation** σ around the estimate E for the expected value. Then the "two-sigma error bars" $[E - 2\sigma, E + 2\sigma]$ give (slightly more than) 95% confidence of bounding the true expected value.

[Some footnotes: Again it may seem weird to distinguish an "expected expectation" from a "true expectation." But when you are deciding whether to buy any financial instrument with risk, that's what you are hoping to equate---or put in a confidence range. The usual convention in statistics is to use a hat $\hat{}$ for a projected quantity, so we should start by saying that a predicive analytic model generates probability estimates $\hat{p}_1, \hat{p}_2, \dots, \hat{p}_\ell$ and so on. This could get cumbersome, so instead I'm using an inverted hat $\check{}$ for a ground-truth quantity. We will have to be careful not to get overconfident by confusing model projections with true values.

We haven't stated that the probabilities make $c_t = p_{1,t} + \cdots + p_{\ell_t,t}$ equal to 1 for each t. But if they don't, we can postulate a "null event" $p_{0,t}$ of value $v_{0,t} = 0$ and probability $1 - c_t$. Furthermore, if we define ℓ to be the maximum of ℓ_t over t, then we can pad every situation t with $\ell_t < \ell$ to have "dummy outcomes" $m_{\ell+1}, \dots, m_{\ell_t}$, each of probability 0. Thus we can pretend that ℓ is always the same for any situation t. Neither of these changes should affect either the projected variance var_t or its true counterpart $\check{var}_t$. Dividing by "ℓ" may not be meaningful even apart from the fact that the situations t need not have the same number ℓ_t of possible outcomes, but dividing by T to make averages out of the aggregates is always fine.]

In Chess

The situations t are chess positions with a given player to move. We can think of t as meaning "game turn." Then:

- The $m_1, \ldots, m_\ell$ are the legal moves in the position.
- Each m_i has a value v_i given by one or more strong chess programs (called **engines**).
- Traditionally v_i is in **centipawn units**: a possibly-negative integer of $1/100s$ of a pawn. When written to two decimal places we speak of "pawn units." Thus -150cp and -1.50 pawns are the same, meaning that the value is figuratively a pawn-and-a-half disadvantage.
- The engine itself orders the moves $m_1, \ldots, m_\ell$ in nonincreasing order of value: $v_1 \geq v_2 \geq \cdots \geq v_\ell$. Even though v_2 and further values may equal v_1, move m_1 is called the `bestmove` and is the one the engine will play in a game.

Now suppose we have generated projected probabilities $p_1, \ldots p_\ell$ for the choice of moves. Then:

- p_1 is the projected chance of making the computer's first move. Its variance is $p_1(1-p_1)$.
- $p = p_1 + p_2 + p_3$ is the projection for making one of the top three moves. Its variance is $p(1-p)$. If $\ell \leq 3$, so that this adds to 100%, then the variance is zero.
- $p_{EV} = \sum_{i:v_i=v_1} p_i$ is the projected probability of making a move that is either the first move or has equal-optimal value. Again the variance is $p_{EV}(1-p_{EV})$.

We may exclude positions with only one legal move as trivial. About 8--10% of posiitons have tied-optimal moves. Just over half of those are with $v_1 = v_2 = \ldots = 0.00$. This can depend on how long the engine is run in its Multi-PV mode and whether there is a turn-number cutoff to discard dead-drawn endgame positions.

- $E[v] = \sum_{i=1}^{\ell} p_i v_i$ is the projected position value after making the move. The projected centipawn loss is $E[\delta] = v_1 - E[v] = E[v_1 - v] = \sum_{i=1}^{\ell} p_i \delta_i$ where $\delta_i = v_1 - v_i$. Note that we could sum the latter from $i = 2$.

To compute the associated projected variance, we need to invoke the formula

$$Var(X) = E\left[(X - E[X])^2\right] = E\left[X^2\right] - E[X]^2.$$

Incidentally, in the case of first-line match, X is the indicator function: $X = 1$ if m_1 is played, else $X = 0$. Then $E[X] = E\left[X^2\right] = p_1$. So the variance is $p_1 - p_1^2 = p_1(1-p_1)$. Now in the case of the average centipawn loss, $E\left[\delta^2\right]$ is just $\sum_{i=1}^{\ell} p_i(v_1 - v_i)^2$. There isn't any better way to compute it than $Var(\delta) = E\left[\delta^2\right] - E[\delta]^2$. (I'm not sure exactly what the convention on square brackets versus parens is supposed to be, but I use parens to mean "variance of" as a standalone quantity, whereas brackets mean the item inside gets expanded.)

Is the variance of the value itself the same? Yes: The expectation of the value is $E[v_1 - \delta]$, which equals $v_1 - E[\delta]$. Now using the first definition of variance,

$$E\Big[(v_i - (v_1 - E[\delta]))^2\Big] \;=\; E\Big[(v_i - v_1 + E[\delta])^2\Big] \;=\; E\Big[(E[\delta] - \delta)^2\Big] \;=\; E\Big[(\delta - E[\delta])^2\Big] \;=\; Var(\delta).$$

One good conceptual point of using "deltas" comes from the scaling. This uses the generalization that taking a difference $v_1 - v_i$ is the same as integrating the "unit metric" $d\mu(x) = 1$ from $x = v_i$ to $x = v_1$. Now suppose we want to consider other metrics. Suppose we say the incremental value of an extra centipawn ($= 0.01$) value is at face value when the game is dead-even but tapers off the more one side has an advantage. Then we want $d\mu(0) = 1$ and $d\mu(x) < 1$ for $x \neq 0$. If it tapers off in "affine linear proportion" to the absolute value of x, then the metric we want is

$$d\mu(x) \;=\; \frac{1}{1 + C|x|}dx$$

for any fixed constant C. Now suppose for sake of convenience that v_i as well as v_1 is positive (that is, the move m_i is a mistake but it doesn't cost all the advantage). Then the **scaled difference** is

$$\int_{x=v_i}^{x=v_1} d\mu(x) \;=\; \int_{x=v_i}^{x=v_1} \frac{1}{1 + Cx}dx \;=\; \frac{1}{C}\ln(1 + Cx)\Big|_{x=v_i}^{x=v_1} \;=\; g(v_1) - g(v_i),$$

where g is the function $g(x) = \frac{1}{C}\ln(1 + Cx)$. The essence is that we can precompute all the "scaled values" $v' = g(v)$ ahead of time, and then $\delta' = v_1' - v_i'$ becomes the simple "**scaled delta**." The definition of variance for scaled difference is then much the same as for unscaled difference, just with δ' in place of δ. Note also that $g(0) = 0$, so the "constant of integration" can be taken as zero.

If $v_1 \leq 0$ then $v_i \leq 0$ too since $v_i \leq v_1$ and the calculation is much the same. If $v_1 > 0$ but $v_i < 0$ (a mistake that puts you suddenly at a disadvantage), then you have to break up the computation into two pieces, one from v_1 down to 0 and then from 0 down to v_i. But actually, simply making $g(v_i)$ negative when v_i is negative handles this case gracefully too.

A final point is that this metric formulation immediately explains why the slope of average error is steeper on the negative side in the diagrams linked here. The scaling represents the psychologically perceived magnitude of the error. An average error e made when you are a pawn behind is greater in the diagram than an average error d when you are a pawn ahead. But e goes through a thinner part of the metric from -1.00 to $-1.00 - e$, whereas d goes from $+1.00$ to $1 - d$ through the fattest part of the metric. The thin/fat difference makes the scalings e' and d' come out pretty much equal. [In fact, my code makes $g(x)$ follow the slopes of those lines directly, as a function of rating, rather than use the particular logarithmic metric.]

Using **expectation loss** instead of (scaled) centipawn loss is a headache because the expectation values depend on rating all the time, but the mathematical procedure is similar.

Aggregate Stats

Now we add these up and take averages over sets of T-many positions. We immediately get projections for our major raw metrics:

- Projected **T1**-Matches: $\sum_{t=1}^{T} p_{1,t}$. As a percentage ("MMP" in my files): $\frac{1}{T}\sum_{t=1}^{T} p_{1,t}$.
- Projected **EV**-Matches: $\sum_{t=1}^{T} p_{EV,t}$, average version $\frac{1}{T}\sum_{t=1}^{T} p_{EV,t}$
- Average Centipawn Loss (**ACPL**, unscaled): $\frac{1}{T}\sum_{t=1}^{T} E[\delta_t]$.
- Average Scaled Difference (**ASD**): $\frac{1}{T}\sum_{t=1}^{T} E[\delta']$.

If game turns were independent, the variances of the summed quantities would simply add over t. The variances of the averages wound then divide by T^2. But...but... The resulting *adjusted variances* give rise to *adjusted sigmas* σ' and ***adjusted z-scores*** via the recipe:

$$z' = \frac{actual - projected}{\sigma'}$$

[Demo the program---didn't quite get to this, will begin that way tomorrow.]