CSE702 Week 4A: Normality, Mapping, CLT, Screening Test

Show https://cse.buffalo.edu/~regan/Talks/CrossLabsOct2022.pdf

Show https://en.wikipedia.org/wiki/Normal_distribution

Any two normal distributions $\mathcal{N}(\mu_1, \sigma_1)$ and $\mathcal{N}(\mu_2, \sigma_2)$ can be mapped onto each other.

- $\mathcal{N}(0, 1)$ is **standard**.
- $\mathcal{N}(50, 5)$ approximates the distribution of flipping a fair coin 100 times.

Let's consider just the T_1-match for the time being.
The Central Limit Theorem (CLT) says that whatever goes on distributionally inside players' heads, the distribution of the T_1-match %, being a mean of an independent(??) sample, approaches the normal distribution as the number n of samples used in the mean grows. The "Rule of 30" is a convention that $n = 30$ is usually good enough.

CLT presumes

- **independent** samples
- from the **same** (unknown) distribution

Neither assumption holds in chess:

- Consecutive chess moves by a player in a game used to form the sample are not independent. Carlsen-Anand Double Blunder example: Anand missed his opportunity because he was fixated on moving his rightmost pawn down the board.
- The moves in a sample are all from different positions.

Nevertheless, the distributions obtained are fairly close to normal.
[show examples]

Hence the screening scores of the form $value = 50 + 5z$ can be treated as "nominal z-scores."

Main Question: Is it fair to use them for judgment, without using predictive analytics?
(My answer: **No**. But things like this get done in the world at large.)

How the Screening Test Works

The main points are:

- The screening test is completely data-driven. There is no "theory".
- It involves simple raw counting of features of games with regard to a chess engine (or engines) used as a benchmark. It is "entirely objective" in my parlance.

- However, it makes no attempt to distinguish "easy" chess positions from "difficult" ones.
  - A player who makes his/her own games easier to play may incur artificially high screening scores. (J.-R. Capablanca? Anatoly Karpov? Wesley So?)

Currently, my screening uses only two raw metrics: T1-match and ASD. The ASD is computed in the rating-independent manner of my "When Data Serves Turkey" article, which is there shown to be imperfect---but a followup regression over ratings mitigates the imperfection. Here is roughly the process I use:

1. For each rating "bucket" R in the training sets, compute the mean T1-match μ_R and its standard deviation σ_R among games by players in that bucket.
2. Smooth things out by doing ~~linear~~ cubic (back to linear after the Sonas correction?) regressions, first μ_R versus R to obtain values $\widehat{\mu}_R$, then separately σ_R versus R to obtain smoothed values $\widehat{\sigma}_R$. Weight the regressions by the numbers of positions in each bucket.
3. Note that the regressions give mappings $\mu_{T1}(R) = \widehat{\mu}_R$ and $\sigma_{T1}(R) = \widehat{\sigma}_R$ continuously for any rating R, not just "bucketwise."
4. Repeat steps 1--3 for ASD to obtain mappings $\mu_{ASD}(R)$ and $\sigma_{ASD}(R)$.
5. Also do regressions to fit and compute a mapping $\rho(R) =$ the *covariance* between the T1 values and ASD values at rating R, normalized as their Pearson correlation.
6. Travel ahead in time to pick a "magic number" n_0. I got $n_0 = 187$ as usable for all kinds of chess: Standard, Rapid, and Blitz. Then travel back in time to whee you were.
7. Given an actual performance mm for T1-match and asd for ASD from n game positions, form the ersatz scores
$$z_{T1} = \frac{mm - \mu_{T1}(R)}{\sigma_{T1}(R)}\sqrt{\frac{n}{n_0}} \quad \text{and} \quad z_{ASD} = \frac{asd - \mu_{ASD}(R)}{\sigma_{ASD}(R)}\sqrt{\frac{n}{n_0}}.$$
8. Finally, combine the two into one score using the generalized Fisher-Stouffer rule:
$$z = \frac{z_{T1} + z_{ASD}}{\sqrt{2 + 2\rho(R)}}.$$
9. Except, hide the fact that this is not really a z-score---and not to be used as one for judgment purposes---by transporting it onto the scale of 100 flips of a fair coin:
$$ROI = 50 + 5z.$$

A **better** way to do this is to make a single combination $C(mm, asd)$. Even better, do $C(mm, asd, n)$ to return an absolute number that scales with the number n of positions. Then you only have to do steps 1--3 once, to obtain mappings $\mu_C(R) =$ the regressed mean value of C at rating R and $\sigma_C(R) =$ the smoothed-out standard deviation of C as a function of R. Then just do
$$z = \frac{C(mm, asd, n) - \mu_C(R)}{\sigma_C(R)}$$
and $ROI = 50 + 5z$ as before. Well, there are two reasons I've shied away from this:

- T1-match and ASD do not have common units. Any combined score would seem to make an arbitrary decision in how to weight them. Whereas, the above process maps them to normalized units ahead of time, so they are combined with equal weighting.
- I am essentially doing this anyway, using the combination function $C^*(mm, asd, n) \;=\;$ the z in step 8. Well, then I would have to compute $\mu_{C^*}(R)$ and $\sigma_{C^*}(R)$ and finally combine them as above. The final secret is that I computed the magic number $n_0 = 187$ just so that this further step would give the identical answer to step 8.

Clear as mud? Well, in fact this is not what I actually do either---I build a C' that shortcuts the Fisher-Stouffer rule with a close-but-fudged number for the correlation, then use n_0 to make the resulting fudged measure agree in variance as well as mean with the training data.

Combining Z-Scores

One takeaway for possible future reference is the following formula for combining multiple z-scores $z_1, \ldots, z_k$ with arbitrary nonnegative weights vector $\boldsymbol{w} \;=\; (w_1, \ldots, w_k)$ and normalized (e.g. Pearson) covariance matrix $\boldsymbol{P}_{i,j} \;=\; \rho(z_i, z_j)$. Note that the main diagonal has all-1s, since a z-score is perfectly correlated with itself. The combination formula is

$$z \;=\; \frac{\sum_i w_i z_i}{\sqrt{\boldsymbol{w}^T \boldsymbol{P} \boldsymbol{w}}}.$$

- When the weights are all equal, this is the sum of the z-scores divided by the square root of the sum of the entries in the covariance matrix.
- When the weights are all equal and the z-scores are independent, the off-diagonal elements are all zero, so you get Stouffer's Rule proper: the sum of the z-scores divided by $\sqrt{k}$.
- When the weights are all equal and the z-scores are perfectly correlated, then you get the sum divided by k---i.e., the average of the z-scores.
- When the z-scores are independent but the weights are not necessarily equal, the denominator becomes the square root of $\boldsymbol{w}^T \boldsymbol{I} \boldsymbol{w}$, which is the Euclidean norm of $\boldsymbol{w}$. So you get

  $$z \;=\; \frac{\sum_i w_i z_i}{||\boldsymbol{w}||}.$$

  When $\boldsymbol{w}$ is a Euclidean unit vector, the denominator becomes 1, so you get the weighting of the z-scores. This does not, however, happen so simply when w is a linear combination summing to 1, i.e., an L1-norm unit vector. If w can be complex, we might verge into quantum mechanics...

Too wit, suppose we have two independent z-scores z_1 with weight 2/3 and z_2 with weight 1/3. If $w = (2/3, 1/3)$ were considered a unit vector, then these would combine to give $z \;=\; \frac{2}{3}z_1 \;+\; \frac{1}{3}z_2$.

However, z_1 is the same as two completely correlated z-scores $z_1' = z_1$ of weight 1/3 each. So the equal-weights rule applies and we get $z = z_1' + z_1' + z_2$ divided by the square root of the sum of the covariance matrix $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, so we get

$$z = \frac{2z_1 + z_2}{\sqrt{5}}.$$

which is not the same. The Euclidean norm of w is $\sqrt{4/9 + 1/9} = \sqrt{5}/3$, so this agrees with the formula in blue. Conclusion: It is logical for Nature to be quantum, not classical!

My reference for the formula in magenta is a 2011 paper by Dmitry V. Zaykin: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3135688/ Can anyone find a simpler reference, say in a machine-learning book?