Why the Best Predictive Models Are Often Different from the Best Explanatory Models: A Theoretical Explanation

Songsak Sriboonchitta[1], Luc Longpré[3]
Vladik Kreinovich[3], and Thongchai Dumrongpokaphan[2]

[1]Faculty of Economics, [2]Dept. of Mathematics, Chiang Mai University, Thailand, songsakecon@gmail.com, tcd43@hotmail.com
[3]University of Texas at El Paso, El Paso, Texas 79968, USA, longpre@utep.edu, vladik@utep.edu

1. Predictive vs. Explanatory Models: Traditional Confusion

- Many researchers implicitly assume that predictive and explanatory powers are strongly correlated.
- They assumed that a statistical model that leads to accurate predictions also provides a good explanation.
- They also assume that models providing a good explanation lead to accurate predictions.
- In practice, models that lead to good predictions do not always explain the observed phenomena.
- Vice versa, models that explain do not always lead to most accurate predictions.

2. Predictive vs. Explanatory Models: Example

- Newton's equations provide a very clear explanation of why and how celestial bodies move.
- In principle, we can predict the trajectories of celestial bodies by integrating the corresponding equations.
- This would, however, require a lot of computation time on modern computers.
- On the other hand, people successfully predicted the observed positions of planets way before Newton.
- For that, they use *epicycles*, i.e., in effect, trigonometric series.
- Such series are still used in celestial mechanics to predict the positions of celestial bodies.
- They are very good for predictions, but they are absolutely useless in explanations.

3. Remaining Problem: Why?

- The empirical fact that the best predictive models are often different from the best explanatory models.
- But from the theoretical viewpoint, this empirical fact still remains a puzzle.
- In this talk, we provide a theoretical explanation for this empirical phenomenon.

4. Need for Formalization

- In order to provide a theoretical explanation for the difference, we need to first formally describe:
  - what it means for a model to be the best predictive model, and
  - what it means for a model to be the best explanatory model.
- The "explanatory" part is intuitively understandable.
- We have some equations or formulas that *explain* all the observed data.
- This means that all the observed data satisfy these equations.

5. Need for Formalization (cont-d)

- Of course, these equations must be checkable – else:
  - if they are formulated purely in terms of complex abstract mathematics,
  - so that no one knows how to check whether observed data satisfy these equations or formulas,
  - then how can we know that the data satisfies them?
- Thus, when we say that we have an explanatory model, what we are saying is that we have an algorithm that:
  - given the data,
  - checks whether the data is consistent with the corresponding equations or formulas.
- From this pragmatic viewpoint, by an explanatory model, we simply means a program.
- Of course, this program must be non-trivial.

6. Need for Formalization (cont-d)

- It is not enough for the data to be simply *consistent* with the data.
- Explanatory means that we must *explain* all this data; for example:
  - if we simply state that, in general, the trade volume grows when the GDP grows,
  - all the data may be consistent with this rule.
- However, this consistency is not enough: for a model to be truly explanatory.
- It needs to explain *why* in some cases, the growth in trade is small and in other cases, it is huge.
- In other words, it must explain the exact growth rate.
- Of course, this is economics, not fundamental physics.

7. Need for Formalization (cont-d)

- We cannot explain all the numbers based on first principles only.
- We have to take into account some quantities that affect our processes.
- But for the model to be truly explanatory we must be sure that,
  - once the values of these additional quantities are fixed,
  - there should be only one sequence of numbers that satisfies the corresponding equations or formulas,
  - namely, the sequence that we observe (ignoring noise, of course).
- This is not that different from physics.

8. Need for Formalization (cont-d)

- For example, Newton's laws of gravitation allow many possible orbits of celestial bodies.
- However, once you fix the masses and initial conditions, Newton's laws uniquely determine the orbits.
- In algorithmic terms, if:
  - to the original program for checking whether the data satisfies the given equations and/or formulas,
  - we add checking the values of additional quantities,
  - then the observed data is the only possible sequence of observations that is consistent with this program.
- Once we know such a program that uniquely determines all the data, we can, in principle, find this data.
- We can try all possible combinations of possible data values until we satisfy all the corresponding conditions.

9. Need for Formalization (cont-d)

- How can we describe this in precise terms?
- All the observations can be stored in the computer, and in the computer, everything is stored as 0s and 1s.
- From this viewpoint, the whole set of observed data is simply a finite sequence x of 0s and 1s.
- The length n of this sequence is known.
- There are 2^n sequences of length n.
- There are finitely many such sequences, so we must potentially check them all.
- Thus, we find the desired sequence x – the only one that satisfies all the required conditions.
- Of course, for large n, the time 2^n can be unrealistically astronomically large.

10. Need for Formalization (cont-d)

- So, we are talking about *potential* possibility to compute – not practical computations.
- One does not solve Newton's equations by trying all possible trajectories.
- But it is OK, since our goal here is:
  - not to provide a practical solution to the problem,
  - but rather to provide a formal definition of an explanatory model.
- For the purpose of this definition, we can associate each explanatory model:
  - not only with the original checking program,
  - but also with the related exhaustive-search program p that generates the data.
- The exhaustive search part is easy to program.

11. Need for Formalization (cont-d)

- It practically does not add to length of the original checking program.
- So, we arrive at the following definition.
- *Let a binary sequence x be given. We will call this sequence* data.
- *By an* explanatory model, *we mean a program p that generates the binary sequence x.*
- The above definition, if we read it without the previous motivations part, sounds very counter-intuitive.
- However, we hope that the motivation part has convinced the reader.
- For each data, there is at least one explanatory model.
- Indeed, we can always have a program that simply prints all the bits of the given sequence x one by one.

12. What Do We Mean by the Best Explanatory Model: Analysis of the Problem

- There are usually several possible explanatory models, which of them is the best?
- To formalize this intuitive notion, let us again go back to physics.
- Before Newton, the motion of celestial bodies was described by epicycles.
- To accurately describe the motion of each planet, we needed to know a large number of parameters.
- In the first approximation, the orbit is a circle.
- We need to know the radius of this circle, the planet's initial position on this circle, and its velocity.
- In the second approximation, we have a circular motion that describes the deviation from the circle.

13. Analysis of the Problem (cont-d)

- We need to know similar parameters of this auxiliary circular motion.
- In the 3rd approximation, we need to know similar parameters of the 2nd auxiliary circular motion, etc.
- Then came Kepler's idea that celestial bodies follow elliptical trajectories.
- Why was this idea better than epicycles?
- Because now, to describe the trajectory of each celestial body, we need fewer parameters.
- All we need is a few parameters that describe the corresponding ellipse.
- These original parameters formed the main part of the corresponding data checking program.

14. Analysis of the Problem (cont-d)

- Thus, these parameters form the main part of the resulting data generating program.
- By reducing the number of such parameters:
  - we thus drastically reduced the length of the checking program,
  - and thus, of the generating program corresponding to the model.
- Similarly, Newton replaced all the parameters of the ellipses by a few parameters describing the bodies.
- This described not only the regular motion of celestial bodies.
- He also described the tides, he described (explained) why apples from a tree fall down and how exactly, etc.

15. Analysis of the Problem (cont-d)

- Here, we also have fewer parameters needed to explain the observed data.
- Thus, we get a much shorter generating program.
- From this viewpoint, a model is better if its generating program is shorter.
- Thus, the best explanatory model is the one which is the shortest.
- *We say that p_0 is the* best explanatory model *if it is the shortest of all explanatory models for x:*

$$\mathrm{len}(p_0) = \min\{\mathrm{len}(p) : p \text{ generates } x\}.$$

16. What Do We Mean by the Best Predictive Model

- If a trade model takes 10 years to predict next year's trade balance, we do not need it.
- We can as well wait a year and see for ourselves.
- For a model to be useful for predictions, it needs not just to generate the data x but to generate them *fast*.
- The overall computation time includes both:
  - the time needed to upload this program into a computer – which is proportional to $\text{len}(p)$,
  - and the time $t(p)$ needed to run this program.
- The smaller this overall time $\text{len}(p)+t(p)$, the better.
- *We say that p_0 is the* best predictive model for x *if:*

$$\text{len}(p_0)+t(p_0)=\min\{\text{len}(p)+t(p) : p \text{ generates } x\}.$$

17. Main Result: Formulation and Discussion

- *No algorithm is possible that, given data x, generates the best explanatory model for this data.*
- *There exists an algorithm that, given data x, generates the best predictive model for this data.*
- These results explain why the best predictive models are often different from the best explanatory models.
- If they were the same, then the above algorithm would always generate the best explanatory models.
- However, we know that such a general algorithm is not possible.

18. Proof for Predictive Case

- We want to find the program that generates the given data x in the shortest possible overall time T.
- We start with $T = 1$, then take $T = 2$, $T = 3$, etc.
- We stop when we find the smallest value T for which such a program exists.
- For each T, we need to look for programs from which $\text{len}(p) + t(p) = T$.
- For such programs, we have $\text{len}(p) \le T$.
- So we can simply try all possible binary sequences p of length not exceeding T.
- There are finitely many strings of each length.
- So there are finitely many strings p of length $\text{len}(p) \le T$, and we can try try them all.

19. Proof for Predictive Case (cont-d)

- For each of these strings, we first use a compiler to check whether this string is a program.
- If it is not, we simply dismiss this string.
- If the string p is a syntactically correct program, we run it for time $t(p) = T - \text{len}(p)$.
- If p generates x, we have found the desired best predictive model.
- So we can stop:
  - the fact that we did not stop our procedure earlier, when we tested smaller values of the overall time
  - means that no program can generate x in overall time $< T$ and thus,
  - that the overall time T is indeed the smallest possible.

20. Discussion

- The above algorithm is an exhaustive-search-type algorithm, that requires exponential time 2^n.
- Yes, this algorithm is not practical – but practicality is not our goal.
- Our goal is to explain the difference between the best predictive and the best explanatory model.
- From the viewpoint of this goal, this slow algorithm serves its purpose.
- It shows that:
  - the best predictive models can be computed by *some* algorithm, while,
  - as will now prove, the best explanatory models *cannot* be computed by *any* algorithm.

21. Proof for Explanatory Case

- The quantity $K(x) \stackrel{\text{def}}{=} \min\{\text{len}(p) : p \text{ generates } x\}$ is well known in theoretical computer science.
- It was invented by the famous statistician A. N. Kolmogorov and it is known as *Kolmogorov complexity*.
- One of the results that Kolmogorov proved is that no algorithm is possible for computing $K(x)$.
- This immediately implies our result: indeed,
  - if it was possible to produce, for each data x, the best explanatory model p_0,
  - then we would be able to compute its length $\text{len}(p_0)$ which is exactly $K(x)$,
  - and $K(x)$ is not computable.

22. Discussion

- Kolmogorov complexity was originally introduced for a different purpose.
- It was invented to separate random from non-random sequences.
- In the traditional statistics, the very idea that some sequences are random and some are not was taboo.
- One could only talk about probabilities of different sequences.
- However, intuitively, everyone understands that:
  - while a sequence of bits generated by flipping a coin many times is random,
  - a sequence like 010101...01 in which 01 is repeated million times is clearly not random.
- How can we formally explain this intuitive difference?

23. Discussion (cont-d)

- A sequence 0101...01 is not random because it can be generated by a short program: repeat 01 many times.
- Thus, the shortest possible length $K(x)$ of a program generating x is much smaller than $\text{len}(x)$:

$$K(x) \ll \text{len}(x).$$

- On the other hand, if a sequence is truly random, there is no dependency between different bits.
- So the only way to print this sequence is to literally print the whole sequence bit by bit: $K(x) \approx \text{len}(x)$.
- So, Kolmogorov defined a binary sequence x as *random* if $K(x) \geq \text{len}(x) - c_0$, for some constant c_0.

24. Proof that Kolmogorov Complexity Is Not Computable

- The main idea behind this proof comes from the following *Barry's paradox*.
- Some English expressions describe numbers; e.g.:
  - "twelve" means 12,
  - "million" means 1000000, and
  - "the smallest prime number above 100" means 101.
- There are finitely many words in the English language.
- So there are finitely many combinations of less than twenty words.
- Thus, there are finitely many numbers which can be described by such combinations.
- Hence, there are numbers which cannot be described by such combinations.

25. Proof that $K(x)$ Is Not Computable (cont-d)

- Let n_0 denote the smallest of such numbers.
- Therefore, n_0 is "the smallest number that cannot be describe in fewer than twenty words".
- But this description of the number n_0 consists of 12 words – less than 20.
- So n_0 *can* be described by using fewer than twenty words – a clear paradox.
- This paradox is caused by the imprecision of natural language.
- However, if we replace "described" by "computed", we get a proof that $K(x)$ is not computable.
- Indeed, let us assume that $K(x)$ is computable, and let L be the length of the program that computes $K(x)$.

26. Proof that $K(x)$ Is Not Computable (cont-d)

- Binary sequences can be interpreted as binary integers, so we can talk about the smallest of them.
- Then, the following program computes the smallest sequence x_0 for which $K(x) \geq 3L$.
- We try all possible binary sequences of lengths 1, 2, etc., until we find the first x for which $K(x) \geq 3L$:

$$\text{int x} = \ 0; \ \text{while}(\text{K}(\text{x}) < 3 * \text{L})\{\text{x} + +; \}$$

- This program adds just two short lines to the length-L program for computing $K(x)$.
- Thus, its length is $\approx L \ll 3L$, so $K(x_0) \ll 3L$.
- On the other hand, we defined x_0 as the smallest number for which $K(x) \geq 3L$.
- So we have $K(x_0) \geq 3L$ – a contradiction.

27. Acknowledgments

- This work was supported:
  - by the Center of Excellence in Econometrics, Faculty of Economics, Chiang Mai University;
  - by the Department of Mathematics, Chiang Mai University;
  - by the US National Science Foundation via grant HRD-1242122 (Cyber-ShARE Center of Excellence).
- The authors are greatly thankful to Professors Hung T. Nguyen and Galit Shmueli for valuable discussions.