Algorithmic Need for Subcopulas

Thach Ngoc Nguyen[1], Olga Kosheleva[2],
Vladik Kreinovich[2], and Hoang Phuong Nguyen[3]

[1]Banking University of Ho Chi Minh, Vietnam,
Thachnn@buh.edu.vn
[2]University of Texas at El Paso, El Paso, Texas 79968, USA,
olgak@utep.edu, vladik@utep.edu
[3]Division Informatics, Thang Long University, Hanoi, Vietnam,
nhphuong2008@gmail.com

1. Copulas: a Brief Reminder

- There are many ways to describe a probability distribution of a random variable:
  - we can use its probability density function (pdf),
  - we can use its moments,
  - its cumulative distribution function (cdf), etc.
- Most of these types of descriptions are not always applicable:
  - for a discrete distribution, pdf is not defined,
  - for a distribution with heavy tails, moments are sometimes infinite, etc.
- Out of the known representations, the representation as a cdf is the most universal.
- It does not seem to have limitations.

2. Copulas (cont-d)

- In econometrics, one can encounter:
  - discrete distributions (for which no pdf is known),
  - heavy-tailed distributions (for which moments are infinite), etc.
- It is reasonable to use a cdf $F_X(x) = \text{Prob}(X \le x)$ to describe a random variable X.
- Similarly, to describe a joint distribution of two random variables (X, Y), it is reasonable to use a joint cdf

$$F_{XY}(x, y) = \text{Prob}(X \le x \,\&\, Y \le y).$$

3. Copulas (cont-d)

- When random variables X and Y are independent, we have $F_{XY}(x, y) = F_X(x) \cdot F_Y(y)$.
- In general, the dependence may be more complicated.
- It is reasonable to describe this dependence by a function $C(u, v)$ for which $F_{XY}(x, y) = C(F_X(x), F_Y(y))$.
- A function with this property is known as a *copula*.
- Copulas have been successful used in many application areas, in particular, in econometrics.

4. Existence and Uniqueness of Copulas

- It has been proven that such a copula always exists.
- A copula $C(u, v)$ is itself a 2-D cdf on the unit square.
- When the distributions of X and Y are continuous, the copula is uniquely determined.
- Indeed, in this case, $F_X(x)$ continuously depends on x and thus, attains all possible values between 0 and 1.
- So, to find $C(u, v)$, it is sufficient to find the values x and y for which $F_X(x) = u$ and $F_Y(y) = v$.
- Then $F_{XY}(x, y)$ will give is the desired value of $C(u, v)$.
- However, if X – is discrete, then the value $F_X(x)$ jumps.
- Thus, for some intermediate values u, we do not have values x for which $F_X(x) = u$.
- Then, the copula is *not* uniquely determined: we can have different values $C(u, v)$ for jumped-over u.

5. Subcopulas: Reminder

- While the copula is not always unique, there is a variant of this notion which is always unique.
- This variant is known as a *subcopula*.
- In precise terms, a subcopula $C(u, v)$ is also defined by the formula $F_{XY}(x, y) = C(F_X(x), F_Y(y))$.
- The difference is that a subcopula $C(u, v)$ is only defined for the values $u = F_X(x)$ and $v = F_Y(y)$.
- Subcopulas have also been successfully used in econometrics.

6. Main Question: Do We Need Subcopulas?

- From the purely mathematical viewpoint, it may seem that do not need subcopulas.
- Indeed, every subcopula can be, in principle, extended to a copula.
- However, many researchers use subcopulas.
- This seem to indicate that subcopulas may not be easy to extend to copulas.
- In this talk, we prove that, in general, it is not algorithmically possible to always construct a copula.
- In other words, we prove that, from the algorithmic viewpoint, subcopulas are indeed needed.

7. What Is Computable: Main Definitions

- We wan to analyze when a copula is computable and when it is not.
- Let us recall the main definitions of computability.
- A real number x is *computable* if we can compute it with any given accuracy.
- In other words, a number is computable if there exists an algorithm that:
  - given an integer n (describing the accuracy),
  - returns rational r_n for which $|x - r_n| \le 2^{-n}$.
- Intuitively, a function $f(x)$ is computable if there is an algorithm that, given x, returns the value $f(x)$.
- So, for any desired accuracy n, we can compute a rational number r_n for which $|f(x) - r_n| \le 2^{-n}$.

8. What Is Computable (cont-d)

- In this computation, the program can pick some integer m and ask for an 2^{-m}-approximation to the input.
- Similarly, a function $f(x, y)$ of two variables is called computable if:
  - given x and y,
  - it can compute the value $f(x, y)$ with any given accuracy.
- In the process of computations, this program can ask for a 2^{-m}-approximation to x and to y.
- These definitions describe the usual understanding of computability.
- So, not surprisingly, all usual computable functions are computable in this sense as well.

9. What Is Not Computable

- What is not computable in this sense are discontinuous functions such as sign(x) which is equal:
  - to -1 when $x < 0$,
  - to 0 when $x = 0$, and
  - to 1 when $x > 0$.
- Indeed, if this function was computable, then we would be able to check whether $x = 0$ for a computable real number x.
- It is known that such checking is not algorithmically possible.
- Indeed, the possibility of such checking contradicts to the known result that it is not possible,
  - given a program,
  - to check whether this program halts or not.

10. What Is Not Computable (cont-d)

- Indeed, based on each program, we can form a sequence r_n each element of which is:
  - equal to 2^{-n} if the program did not yet halt by time n and
  - equal to 2^{-t} if it halted at time $t \le n$.
- If the program does not halt, this sequence describes the computable real number $x = 0$.
- If the program halts at time t, this sequence describes the computable real number $x = 2^{-t} > 0$.
- Thus, if we could check whether $x = 0$, we would be able to check whether a program halts.
- And this is known to be not algorithmically possible.

11. What Does It Mean for the cdf to Be Computable

- In real life, when we say that we have a random variable, it means that:
  - we have a potentially infinite sequence of observations
  - which follow the corresponding distribution.
- Based on these observations, for each computable real number x, we would like to compute the value $F(x)$.
- The value $F(x)$ is the probability that the value of a random variable X is $\leq x$.
- A natural practical way to estimate a probability based on a finite sample is to find the corr. frequency.

12. Computable cdf (cont-d)

- Thus, to estimate $F(x)$, a natural idea is:
  - to take n observations $X_1, \ldots, X_n$,
  - to find out how many of them are $\leq x$, and then
  - to compute the desired frequency by dividing the result of the counting by n.
- The frequency f is, in general, different from the probability p.
- For large n, the difference $f - p$ is approximately normally distributed, with 0 mean and standard deviation

$$\sigma = \sqrt{\frac{p \cdot (1-p)}{n}} \leq \frac{0.5}{\sqrt{n}}.$$

- From the practical viewpoint, any deviation larger than 6 sigma has a probability of less than 10^{-8}.
- It is, thus, usually considered practically impossible.

13. Computable cdf (cont-d)

- If you do not view 6 sigma as impossible, take 20 sigma.
- One can always come up with a probability so small that it is practically impossible.
- Thus, if we select n so large that $6\sigma \le 6 \cdot \dfrac{0.5}{\sqrt{n}} \le \varepsilon$, then $|f - F(x)| \le \varepsilon$, i.e., $F(x) - \varepsilon \le f \le F(x) + \varepsilon$.
- We also need to take into account that the values X_i can only be measured with a certain accuracy δ.
- We do not get the actual values.
- We get the results $\widetilde{X}_i$ of their measurement which are δ-close to X_i.
- If $\widetilde{X}_i \le x$, we cannot conclude that $X_i \le x$, we can only conclude that $X_i \le x + \delta$.

14. Computable cdf (cont-d)

- Similarly, if $\widetilde{X}_i > x$, we cannot conclude that $X_i > x$, we can only conclude that $X_i > x - \delta$.
- Thus, the only thing that we can guarantee for the observed frequency f is that

$$F(x - \delta) - \varepsilon \le f \le F(x + \delta) + \varepsilon.$$

- This is how a computable cdf is defined: that,
  - given every a computable number x and rational numbers $\varepsilon > 0$ and $\delta > 0$,
  - we can efficiently find a rational number f that satisfies the above inequality.
- A similar inequality defines a computable 2-D cdf:

$$F(x - \delta, y - \delta) - \varepsilon \le f \le F(x + \delta, y + \delta) + \varepsilon.$$

- Note that a cdf can be discontinuous.

15. Computable cdf (cont-d)

- For example, if we have a random variable that is equal to 0 with probability 1, then:
  - $F(x) = 0$ for $x < 0$ and
  - $F(x) = 1$ for $x \geq 0$.
- We already know that such a function cannot be computable.
- So a computable cdf is not necessarily a computable function.
- First result: *if the computable cdf is continuous, it is a computable function.*

16. Main Results

- *There exists an algorithm that,*
  - *given a continuous computable cdf* $F_{XY}(x, y)$,
  - *generates the corresponding copula,*
  - *i.e., generates the corresponding computable cdf* $C(u, v)$.
- *No general algorithm is possible that:*
  - *given a computable cdf* $F_{XY}(x, y)$,
  - *would generate the corresponding copula,*
  - *i.e., that would generates a corresponding computable cdf* $C(u, v)$.
- This result proves that it is, in general, not possible to always generate a copula.
- Thus, from the algorithmic viewpoint, subcopulas are indeed needed.

17. Acknowledgments

- This work was supported by the US National Science Foundation via grant HRD-1242122 (Cyber-ShARE).
- The authors are thankful to Professor Hung T. Nguyen for valuable discussions.

18. Proof that Continuous Computable cdf Is a Computable Function

- The inequalities defining computable cdf can be rewritten as $f_{\delta,\varepsilon}(x-\delta)-\varepsilon \le F(x) \le f_{\delta,\varepsilon}(x+\delta)$.
- Here $f_{\delta,\varepsilon}(x)$ means a frequency estimated:
  - by comparing the measured values $\widetilde{X}_i$ (measured with accuracy δ) with the value x,
  - based on a sample large enough to guarantee the accuracy ε.
- Also, we have $F(x-2\delta)-\varepsilon \le f_{\delta,\varepsilon}(x-\delta)$ and $f_{\delta,\varepsilon}(x+\delta) \le F(x+2\delta)+\varepsilon$. Thus:

$$F(x-2\delta)-2\varepsilon \le f_{\delta,\varepsilon}(x-\delta)-\varepsilon \le F(x) \le$$

$$f_{\delta,\varepsilon}(x+\delta)+\varepsilon \le F(x+2\delta)+2\varepsilon.$$

- The cdf $F(x)$ is a continuous function.

19. First Proof (cont-d)

- So, for each x, the difference $F(x + 2\delta) - F(x - 2\delta)$ tends to 0 as δ decreases.
- Thus, the difference between the values $F(x+2\delta)+2\varepsilon$ and $F(x-2\delta)-2\varepsilon$ also tends to 0 as $\delta \to 0$ and $\varepsilon \to 0$.
- So, if we take $\delta = \varepsilon = 2^{-k}$ for $k = 1, 2, \ldots$, we will eventually find k for which this difference is $\leq 2^{-n}$.
- In this case, the difference between the inner bounds $f_{\delta,\varepsilon}(x + \delta) + \varepsilon$ and $f_{\delta,\varepsilon}(x - \delta) - \varepsilon$ is also $\leq 2^{-n}$.
- So, each of these bounds can be used as the desired 2^{-n}-approximation to $F(x)$.
- Thus, to compute $F(x)$ with accuracy 2^{-n}, it is sufficient to compute, for $k = 1, 2, \ldots$,
  - values $\varepsilon = \delta = 2^{-k}$ and then
  - values $f_{\delta,\varepsilon}(x + \delta) + \varepsilon$ and $f_{\delta,\varepsilon}(x - \delta) - \varepsilon$.

20. First Proof (cont-d)

- We continue these computations for larger and larger k until $|(f_{\delta,\varepsilon}(x+\delta)+\varepsilon)-(f_{\delta,\varepsilon}(x-\delta)-\varepsilon)| \le 2^{-n}$.
- Once this condition is satisfied, we return $f_{\delta,\varepsilon}(x+\delta)+\varepsilon$ as the desired 2^{-n}-approximation to $F(x)$.
- In the 2-D case, we can use a similar algorithm.

21. Proof That in the Continuous Case, Copula Is Computable

- This proof follows the idea of finding the copula for a continuous cdf.
- Suppose that we are given two computable numbers $u, v \in [0, 1]$.
- We want to find the desired approximation to $C(u, v)$.
- To do that, we first find x for which $F_X(x)$ is δ-close to u.
- This value can be found as follows.
- First, we pick any x_0 and compute $F_X(x_0)$ with accuracy δ.
- We can do it, since, for continuous distributions, the cdf is a computable function.
- If we get a value which is δ-close to u, we are done.

22. Second Proof (cont-d)

- If the approximate value $F_X(x_0)$ is larger than u, we take $x_0 - 1$, $x_0 - 2$, etc.
- We stop when we find a value x_- for which $F_X(x_-) < u$.
- Similarly, if the approximate value $F_X(x_0)$ is smaller than u, we take $x_0 + 1$, $x_0 + 2$, etc.
- We stop when we find a value x_+ for which $F_X(x_+) > u$.
- In both cases, we have an interval $[x_-, x_+]$ for which $F(x_-) < u < F(x_+)$.
- Now, we can use bisection to find the desired x.
- Namely, we take a midpoint x_m of the interval.
- Then, if $|F_X(m) - u| \le \delta$, we are done.
- If this ideal inequality is not satisfied, then we have either $F_X(x_m) < u$ or $F_X(x_m) > u$.

23. Second Proof (cont-d)

- In the first case, we know that the desired value x is in the half-size interval $[x_m, x_+]$.
- In the second case, we know that the desired value x is in the half-size interval $[x_-, x_m]$.
- In both cases, we get a new half-size interval.
- To the new interval, we apply the same procedure until we get the desired x.
- Similarly, we can compute y for which $F_Y(y) \approx v$.
- Now, we can take the approximation to $F_{XY}(x, y)$ as the desired approximation to $C(u, v)$.

24. Proof That in Non-Continuous Case, Copulas Are, in General, Not Computable

- For each real number a s. t. $|a| \leq 0.5$, we can form the following distribution $F_a(x, y)$ on the unit square.
- It is uniformly distributed on a straight line segment $y = 0.5 + \text{sign}(a) \cdot (x - 0.5)$ corr. to $x \in [0.5 - |a|, x + |a|]$.
- Thus, when $a > 0$, we take $y = x$, and when $a < 0$, we take $y = 1 - x$.
- One can easily check that $F_a(x, y)$ is indeed a computable cdf.
- Note that it is *not* always a computable function.
- For example, for $a = 0$ the whole probability distribution is concentrated at the point $(0.5, 0.5)$.
- For each a, the marginal distribution $F_X(x)$ is uniformly distributed on the interval $[0.5 - |a|, 0.5 + |a|]$.

25. Third Proof (cont-d)

- Thus, for the values x from this interval, we have

$$F_X(x) = \frac{x - (0.5 - |a|)}{2|a|}.$$

- So for any $u \in [0, 1]$, to get $F_X(x) = u$, we must take $x = 0.5 - |a| + 2u \cdot |a| = 0.5 - (1 - 2u) \cdot |a|$.
- Similarly, to get the value y for which $F_Y(y) = v$, we should take $y = 0.5 - (1 - 2v) \cdot |a|$.
- For $u, v \le 0.5$, we get $x \le 0.5$ and $y \le 0.5$.
- For $u = v = 0.25$, we take $x = y = 0.5 - 0.5|a|$.
- For $u = v = 0.5$, we take $x = y = 0.5$.
- The distribution is symmetric.
- So, when $u = v$, we have the same values $x = y$ for which $F_X(x) = u$ and $F_Y(y) = u$.

26. Third Proof (cont-d)

- When $a > 0$, $X = Y$, so

$$C(u, u) = F_{XY}(x, x) = \text{Prob}(X \le x \,\&\, Y \le x) =$$

$$\text{Prob}(X \le x) = u, \text{ i.e., } C(u, u) = u.$$

- In particular,

$$C(0.25, 0.25) = 0.25 \text{ and } C(0.5, 0.5) = 0.5.$$

- When $a < 0$, then $Y = 1 - X$.
- Thus, when $X \le 0.5$, we have $Y \ge 0.5$.
- So when $u, v \le 0.5$, we cannot have both $X \le u$ and $Y \le v$, and thus, we get

$$C(u, u) = F_{XY}(x, x) = \text{Prob}(X \le x \,\&\, Y \le x) = 0.$$

- In particular, we have $C(0.25, 0.25) = C(0.5, 0.5) = 0$.

27. Third Proof (cont-d)

- Let us assume that it is possible, given a computable real number a, to compute a computable cdf $C(u, v)$.
- Then, by definition of a computable cdf, we would be able to compute:
  - given a,
  - the value $f_{\delta,\varepsilon}(x)$ corresponding to $x = 0.375$, $\delta = 0.125$ and $\varepsilon = 0.1$,
- So, we can compute $f \stackrel{\text{def}}{=} f_{0.125,0.1}(0.375)$ for which

$$C(x-\delta, x-\delta) - \varepsilon \le f \le C(x+\delta, x+\delta) + \varepsilon, \text{ i.e.,}$$

$$C(0.25, 0.25) - 0.1 \le f \le C(0.5, 0.5) + 0.1.$$

- When $a < 0$, we have $C(0.5, 0.5) = 0$, hence $f \le 0.1$ and therefore $f < 0.125$.
- When $a > 0$, then $C(0.25, 0.25) = 0.25$, hence $f \ge 0.15$ and therefore $f > 0.125$.

28. Third Proof (cont-d)

- So, by comparing f with 0.125, we will able to check whether $a > 0$ or $a < 0$.
- And this is known to be algorithmically impossible.
- This contradiction shows that it is indeed not possible to have an algorithm that always computes the copula.