McFadden's Discrete Choice and Softmax under Interval (and Other) Uncertainty: Revisited

Bartlomiej Jacek Kubica[1], Olga Kosheleva[2], and Vladik Kreinovich[2]

[1]Department of Applied Informatics, Warsaw University of Life Sciences
ul. Nowoursynowska 159, 02-776 Warsaw, Poland
bartlomiej.jacek.kubica@gmail.com

[2]Olga Kosheleva and Vladik Kreinovich
University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, olgak@utep.edu, vladik@utep.edu

1. What is McFadden's discrete choice: a brief reminder

- According to the decision theory, preferences of a rational decision maker are described by a special function – called *utility* u.
- In this description, the decision makes always selects the alternative i with the largest possible value of utility u_i.
- In particular, this means that decisions of a rational decision maker should be deterministic – in the sense that:
  - if we offer, to the decision maker, the same choice some time in the future,
  - he/she will make the exact same decision.
- In practice, however, people's decisions are not deterministic:
  - in many cases, we select the alternative with the largest utility,
  - but sometimes, we select an alternative with somewhat smaller utility.

2. What is McFadden's discrete choice (cont-d)

- In general:
  - alternatives with higher utility are selected more frequently, while
  - alternatives with lower utility are selected less frequently.
- A study of this phenomena led Daniel L. McFadden to the following empirical formula that:
  - uses the utilities $u_1, \ldots, u_n$ of different alternatives
  - to predict the probability p_i that this alternative will be selected:

$$p_i = \frac{\exp(k \cdot u_i)}{\sum_{j=1}^{n} \exp(k \cdot u_j)} \text{ for some constant } k > 0.$$

- For this discovery, Professor McFadden was awarded the Nobel Prize in Economics.

3. What is softwax: a brief reminder

- One of the main applications of deep learning is the classification problem, when:
  - we are given several classes of objects, and
  - we need to decide to which of the classes the given object belongs.
- For example:
  - in autonomous driving systems,
  - we need to be able to tell whether an object in front of our car is a person, a bicycle, or another car.
- In the vast majority of cases, a trained neural network provides the correct answer, but sometimes the neural network errs.
- It is therefore desirable to make sure that:
  - the system not only provide an answer, but
  - that it should also provide us with the probability that this answer is correct.

4. What is softwax: a brief reminder (cont-d)

- This way, if this probability is low, we can perform additional measurements and observations.
- In a nutshell, the neural networks classify the objects as follows.
- For each class i of objects, a sub-network is trained to recognize objects of this class.
- For each object, each of these sub-networks produces a degree u_i to which:
  - according to this sub-network,
  - the given object belongs to the i-th class.
- If we want a single answer, then, of course, we select the class i for which the corresponding degree is the largest.
- In addition to this selection, we also want to estimate the probabilities.

5. What is softwax: a brief reminder (cont-d)

- In other words:
  - based on the n degrees $u_1, \ldots, u_n$,
  - we need to estimates the probabilities $p_1, \ldots, p_n$ that the given object belongs to the corresponding class.
- Interestingly, an empirically reasonable way to estimate these probabilities is to use the following formula – same as McFadden's:

$$p_i = \frac{\exp(k \cdot u_i)}{\sum\limits_{j=1}^{n} \exp(k \cdot u_j)}.$$

- This formula is known as *softmax*.
- Indeed, instead of the "hard" maximum, when we simply select the class i with the largest degree u_i, we have "soft" maximum, when:
  - we select the most probable class with higher probability, but
  - we also, with some non-zero probability, select other classes.

6. Need to consider interval uncertainty

- McFadden's formula assumes that we know the exact utility values u_i.
- In practice, we only get these values with some uncertainty.
- For example, we may only know the range $[\underline{u}_i, \overline{u}_i]$ of possible values of u_i.
- For different values u_i from the corresponding intervals, we get, in general, different values of the probabilities p_i.
- A natural question is: what is the resulting range $[\underline{p}_i, \overline{p}_i]$ of possible values of each probability p_i?
- A similar problem emerges in the softmax situation.
- The values u_i are computed based on the results of measuring the corresponding object.

7. Need to consider interval uncertainty (cont-d)

- Measurement are never absolutely accurate:
  - the result $\widetilde{x}$ of measuring a quantity x is, in general, somewhat different from
  - the actual (unknown) value of the corresponding quantity.
- In many practical situations:
  - the only information that we have about the measurement error $\Delta x \stackrel{\text{def}}{=} \widetilde{x} - x$ is
  - the upper bound Δ on the error's absolute value: $|\Delta x| \le \Delta$.
- In this case, after the measurement, the only information that we get about the actual value x is that this value belongs to the interval

$$[\widetilde{x} - \Delta, \widetilde{x} + \Delta].$$

- For different values x from the corresponding intervals, we get, in general, different values u_i.

8. Need to consider interval uncertainty (cont-d)

- As a result, for each i, we get the interval $[\underline{u}_i, \overline{u}_i]$ of possible values u_i corresponding to different values of x.
- Computing this interval is an important particular case of *interval computations*.
- For different values u_i from the corresponding intervals, we get, in general, different values of the probability p_i.
- It is therefore desirable to find the range $[\underline{p}_i, \overline{p}_i]$ of possible values of each probability p_i.
- This is useful in practice: for example:
  - it is one thing to say that an estimate of the probability that the classification is correct is 80%, and
  - it is a different thing to say that this probability is somewhere 70% and 90%.

9. Resulting problem

- From the computational viewpoint, in both cases, we face the same problem:
  - *we know* the value k, and we know intervals $[\underline{u}_i, \overline{u}_i]$ of possible values of u_i;
  - *we want to find* the range $[\underline{p}_i, \overline{p}_i]$ of possible values of the above expression .

10. At first glance, this problem sounds very complicated but, as we show, it is not

- In general, problems of interval computations are NP-hard.
- This means, crudely speaking, that:
  - unless P = NP (which most scientists believe not to be the case),
  - no feasible algorithm can solve all particular cases of this problem.
- Interval computation problems are even NP-hard if we want to compute the range of a quadratic function.
- The expression for p_i has exponential functions and division.

11. At first glance, this problem sounds very complicated but, as we show, it is not (cont-d)

- These operations are much more complex than addition and multiplication needed to compute a quadratic expression.
- So, it seems reasonable to expect that computing the range of p_i is also computationally complicated.
- In this talk, we show, however, that the above problem is quite feasible.
- Moreover, it can be solved in linear time.
- We also show that this feasibility holds for reasonable generalizations of the McFadden's formula and of interval uncertainty.

12. Why we say "Revisited"

- We use the word Revisited, since we dealt with softmax and discrete choice under interval uncertainty in our previous paper.
- The difference is as follows.
- In that paper, we dealt with a different problem:

how to select the most reasonable single value of each probability p_i under interval uncertainty.

- In contrast, in this talk, we are interested in finding the whole *range* of possible probability values.

13. Preliminary analysis of the problem

- To simplify our analysis, let us divide both the numerator and the denominator of McFadden's formula by its numerator.
- As a result, we get the following expression:

$$p_i = \frac{1}{1 + \sum_{j \neq i} \frac{\exp(k \cdot u_j)}{\exp(k \cdot u_i)}}.$$

- Here, each fraction $\dfrac{\exp(k \cdot u_j)}{\exp(k \cdot u_i)}$ increases with u_j $(j \neq i)$ and decreases with u_i.
- Thus, the denominator – which is the sum of these terms – also increases with each u_j $(j \neq i)$ and decreases with u_i.

14. Preliminary analysis of the problem (cont-d)

- Since the function $1/x$ is decreasing, the probability p_i – which is equal to 1 over denominator:
  - decreases with u_j ($j \neq i$) and
  - increases with u_i.
- So, the probability p_i is the largest:
  - when u_i is the largest possible and other values u_j are the smallest possible,
  - i.e., when $u_i = \overline{u}_i$ and $u_j = \underline{u}_j$ for all $j \neq i$.
- The probability p_i is the smallest:
  - when u_i is the smallest possible and other values u_j are the largest possible,
  - i.e., when $u_i = \underline{u}_i$ and $u_j = \overline{u}_j$ for all $j \neq i$.

15. Preliminary analysis of the problem (cont-d)

- Thus, we arrive at the following formulas:

$$\underline{p}_i = \frac{\exp(k \cdot \underline{u}_i)}{\exp(k \cdot \underline{u}_i) + \sum\limits_{j \neq i} \exp(k \cdot \overline{u}_j)};$$

$$\overline{p}_i = \frac{\exp(k \cdot \overline{u}_i)}{\exp(k \cdot \overline{u}_i) + \sum\limits_{j \neq i} \exp(k \cdot \underline{u}_j)}.$$

16. What if we follow these formulas directly?

- According to the above formulas, to compute each of the $2n$ bounds $\underline{p}_i$ and $\overline{p}_i$, we need a linear number of steps $C \cdot n$ for some constant C.
- Thus, overall, we need $2n \cdot C \cdot n = O(n^2)$, i.e., quadratic time.
- Can we compute all the probabilities faster?
- Yes, it we take into account that, e.g., for the first formula:
  - if we add and subtract the term $\exp(k \cdot \overline{u}_i)$ to its denominator – thus not changing the value of the denominator,
  - we get the form

$$\exp(k \cdot \underline{u}_i) - \exp(k \cdot \overline{u}_i) + \sum_{j=1}^{n} \exp(k \cdot \overline{u}_j).$$

17. What if we follow these formulas directly (cont-d)

- Similarly, if we add and subtract the term $\exp(k \cdot \underline{u}_i)$ to the denominator of the second formula, we get the following expression:

$$\exp(k \cdot \overline{u}_i) - \exp(k \cdot \underline{u}_i) + \sum_{j=1}^{n} \exp(k \cdot \underline{u}_j).$$

- The n-term sums in these expressions are the same for all i, so they can be computed only once.
- Thus, we arrive at the following linear-time algorithm.

18. Linear-time algorithm for computing the ranges $[\underline{p}_i, \overline{p}_i]$

- We are given the values $\underline{u}_i$ and $\overline{p}_i$. Based on these values:
- First, we compute the values $\exp(k\cdot\underline{u}_i)$, $\exp(k\cdot\overline{u}_i)$, and the differences

$$m_i \stackrel{\text{def}}{=} \exp(k\cdot\overline{u}_i) - \exp(k\cdot\underline{u}_i).$$

- Then, we compute the sums $\underline{s} = \sum\limits_{i=1}^{n} \exp(k\cdot\underline{u}_i)$ and $\overline{s} = \sum\limits_{i=1}^{n} \exp(k\cdot\overline{u}_i)$.
- After that, we compute the desired values

$$\underline{p}_i = \frac{\exp(k\cdot\underline{u}_i)}{\overline{s} - m_i}; \quad \overline{p}_i = \frac{\exp(k\cdot\overline{u}_i)}{\underline{s} + m_i}.$$

- One can easily check that this algorithm requires linear time.

19. Comment

- We cannot compute the desired bounds faster than in linear time.
- Indeed, we need to process all $2n$ inputs $\underline{u}_i$ and $\overline{u}_i$.
- Each elementary operation – arithmetic operation of an application of an elementary function like $\exp(x)$ – can process at most two values.
- Thus, to process all $2n$ inputs, we need at least $(2n)/2 = n$ computational steps.
- So, from the computational viewpoint, our algorithm is asymptotically optimal.

20. What if we only know the value k with interval uncertainty?

- We are taking into account that many values are known with uncertainty.
- So, it is reasonable to also consider the case when:
  - the value of the parameter k is also known with interval uncertainty, i.e.,
  - when we only know the interval $[\underline{k}, \overline{k}]$ of possible values k.
- We should look for the range of values p_i corresponding to all possible combinations of values u_i and k from the corresponding intervals.
- In classification problems, we are mostly interested in the probability p_i that the generated answer is correct.
- This probability that corresponds to the largest value of u_i.

21. What if we only know k with interval uncertainty (cont-d)

- For this value i, we can reformulate the softmax expression in the following equivalent form:

$$p_i = \frac{1}{1 + \sum\limits_{j \neq i} \exp(k \cdot (u_j - u_i))}.$$

- Here, $u_i \geq u_j$ for all j, so $u_j - u_i \leq 0$.
- Thus, $\exp(k \cdot (u_j - u_i))$ decreases with k, so the sum of these terms also decreases with k, and so it the denominator of the expression.
- So, the fraction increases with k.
- So, to compute the lower endpoint $\underline{p}_i$, it is sufficient to consider the smallest possible value of k, namely $k = \underline{k}$.
- To compute the upper endpoint $\underline{p}_i$, it is sufficient to consider the largest possible value of k, namely $k = \overline{k}$.

22. What if we only know k with interval uncertainty (cont-d)

- So, we get the following formulas:

$$\underline{p}_i = \frac{\exp(\underline{k} \cdot \underline{u}_i)}{\exp(\underline{k} \cdot \underline{u}_i) + \sum\limits_{j \ne i} \exp(\underline{k} \cdot \overline{u}_j)}; \quad \overline{p}_i = \frac{\exp(\overline{k} \cdot \overline{u}_i)}{\exp(\overline{k} \cdot \overline{u}_i) + \sum\limits_{j \ne i} \exp(\overline{k} \cdot \underline{u}_j)}.$$

- We are only interested in computing the values $\underline{p}_i$ and $\overline{p}_i$ for one class i.
- So, we can simply follow these formulas and get a linear-time algorithm.

23. Comment

- The possibility to have a linear-time algorithm depends on the fact that:
  - for the class i with the largest value u_i,
  - the probability p_i monotonically depends on k – namely, it increases with k.
- One can similarly show that for the smallest value u_i, we also have a monotonic dependence – namely, p_i decreases with k.
- However, for intermediate values u_i, the dependence on k is not necessarily monotonic, as the following simple example shows.
- Let us take $u_1 = \ln(1) = 0 > u_2 = \ln(0.6) > u_3 = \ln(0.1)$.
- Then, for $k = 0$, we get $\exp(k \cdot u_i) = 1$ for all i, so

$$p_2(0) = \frac{1}{1+1+1} = \frac{1}{3} = 0.33\ldots$$

24. Comment (cont-d)

- For $k = 1$, we get $\exp(k \cdot u_1) = 1$, $\exp(k \cdot u_2) = \exp(\ln(0.6)) = 0.6$, and $\exp(k \cdot u_3) = \exp(\ln(0.1)) = 0.1$, so
$$p_2(1) = \frac{0.6}{1 + 0.1 + 0.6} = \frac{0.6}{1.7} = 0.35\ldots$$
- For $k \to \infty$, we get $\exp(k \cdot u_1) = 1$ while $\exp(k \cdot u_2)$ and $\exp(k \cdot u_3)$ tend to 0.
- So in the limit, we get
$$p_2(\infty) = \frac{0}{1 + 0 + 0} = 0.$$
- So here $k = 0 < k = 1 < k = \infty$, but for the corresponding values of p_2, we do not get monotonicity:
$$p_2(0) = 0.33\ldots < p_2(1) = 0.35\ldots > p_2(\infty) = 0.$$
- Thus, whether we can feasibly compute the range of the other probabilities p_i, is still an open question.

25. What if we use generalizations of the softmax formulas?

- Softmax are empirical, they work well but not always perfectly.
- To have a better fit with the data, researchers proposed more general formulas, of the type

$$p_i = \frac{f(u_i)}{\sum\limits_{j=1}^{n} f(u_j)}, \text{ for some non-negative increasing function } f(u).$$

- All our results can be naturally extended to this more general case.
- Namely, in this case, we have

$$\underline{p}_i = \frac{f(\underline{u}_i)}{f(\underline{u}_i) + \sum\limits_{j \neq i} f(\overline{u}_j)};$$

$$\overline{p}_i = \frac{f(\overline{u}_i)}{f(\overline{u}_i) + \sum\limits_{j \neq i} f(\underline{u}_j)}.$$

26. Generalizations of the softmax formulas (cont-d)

- We also have the following linear-time algorithm for computing $\underline{p}_i$ and $\overline{p}_i$:
- First, we compute the values $f(\underline{u}_i)$, $f(\overline{u}_i)$, and the differences
$$m_i \stackrel{\text{def}}{=} f(\overline{u}_i) - f(\underline{u}_i).$$
- Then, we compute the sums $\underline{s} = \sum\limits_{i=1}^{n} f(\underline{u}_i)$ and $\overline{s} = \sum\limits_{i=1}^{n} f(\overline{u}_i)$.
- After that, we compute the desired values
$$\underline{p}_i = \frac{f(\underline{u}_i)}{\overline{s} - m_i}; \quad \overline{p}_i = \frac{f(\overline{u}_i)}{\underline{s} + m_i}.$$

27. What if we consider fuzzy uncertainty instead of interval uncertainty?

- In the previous text, we considered situations when the approximate values $\widetilde{x}$ come from measurements.
- There is another possibility: that the approximate values come from an expert estimate.
- Experts usually describe the accuracy of their estimates:
  - not in terms of precise bounds,
  - but rather by using imprecise ("fuzzy") words from natural language.
- For example, they can say that "the value is approximately 1 with accuracy about 0.1."
- To describe such information in precise terms, Lotfi Zadeh came up with an idea that he called *fuzzy logic*.

28. What if we consider fuzzy uncertainty instead of interval uncertainty (cont-d)

- Specifically, for each imprecise property like "approximately 1 with accuracy about 0.1," he suggested to assign:
  - to each real number x,
  - the degree $\mu(x)$ – from the interval $[0, 1]$ – the degree to which this number x satisfies this property.
- Here, 1 means that the expert is absolutely sure that x satisfies this property.
- 0 means that the expert is absolutely sure that x does not satisfy this property.
- Intermediate values mean that the expert is somewhat sure.
- The function that assigns the degree $\mu(x)$ to each number x is known as the *membership function*, or, alternatively, as the *fuzzy set*.
- In this case, instead of intervals $[\underline{u}_i, \overline{u}_i]$, we have fuzzy sets $\mu_i(u_i)$.

29. What if we consider fuzzy uncertainty instead of interval uncertainty (cont-d)

- It is known that in general:
  - application of an algorithm $y = F(u_1, \ldots, u_n)$ to fuzzy inputs $\mu_i(u_i)$ – that should result in a fuzzy set $\mu(y)$
  - can be reduced to processing intervals if we use the following alternative representation of fuzzy sets.
- Namely, for each fuzzy set $\mu(x)$, for each $\alpha \in (0, 1]$, we can form an *α-cut* $\mathbf{x}(\alpha) \stackrel{\text{def}}{=} \{x : \mu(x) \geq \alpha\}$.
- For $\alpha = 0$, the α-cut is defined as $\overline{\{x : \mu(x) > 0\}}$, where $\overline{S}$ means the closure of the set S, i.e., the set S and all its limit points.
- Once we know all the α-cuts $\mathbf{x}(\alpha)$, we can reconstruct the membership function as $\mu(x) = \sup\{\alpha : x \in \mathbf{x}(\alpha)\}$.

30. What if we consider fuzzy uncertainty instead of interval uncertainty (cont-d)

- Then, it turns out that for each α, the α-cut $\mathbf{y}(\alpha)$ of y can be obtained by applying interval computations to the α-cuts $\mathbf{u}_i(\alpha)$:

$$\mathbf{y}(\alpha) = \{F(u_1, \ldots, u_n) : u_i \in \mathbf{u}_i(\alpha) \text{ for all } i\}.$$

- So, all we need to do is select, e.g., levels $\alpha = 0, 0.1, 0.2, \ldots, 0.9.1$.
- For each level, we apply the above algorithm to the α-cuts $\mathbf{u}_i(\alpha)$, and thus get the desired α-cuts $\mathbf{p}_i(\alpha)$ for the probabilities p_i.

31. Acknowledgments

This work was supported in part by:

- National Science Foundation grants 1623190, HRD-1834620, HRD-2034030, and EAR-2225395;
- AT&T Fellowship in Information Technology;
- program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and
- a grant from the Hungarian National Research, Development and Innovation Office (NRDI).