Minimum entropy algorithm for multinomial NPI

In this short paper I like to propose an algorithm to calculate the minimum entropy distribution for f–probability intervals obtained by the multinomial NPI.

I start out with a the underlying concepts and proofs why the algorithms reaches its goals. Later a schematic outline of the algorithm itself is presented. It is mainly a port of the ideas of the minimum entropy algorithm for ordinal NPI as in [1].

General ideas

While the maximum entropy distribution may be obtained by only one single distribution, this does not hold for the minimum entropy distribution. There are likely to be more than on distribution having a minimum entropy.

For each of those distributions it still holds that for each category the value lies within the probability interval obtained by the NPI. Accordingly to the approach when calculating the maximum entropy distribution, this algorithm starts with the lower interval probabilities and increases them until a probability distribution is reached, i.e. they sum up to 1.

Proofs

At first 2 lemmas concerning the mulitnomial NPI.

Lemma 1. For any observed categories c_i and c_j with $i \neq j$ and any either observed or unobserved category c_k the following holds in the multinomial NPI case:

1. $$l_i < l_j \Longleftrightarrow u_i < u_j$$
2. $$l_k = l_i \Longrightarrow u_k \leq u_i$$

Proof. From the multinomial NPI model, excluding the trivial case when all observations are of the same category, we obtain the following results for ...

... any observed category c_o the lower probability to $\frac{n_o-1}{n}$ and the according upper to $\frac{n_o+1}{n}$.

... any unobserved category c_u the lower probability to 0 and the according upper to $\frac{n_u+1}{n} = \frac{1}{n}$.

Thus we get for the first:

$$l_i < l_j \Longrightarrow \frac{n_i - 1}{n} < \frac{n_j - 1}{n} \Longrightarrow n_i < n_j \Longrightarrow \frac{n_i + 1}{n} < \frac{n_j + 1}{n} \Longrightarrow u_i < u_j.$$

If $n_i = n_j$ then the relations are of equality.

For the second statement we have equality in the lower probability if $n_i = n_k$, leading trivially to equality in the upper probabilities, but also when $n_i = 1$ and $n_k = 0$. For the latter we obtain for the lower probabilities $l_i = l_k = 0$, but with $n_i > n_k$ we get $u_i > u_k$. Both cases give the second statement. □

With Lemma 1 we guarantee that when ordering the categories according to their observations, the according lower and upper probabilities are ordered alongside. This means with the multinomial NPI no cross-over effect[1] may happen.
Let us define $H_1(x_1, x_2) := -\ln(x_1)x_1 - \ln(x_2)x_2$ as the contribution of x_1 and x_2 to the whole entropy H, and assume we need to assign a mass of m to either x_1 or x_2 or both.

Lemma 2. When assigning mass m to either x_1 or x_2 or both and $x_1 = x_2$ then we minimize the entropy when assigning m completely to one side.

Proof. As H and therefore H_1 are concave, me get for any c with $0 \leq c \leq m$:

$$H_1(x_1 + m - c, x_2 + c) \geq H_1(x_1 + m, x_2) = H_1(x_1, x_2 + m)$$

The latter equality only holds as $x_1 = x_2$. □

Lemma 3. When assigning mass m to either x_1 or x_2 or both and $x_1 > x_2$ then we get the minimal entropy when assigning m completely to the larger one, x_1.

Proof. As we learn from Lemma 2, it is optimal to assign the complete mass m completely to a category. Again with the concaveness of the entropy the following may be proven:

$$H_1(x_1 + m, x_2) \leq H_1(x_1, x_2 + m)$$

□

Lemma 4. Assigning mass to an unobserved category should be avoided as it increases the entropy.

Proof. This is easy to see from Lemma 3 when setting $x_2 = 0$. □

Lemma 5. When considering only two different categories c_i and c_j, with according upper and lower probability limits and number of observations, and a free mass of $\frac{2}{n}$ to assign, then assigning the mass to the category with larger number of observations minimizes the entropy gain. If $n_i = n_j$ then assigning $\frac{2}{n}$ to any of the categories leads to the same entropy.

Proof. Without loss of generality assume that the categories are placed next to each other on the wheel[2] and $n_i = n_j + \frac{\alpha}{n}$ with $\alpha \in \mathbb{N}$. Free mass of $\frac{2}{n}$ means two slices of mass $\frac{1}{n}$ each. As the categories are next to each other, one and only one slice (s_2) must be in between the according segments. Without loss of generality assume further the other slice (s_1) is to the left of both categories. So the configuration of the wheel for this special part may have the following shapes:

labels:	s_1	c_i	s_2	c_j	
masses:	$\frac{1}{n}$	l_i	$\frac{1}{n}$	l_j	

or

[1] cross-over in the sense that one may have $l_i < l_j$ and $u_i > u_j$

[2] If they actually aren't one may move them on the wheel as ordering of the categories is of no interest in the multinomial NPI model.

labels:	s_1	c_j	s_2	c_i
masses:	$\frac{1}{n}$	l_j	$\frac{1}{n}$	l_i

With those 2 configurations three different mass assignments are obtainable, according to Lemma 3 such that each $\frac{1}{n}$ is completely assigned to any of the categories.

	p_i	p_j	$\lvert p_i - p_j \rvert$
1. all mass to c_i	$u_i = l_i + \frac{2}{n} = l_j + \frac{\alpha}{n} + \frac{2}{n}$	l_j	$\frac{\alpha}{n} + \frac{2}{n}$
2. $\frac{1}{n}$ to each	$l_i + \frac{1}{n} = l_j + \frac{\alpha}{n} + \frac{1}{n}$	$l_j + \frac{1}{n}$	$\frac{\alpha}{n}$
3. all mass to c_j	$l_i = l_j + \frac{\alpha}{n}$	$l_j + \frac{2}{n} = u_j$	$\lvert \frac{\alpha}{n} - \frac{2}{n} \rvert$

Due to concaveness of the entropy function H_1, the entropy is reduced when the difference between the 2 arguments increases. Applying on the above, entropy gain is reduced by maximizing $|p_i - p_j|$. Regardless of the actual values of n_i and n_j, the second assignment is always dominated by the first, therefore should never be applied.
For any $\alpha > 0$, implying $n_i > n_j$ the first assignment also dominates the third. Only in the case of $\alpha = 0$ the first and third assignment are equal in terms of difference, which means the assignments are exchangeable in terms of entropy. □

When thinking of the probability wheel in any configuration, there are certain slices which we are forced to assign to specific categories, i.e. slices between two observations of the same category. According to the restrictions on the probability wheel one may assign those slices of potential free mass to adjacent or unobserved categories. With Lemma 4 we see that we should avoid at all costs to assign mass to unobserved categories. As we are dealing with a probability wheel there are only 2 adjacent categories which need to be considered. From Lemma 3 we learn that we should assign mass 'en block' which means $\frac{1}{n}$ to either of the categories if they have same size or to the one with larger number of observations(Lemma 5).

Remark 1. Furthermore with Lemma 5 it is evident that in case of a probability distribution p over the categories, with $p_i = u_i$ and $p_j = l_j$ and $n_i < n_j$, has a higher entropy than a distribution p^* which is the same on all other categories but has $p_i^* = l_i$ and $p_j^* = u_j$. Accordingly if $n_i = n_j$ then p and p^* have the same value of entropy.

Lemma 6. In the minimum entropy distribution all but a single category are at their lower or upper probability limit.

Proof. From Lemma 3 it is evident that masses are to assigned in complete slices, i.e. in chunks of $\frac{1}{n}$. Therefore a category c_i whose assigned mass neither equals its lower or upper probability has the mass $\frac{n_i}{n} = p_i$ assigned with $n_i > 0$. Such a category I call incomplete category.
If there are now any two incomplete categories c_i and c_j, the probability wheel implies that each of them has a slice of potential free mass $\frac{1}{n}$ already assigned to it. Dealing with the multinomial case, one can rearrange the wheel in the way that observations of c_i and c_j are next to each other.

Then we are able to apply Lemma 5, which gets us that all $\frac{2}{n}$ mass is assigned to the category with larger number of observations, hence resulting in the category which gets the assignment at its upper probability limit and the other at its lower probability limit.
With this method we are able to reduce any number of categories which have not attained their upper or lower probability to either 0 or only one remaining category. □

With those lemmas in mind, an algorithm is obtained without much effort.

Minimum entropy algorithm

Algorithm 1 Minimum Entropy Algorithm for NPI

Input:	F-probability intervals $[l_i, u_i]_1^n$ as generated by the NPI
Output:	A probability distribution $\hat{p} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n)$

Helping functions:

Sum(x):	returns the sum of the elements of array x
getMaxIndex(x, S):	returns the first index of the maximum value of the array x considering only indices in S

Initialization: S $\leftarrow 1, \ldots, n$

```
minEntropyNPI(l, u, p̂){
    for (i = 1 to n) do {p̂_i ← l_i}
    mass ← 1 − Sum(p̂)
    while (mass > 0) do {
        index ← getMaxIndex(p̂, S)
        d ← u_index − p̂_index
        if (d ≤ mass) then {
            p̂_index ← u_index
            S ← S − {index}
            mass ← mass − d
        } else {
            p̂_index ← p̂_index + mass
            mass ← 0
        }
    }
}
```

As a starting 'distribution' we take the lower probabilities, as each category needs to get at least this mass. As this working distribution is naturally no probability distribution we need to assign the missing mass to the categories, such that with each assignment we minimize our increase in entropy.

From Lemma 5 we are able to deduce that it is optimal to assign mass to the category with the highest lower probability limit. Assigning all remaining mass does generally not work, as we are restricted by the upper probabilities of that category. Therefore we need to split up the remaining mass and add it iteratively.

In the first step we take that amount from the remaining mass which we are able to assign optimally, in the second step the amount of mass from the now remaining, etc. So we are able to guarantee that we are step-wise optimal. When looking for categories with the highest value of the working distribution, we do not consider those which have already attained their upper probability limit.
As mentioned preliminary this algorithm gives us just one distribution with minimum entropy, yet there may be more which attain the same value. This is especially valid when there are categories with same number of observations. The proposed algorithm always chooses the category which attains the maximum first under the considered categories, as our main interest lies in the entropy value and not its actual distribution.
This algorithm successively assign mass to the largest values. The algorithm never assigns mass to unobserved categories, which is a useful property when thinking of representation of the achieved probability distribution on the wheel. The set S is employed to track those categories which still might be assigned chunks of the remaining mass.

Lemma 7. No unobserved category is assigned mass when applying the above proposed algorithm.

Proof. For all unobserved categories $j \in UJ$ the number of observations $n_j = 0$, quite naturally. Employing the above algorithm, the categories with observations are assigned the upper probabilities, starting with the highest ones. To even consider unobserved categories, all observed categories need to be assigned their upper probabilities. From this we are able to obtain the sum over all those to calculate the remaining mass which needs to be assigned to the unobserved categories. Let us assume the array of observations is ordered increasingly.

$$\begin{aligned}
\sum_{j=1}^{|UJ|} 0 + \sum_{j=|UJ|+1}^{J} \frac{n_j+1}{n} &= \sum_{j=|UJ|+1}^{J} \frac{n_j+1}{\sum_{j=1}^{J} n_j} \\
&= \sum_{j=|UJ|+1}^{J} \frac{n_j+1}{\sum_{j=|UJ|+1}^{J} n_j} \\
&= \frac{\sum_{j=|UJ|+1}^{J} (n_j+1)}{\sum_{j=|UJ|+1}^{J} n_j} \\
&= 1 + \frac{J-|UJ|}{\sum_{j=|UJ|+1}^{J} n_j} > 1
\end{aligned}$$

As the sum is already greater than 1, we would never have reached this point with the above algorithm. So the case of assigning mass to unobserved categories can never happen. □

Furthermore the algorithm return a probability distribution which is in accordance to the underlying probability wheel.

Lemma 8. *minEntropyNPI* return a probability distribution which has a representation on the probability wheel.

Proof. As *minEntropyNPI* looks for the category with the largest segment in each iteration it is essential to separate the largest chunks by categories which have been observed less frequent.
Let us assume that we observed each category j n_j times with $n_j \geq 0$ and $j = 1, \dots K$, resulting in a vector $n_{\{j\}} = (n_1, \dots, n_K)$. As the underlying model is multinomial, ordering of the categories does not matter, hence we may change the ordering of the categories c by ordering $n_{\{j\}}$ increasingly: $n_{\{(j)\}} = n_{(1)}, \dots, n_{(K)}$. If there are ties, we may choose any ordering between them. In Lemma 7 it is demonstrated that only observed categories may be assigned masses, so we define $k1$ as the index of the first observed category. A configuration of the probability wheel obtainable by *minEntropyNPI* is then:

$$c_{(k1)}, c_{(K)}, c_{(k1+1)}, c_{(K-1)}, \dots$$

In this configuration the largest segments are separated by single observations or least frequent observed ones.
Unless there are ties present between $n_{(k1)}$ and $n_{(K)}$, *minEntropyNPI* returns exactly the above presented configuration, else the returned configuration may be obtained by permuting the tied categories. □

Lemma 9. *minEntropyNPI* complies with Lemma 6

Proof. The algorithm starts with a *working distribution* where each category is set to their lower probability limit. In each iteration step one category is set to their upper probability, unless there is not enough free mass available to assign. In that case the remaining mass is assigned to one category and the algorithm exits. Therefore only this specific category is not at its upper or lower probability limit. □

Lemma 10. The algorithm does not produces situations like the first mentioned in Remark 1.

Proof. In order to generate a situation where $p_i = u_i$ and $p_j = l_j$ the algorithm must have processed c_i but not c_j. This implies that c_i is not in the set S whereas c_j is. As the algorithm processes only the categories with largest lower probability limit in each iteration step, it generally holds that

$$\max_{c_i \in S}(l_i) \leq \min_{c_i \notin S}(l_i),$$

provided that S and its compliment are not empty. Equality only happens in case of ties on the lower probability limits, where at least one of the tied categories has already been processed and at least one has not yet.
This mean for c_i and c_j that $l_j \leq \max_{c_i \in S}(l_i) \leq min_{c_i \notin S}(l_i) \leq l_i$. But this contradicts the situation as described in Remark 1, where we assumed $l_i > l_j$. Therefore our proposed algorithm is not able to produce such a situation under the constraint of $l_i > l_j$. □

References

[1] Crossman, R.J., Abellán, J., Augustin, T. and Coolen, F.P.A. (2011) Building Imprecise Classification Trees With Entropy Ranges. *Proceedings*

of the Seventh International Symposium on Imprecise Probability: Theories and Applications.