# Some research directions in Fuzzy Measures at Deakin

Deakin Team:
Prof Gleb Beliakov
A/Prof Simon James
Dr. Marek Gagolewski
Dr Jian-Zhang Wu

February 27, 2023

# CHALLENGES

- Capacities involve $2^n$ parameters, computational challenges, interpreting and assigning values
- Even under simplifications $k$-order capacities, interpreting Mobius values, ensuring constraints
- Learning capacities: are there enough data?
- Tools for operating with capacities (user-friendly)
- Broadening the area of their applications, large universes

# RESEARCH PROBLEMS

- Using fuzzy integrals in optimisation
  - Replacing linear objectives with Choquet integral to account for dependencies
  - Linear $\rightarrow$ nonlinear, but with some structure. For some types of capacities can be solved as LP
  - Mixed integer programming with Choquet integral (eg knapsack)
  - Repace linear constraints with Choquet
  - Non-convex capacities - Difference of convex, other NLP methods
- Multiobjective optimisation : scalarising functions
- Applications in MCDM: Modelling with capacities and their alternatives

# Some things we have done

- Optimising Choquet integrals instead of LPs
- Special types of capacities (anti-bouoyant $\subset$ supermodular)
- Choquet integral as the objective in knapsack
- DC programming

# CONVEX CAPACITIES

- I will go through various related but distinct cases:
- Convex capacities (and Choquet integrals)
- Convex *k*-additive for larger problems
- Convex general, how to fight exponential complexity
- Integer programming (knapsack)
- non-convex: DC optimisation
- In all cases Choquet integral is a piecewise linear function with a particular structure (linear on elements of simplicial partition)

# Convex capacities

- Supermodular capacities
- The Choquet integral is a concave function

$$\begin{aligned} \text{maximise} \quad & \mathcal{C}(\mathbf{x}) = \sum_{A \subseteq N} m(A) \min_A(\mathbf{x}) \\ \text{subject to} \quad & Cx \leq d. \end{aligned} \tag{1}$$

- In addition we use $k$-additive capacities

$$\begin{aligned} \text{maximise} \quad & \sum_{A: a \leq k} m(A) T_A \\ \text{subject to} \quad & Cx \leq d, \\ & x_j \geq T_A, \forall j \in A, A \subseteq N, \\ & a \leq k. \end{aligned} \tag{2}$$

# Convex capacities

- Supermodular 2-additive capacities
- 

$$
\begin{aligned}
\text{maximise} \quad & \sum_i m(\{i\})x_i + \sum_{i,j:i>j} m(\{i,j\})T_{ij} \qquad (3)\\
\text{subject to} \quad & Cx \le d,\\
& x_j \ge T_{ij} \text{ for all } i,j \in N, i > j\\
& x_i \ge T_{ij} \text{ for all } i,j \in N, i > j.
\end{aligned}
$$

- When some of the variables are independent (i.e., the respective $m(\{i,j\}) = 0$), it is straightforward to incorporate this knowledge into (3) by simply excluding the corresponding variables $T_{ij}$ and constraints. That is, sparsity of the problem can be accommodated to reduce the cost of the solution.

# CONVEX CAPACITIES

- Technically this is achievable for even some large $n$.
- The challenge is a) formulations for some practical problems b) identifying the capacity parameters (pairwise interactions)

# Buouyancy property

Context: general supermodular capacities. Challenge: problem size.

Definition

A weighting vector $\mathbf{w}$, $\sum_{i=1}^{n} w_i = 1, w_i \geq 0$ associated with an OWA function is said to be buoyant if $w_i \geq w_j$ whenever $i < j$.

- A weighting vector associated with an OWA operator is buoyant if and only if it is equivalent to a symmetric and submodular fuzzy measure.

Definition

A fuzzy measure $\mu$ is buoyant if all ordered weighting vectors $\mathbf{w}^{\sigma}$ associated with $\mu$ are buoyant. A fuzzy measure is antibuoyant if all ordered weighting vectors are antibuoyant.

# Buouyancy property

- In terms of the set function discrete derivatives, Definition 2 amounts to buoyant fuzzy measures satisfying:

  $\Delta_i(A \cup \{i,j\}) \leq \Delta_j(A \cup \{j\})$ for all $A \subseteq N \setminus \{i,j\}$ and pairs $i,j \notin A$.
- The discrete derivatives are decreasing with set cardinality.
- A fuzzy measure that is buoyant is necessarily submodular.
- For an antibuoyant fuzzy measure, the symmetric (additive) capacity is in its core. For a buoyant fuzzy measure the symmetric capacity is in its anti-core.
- A fuzzy measure $\mu$ is antibuoyant if and only if every reduced fuzzy measure $\mu^{T \setminus S}$ has the symmetric additive capacity on $T \setminus S$ in its core.
- The Choquet integral is consistent with the Pigou-Dalton principle (of progressive transfers) if and only if the fuzzy measure is antibuoyant.

# Buouyancy property

- We solve the problem for a supermodular capacity:

$$\begin{aligned} \text{maximise} \quad & C_\mu(\mathbf{x}) \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{j,i} x_i \leq 1, \quad j = 1, \ldots, m. \end{aligned} \tag{4}$$

- Reformulate as a large separable LP (or a set of LPs)

$$\begin{aligned} \text{maximise} \quad & \min_{\sigma \in P_n} \sum_{i=1}^{n} w_i^\sigma x_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_i x_i \leq 1, \end{aligned} \tag{5}$$

$$\begin{aligned} \text{maximise} \quad & T \\ s.t. \quad & T \leq \sum_{i=1}^{n} w_i^\sigma x_i, \forall \sigma \in P_n \\ & \sum_{i=1}^{n} a_i x_i \leq 1, \end{aligned} \tag{6}$$

# BUOUYANCY PROPERTY

- For anti-bouoyant capacities the constrained problem (5) is solved by restricting the feasible domain to $S_1$. It implies that we fix the permutation $\sigma$ and translate (5) to

$$\begin{aligned} \text{maximise} \quad & \sum_{i=1}^{n} w_i^1 x_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_i x_i \leq 1, \\ & \mathbf{x} \in S_1, \end{aligned} \tag{8}$$

- It then suffices to identify the order in which the coefficients $a_i$ of the constraint form a non-decreasing sequence, relabel accordingly the variables and solve (8).
- The case of several comonotone constraints is treated in the same way as one single constraint.
- It is also a relaxation of the knapsack problem (MIP), and can provide putative solutions.

# KNAPSACK

- Quadratic knapsack

$$\begin{aligned} \text{maximise} \quad & \sum_{i=1}^{n} v_i x_i + \sum_{i,j=1,j>i}^{n} a_{ij} x_i x_j \\ \text{subject to} \quad & \sum_{i=1}^{n} c_i x_i \leq C \\ & x_i \in \{0,1\}. \end{aligned} \tag{9}$$

- Pseudo-Boolean extension - knapsask

$$\begin{aligned} \text{maximise} \quad & Ext_{\mu}(\mathbf{x}) = \sum_{A \subseteq N} m(A) h_A(\mathbf{x}) \\ \text{subject to} \quad & \sum_{i=1}^{n} c_i x_i \leq C \\ & x_i \in D_i. \end{aligned} \tag{10}$$

# KNAPSACK

- Choquet integral (Lovasz extension)

$$\begin{aligned} \text{maximise} \quad & C_\mu(\mathbf{x}) = \sum_{A \subseteq N} m(A) T_A && (11) \\ \text{subject to} \quad & \sum_{i=1}^{n} c_i x_i \leq C \\ & T_A \leq x_i, \ \forall i \in A \text{ and } \forall A \subseteq N \\ & x_i \in D_i. \end{aligned}$$

- For 2-additive capacities the problem (11) translates into

$$\begin{aligned} \text{maximise} \quad & \sum_i m(\{i\}) x_i + \sum_{i,j:i>j} m(\{i,j\}) T_{ij} && (12) \\ \text{subject to} \quad & \sum_{i=1}^{n} c_i x_i \leq C \\ & x_i \geq T_{ij} \text{ for all } i,j \in N, i > j \\ & x_j \geq T_{ij} \text{ for all } i,j \in N, i > j \\ & x_i \in D_i. \end{aligned}$$

# KNAPSACK

- Advantages:
  - It is a standard MIP with plenty of modern computational tools
  - We can easily extend it to *k*-additive (unlike quadratic knapsack)
  - We are not restricted to 0-1 variables (Choquet is homogeneous function)
  - We can accommodate sparse general supermodular capacities
  - We can add more linear constraints if needed
  - can further restrict the class of capacities (anti-bouoyant, k-interactive)

# Difference of convex

- Consider the general optimisation problem. By replacing the linear objective with the Choquet integral we have

$$\begin{aligned} \text{maximise} \quad & \mathcal{C}(\mathbf{x}) = \sum_{A \subseteq N} m(A) \min_A(\mathbf{x}) \\ \text{subject to} \quad & Cx \leq d. \end{aligned} \tag{13}$$

- The capacity is **general** (not sub/super modular)
- We will use a decomposition of $f$ into the sum (or average) of a concave $g$ and a convex $h$ functions

$$f(x) = \frac{1}{2}(g(x) + h(x)).$$

- The class of DC functions includes piecewise linear functions, hence such a decomposition of the Choquet integral is always possible. Any capacity can be written as a weighted sum of a supermodular and a submodular capacities

# Difference of convex

- Optimise the average of concave and convex Choquet integrals,

$$\mathcal{C}(x) = \frac{1}{2}(\mathcal{C}_\mu(x) + \mathcal{C}_\nu(x)). \tag{14}$$

- Multiple locally optimal solutions, try to converge to one of them

---

- ***Algorithm DCA.***
  *Input: starting point* $\mathbf{x}_0$ *and* $k = 0$
  ***While*** *convergence criteria not met* ***do****:*
  1. *Compute* $\mathbf{y} \in \partial\mathcal{C}_\nu(\mathbf{x}_k)$.
  2. *Solve* $\mathbf{z} = arg\max_{\mathbf{x}} \mathcal{C}_\mu(\mathbf{x}) + \langle \mathbf{x}, \mathbf{y} \rangle$ *for a fixed* $\mathbf{y}$.
  3. *Set* $\mathbf{x}_k = \mathbf{z}$ *and* $k = k + 1$.

---

# DIFFERENCE OF CONVEX

- Here $\partial \mathcal{C}_\nu$ denotes the Clarke's subdifferential of $\mathcal{C}_\nu$, so that $\mathbf{y}$ is any of its subgradients.
- The convergence criteria involve reaching a given threshold on pairwise difference of the optima of two iterations.
- The problem at step 2 is a linear programming problem, which in the case of 2-additive capacities simplifies to (12).
- Step 1 requires calculation of any subgradient of a piecewise linear convex function $\mathcal{C}_\nu$. By taking one-sided derivatives :

$$y_i = m_\nu(\{i\}) + m_\nu(A)\delta(x_i, A)$$

with $\delta(x_i, A) = 1/b$ if $x_i = \min_A(x)$, $B = \{j : x_j = \min_A(x)\}$ and 0 otherwise. Note that $\langle \mathbf{x}, \mathbf{y} \rangle \leq \mathcal{C}_\nu(x)$ which can be established by analysing the subdifferential of the function $\min_A(x)$.

# Difference of convex

- The LPs at step 2 in different iterations of the DCA differ only by some of the coefficients in the linear objective. It means that from the practical implementation point of view one can avoid the overheads associated with setting up problem, and only modify its objective, which will lead to significant savings.
- There are many generic methods for solving the NLP. An important challenge is a large number of constraints on the decision variables $x$. The DC approach was designed to use linear programming at the inner step of the DCA, because LP methods successfully handle many thousands of constraints and take advantage of their sparsity.
- We built a tool for solving such DCA, now setting up computational experiments, study scalability.

# Conclusions

- For special types of capacities using Chquet integral amounts to solving one or a sequence of LPs, hence the approach appears to be scalable
- For MIP, like knapsack problem we reduced the problem to classical MIP formulation/and hence numerical tools
- For non-convex objectives we have the DC algorithm, which seems to be quite efficient, although converges locally.
- An important issue is the potentially large number of linear constraints, most likely sparse constraints. Hence the reduction of the NLPs to sequential LPs or linear MIPs brings an advantage of utilising proven free and commercial libraries.