Setwise Max-Margin Formulation

Preliminary assumptions:

- A multi attribute feature space $\mathbf{X}$; n is the number of features. A configuration is written as $(x_1, \ldots, x_n)$.
- We suppose binary features: $x_i \in \{0, 1\}$.
- We assume possible options are specified by linear configuration constraints; as for example $x_1 + x_2 \geq 1$ to express an OR constraint.
  We will simply write $\mathbf{x} \in \mathit{Feasible}(\mathbf{X})$ in the optimizations below.
- Vector of weights $\mathbf{w} = (w_1, ..., w_n)$; the utility of an option $\mathbf{x}$ is then $\mathbf{w} \cdot \mathbf{x} = \sum_{z=1}^{n} w_z x_z$.

Note: The weight vector $\mathbf{w}$ is unknown to the system but we are given preference constraints (that encode a "feasible region").

Input:

- The configuration constraints representing $\mathit{Feasible}(\mathbf{X})$
- $\mathcal{D}$ the "learning set", consisting of pairwise comparisons between a more preferred product $\mathbf{y}_+$ and a less preferred product $\mathbf{y}_-$. Each preference induces a constraint of the type $\mathbf{w} \cdot (\mathbf{y}_+ - \mathbf{y}_-) \geq 0$.
- Possibly other additional constraints (representing some kind of "prior" knowledge, as for example $w_2 > 0.3$)
  We'll write $w_i^{\perp} \leq w_i \leq w_i^{\top}$ to represent such constraints.

Output:

- We want to find a *set* of vectors $\mathbf{w}^1, ..., \mathbf{w}^k$ (with k given) of utility weights and associated configurations $\{\mathbf{x}^1, ..., \mathbf{x}^k\}$ such that
  - All preferences are satisfied and does that by the largest margin; so for that a binary preference $\mathbf{y}_+ \succeq \mathbf{y}_-$ induces the constraint $\mathbf{w} \cdot (\mathbf{y}_+ - \mathbf{y}_-) \geq M$.
  - For each i, we impose $\mathbf{x}^i$ to be the "best" option among the $\mathbf{x}^1, \ldots, \mathbf{x}^k$ when evaluated according to $\mathbf{w}^i$; we require these constraints to hold by at least the shared margin M.
    The imposed constraints are of the type: $\mathbf{w}^i \cdot (\mathbf{x}^i - \mathbf{x}^j) \geq M$ with $i \neq j,\ i, j \in [1, k]$.

Additional requirements:

- **Slack variables**: allow for non satisfied preference constraints and include a penalty term $\sum_{h=1}^{|\mathcal{D}|} \varepsilon_h$ in the objective function

- **Sparsification**: we introduce a L1 norm term in the objective. This is achieved by adding the following term in the objective function

$$\sum_{i=1}^{k}\sum_{z=1}^{n}|w_z^i|.$$

- **Accordance of the $\mathbf{x}^i$s with $\mathbf{w}^i$s**: we would like to have

$$\mathbf{x}^i = \arg\max_{\mathbf{x}\in Feasible(\mathbf{X})} \mathbf{w}^i \cdot \mathbf{x}.$$

  However since this is not possible (the solver will pick $\mathbf{x}^i$ high enough to meet the margin, but not necessarily better), we favor $\mathbf{x}^i$ with high utility by adding a term $\sum_{i=1}^{k} \mathbf{w}^i\mathbf{x}^i$ in the objective function.

Initial Non-linear Optimization We first provide a formulation with quadratic terms, that is not directly solvable.

$$\max_{M,\mathbf{w}^i,\mathbf{x}^i} M - \alpha\sum_h \varepsilon_h - \beta\sum_{i=1}^{k}\sum_{z=1}^{n}|w_z^i| + \gamma\sum_{i=1}^{k}\sum_{z=1}^{n} w_z^i x_z^i \tag{1}$$

$$s.t.\ \ \mathbf{w}^i \cdot (\mathbf{y}_+^h - \mathbf{y}_-^h) \geq M - \varepsilon_h \qquad \forall(\mathbf{y}_+^h,\mathbf{y}_-^h)\in\mathcal{D}, \forall i\in\{1,\dots,k\} \tag{2}$$

$$\mathbf{w}^i \cdot (\mathbf{x}^i - \mathbf{x}^j) \geq M \qquad \forall j\neq i;\ i,j\in\{1,\dots,k\} \tag{3}$$

$$w_i^\perp \leq w_i \leq w_i^\top \qquad \forall i\in\{1,\dots,k\} \tag{4}$$

$$\mathbf{x}^i \in Feasible(\mathbf{X}) \qquad \forall i\in\{1,\dots,k\} \tag{5}$$

$$\varepsilon_h \geq 0 \qquad \forall h\in\{1,\dots,|\mathcal{D}|\} \tag{6}$$

$$M \geq 0 \tag{7}$$

Decision variables:

- $\mathbf{w}^1,\dots,\mathbf{w}^k$: set of k utility weights; each $\mathbf{w}^i$ is a vector of n binary attributes, for example $\mathbf{w}^i = (w_1^i,\dots,w_n^i)$.
- $\mathbf{x}^1,\dots,\mathbf{x}^k$: set of k options; each $\mathbf{x}^i$ is a vector of n binary attributes, for example $\mathbf{x}^i = (x_1^i,\dots,x_n^i)$.
- M: the shared margin.
- $\varepsilon_1,\dots,\varepsilon_{|\mathcal{D}|}$: slack variables representing penalties for violated preferences.

Note that, we are choosing the options $\mathbf{x}^1,...,\mathbf{x}^k$ and the weights $\mathbf{w}^1,...,\mathbf{w}^k$ simultaneously: since we want to maximize M, the optimizer will be better off by choosing a set of outcomes $\mathbf{x}^i$ that divide the weight space roughly equally, and the utility functions such each $\mathbf{w}^i$ should lie (intuitively) near the centre of each subregion.

This initial formulation is problematic, as we have quadratic terms. However, there is a solution: in fact, by using integer programming tricks, the problem can be formulated as a mixed integer linear program (MILP).

***Setwise* max margin** We propose a Mixed Integer Linear Programming (MILP) model with *non-negative weights*.

$$\max_{M,\mathbf{w}^i,\mathbf{x}^i,\mathbf{A}^{i,j}} \quad M - \alpha \sum_h \varepsilon_h - \beta \sum_{i=1}^{k} \sum_z w_z^i + \gamma \sum_{i=1}^{k} \sum_{z=1}^{n} A_z^{i,i} \tag{8}$$

$$s.t. \quad \mathbf{w}^i \cdot (\mathbf{y}_+^h - \mathbf{y}_-^h) \geq M - \varepsilon_h \qquad \forall(\mathbf{y}_+^h, \mathbf{y}_-^h) \in \mathcal{D},\ \forall i \in \{1,\ldots,k\} \tag{9}$$

$$\sum_{z=1}^{n} A_z^{i,i} - A_z^{i,j} \geq M \qquad \forall j \neq i,\ i,j \in \{1,\ldots,k\} \tag{10}$$

$$A_z^{i,i} \leq w\!\uparrow x_z^i \qquad \forall i \in \{1,\ldots,k\},\ z \in \{1,\ldots,m\} \tag{11}$$

$$A_z^{i,i} \leq w_z^i \qquad \forall i \in \{1,\ldots,k\},\ z \in \{1,\ldots,m\} \tag{12}$$

$$A_z^{i,j} \geq w_z^i - C \cdot (1 - x_z^j) \qquad \forall j \neq i,\ i,j \in \{1,\ldots,k\},\ z \in [1,m] \tag{13}$$

$$A_z^{i,j} \geq 0 \qquad \forall j \neq i,\ i,j \in \{1,\ldots,k\},\ z \in \{1,\ldots,m\} \tag{14}$$

$$w_i^{\perp} \leq w_i \leq w_i^{\top} \qquad \forall i \in [1,k] \tag{15}$$

$$\mathbf{x}^i \in Feasible(\mathbf{X}) \qquad \forall i \in \{1,\ldots,k\} \tag{16}$$

$$x_z^i \in \{0,1\} \qquad \forall i \in \{1,\ldots,k\}; z \in \{1,\ldots,m\} \tag{17}$$

$$A_z^{i,j} \geq 0 \qquad \forall i,j \in \{1,\ldots,k\}; \forall z \in \{1,\ldots,m\} \tag{18}$$

$$\varepsilon_h \geq 0 \qquad \forall h \in \{1,\ldots,|\mathcal{D}|\} \tag{19}$$

$$M \geq 0 \tag{20}$$

In the optimization, the decision variables are the following:

- M is the shared margin.
- $\mathbf{w}^1, ..., \mathbf{w}^k$ is a set of utility vectors; each vector defined over m attributes (features): $\mathbf{w}^i = (w_1^i, ..., w_m^i)$.

Important: The weights need to be non-negative in this formulation: $w_z^{\perp} \geq 0\ \forall z$.

- $\mathbf{x}^1, ..., \mathbf{x}^k$ is a set of configurations (options) with each configuration $\mathbf{x}^i = (x_1^i, ..., x_m^i)$; each element x_z^i is binary.
- ε slack variables to represent cost of unsatisfied constraints in D.
- $\mathbf{A}^{i,j}$ encodes the vector $(w_1^i x_1^j, ..., w_m^i x_m^j)$, the element-by-element product of $\mathbf{w}^i$ and $\mathbf{x}^j$; $A_z^{i,j}$ is forced to take value $w_z^i x_z^i$.

We list all the parameters used, including newly introduced parameters:

- n is the number of attributes.
- k is the size of the set.
- $\alpha > 0$ control the tolerance with respect to violated constraints (we can differentiate the tolerance for the two different types of constraints).
- $\beta > 0$ is the weight associated to the L1 norm considering the absolute value of the weights (since this formulation assumes non negative weights w_z^i, we simply sum over the weights in the objective function).
- $\gamma > 0$ in order to favour the choice of $\mathbf{x}^i$ that achieves high utility with respect to $\mathbf{w}^i$.

- $\mathcal{D}$ is the *learning set*: a set of pairwise comparisons known to the system.
- Upper and lower bounds (from prior knowledge) on weights: $w_1^{\perp}, \ldots, w_n^{\perp}$ and $w_1^{\top}, \ldots, w_n^{\top}$,
- $w\uparrow$ is any upper bound of the value of the utility weights (for instnace, it can bet set to $\max_z w_z^{\top}$).
- C is an arbitrary large number.

Intuition:
The solver aims at setting M, that is a decision variable, as large as possible.

- Constraint 3 of the original program is replaced by constraint 10, enforcing M to be smaller than $\sum_z A_z^{i,i} - A_z^{i,j}$ for any i, j.
  $\sum_z A_z^{i,i} - A_z^{i,j}$ is forced to evaluate to $\mathbf{w}^i \cdot (\mathbf{x}^i - \mathbf{x}^j)$ by setting some specific additional constraints.
- Note that, in order to maximize M, the solver will try to keep the $A_z^{i,i}$ as large as possible and the $A_z^{i,j}$ (with $i \neq j$) as small as possible.
- The fact that $A_z^{i,j}$ takes value $w_z^i x_z^j$ (and therefore $\mathbf{w}^i \cdot \mathbf{x}^j = \sum_{z=1}^n A_z^{i,j}$) is achieved by setting additional constraints.
  We differentiate between the cases 1) $A_z^{i,i}$ and 2) $A_z^{i,j}$ with $i \neq j$ (since they appear with opposite signs in the objective function):
  - *[case $A_z^{i,i}$]* Constraints 11 and 12 together force each $\mathbf{A}^{i,i}$ to be the element-by-element product of $\mathbf{w}^i$ and $\mathbf{x}^i$ ($A_z^{i,i} = w_z^i x_z^i$):
    * If $x_z^i = 0$ then (constraint 11) also $A_z^{i,i}$ must be 0;
    * Otherwise (if $x_z^i = 1$) $A_z^{i,i} \leq w_z^i$ (constraint 12) but because of the objective function maximizing M, and each A constrain M, the solver will set $A_z^{i,i}$ to w_z^i.
  - *[case $A_z^{i,j}$ with $i \neq j$]* Similarly constraints 13 and 18 together make it so that $\mathbf{A}^{i,j}$ encodes $(w_1^i x_1^j, ..., w_m^i x_m^j)$, the element-by-element product of $\mathbf{w}^i$ and $\mathbf{x}^j$ (as C is an arbitrary large constant, constraint 13 is binding only when x_z^j is 1, otherwise it is always satisfied).