INTRODUCTION TO MULTICRITERIA DECISION ANALYSIS (MCDA)

05 – TOPSIS and Fuzzy TOPSIS

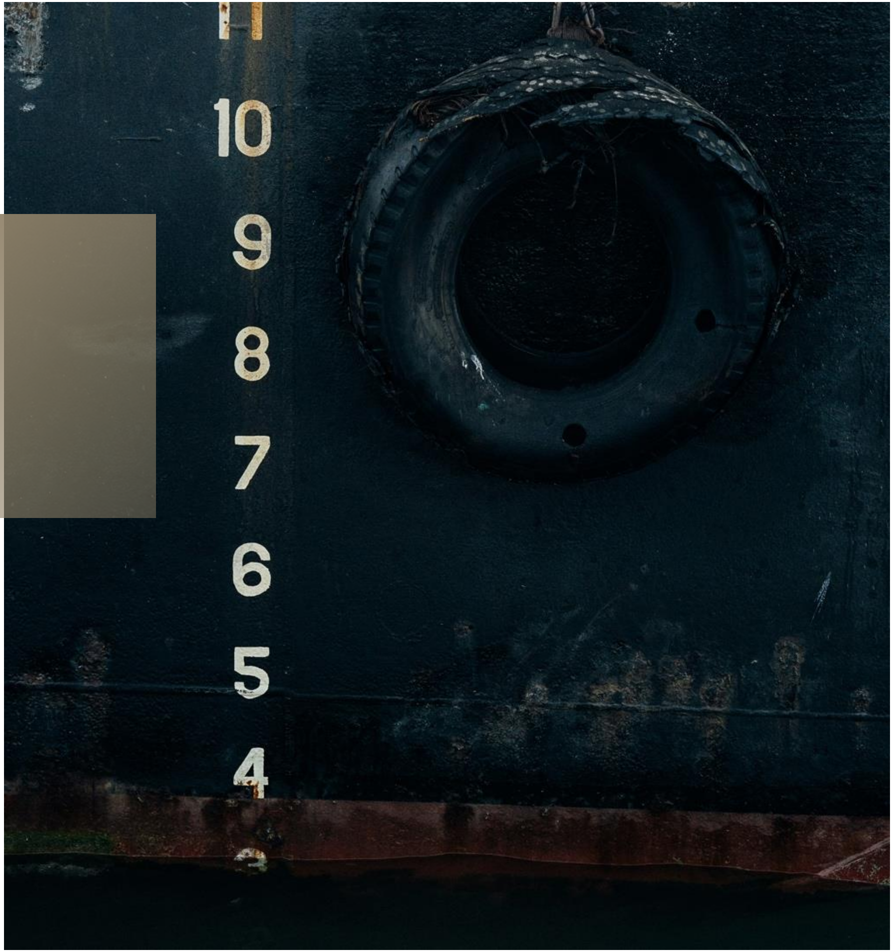
Fellipe Martins

SNEAK PEEK

Today we are going to explore the most famous method in MCDA (probably) – the Analytical Hierarchical Process (AHP).

I hope you had time to read the corresponding materials, and let’s dive in!

TABLE OF CONTENTS



TOPSIS

Simple and elegant
(but also limited)

What does TOPSIS stand for?

TOPSIS stands for **Technique for Order Preference by Similarity to Ideal Solution**.

It is a decision-making method that **ranks alternatives by comparing their distances to the best (ideal) and worst (negative-ideal) solutions**.

The alternative **closest to the ideal** and **farthest from the negative-ideal** is ranked highest.

As such it attempts at **minimizing the distance to the best** and **maximize the distance to the worst** points in the data.

INTUITION

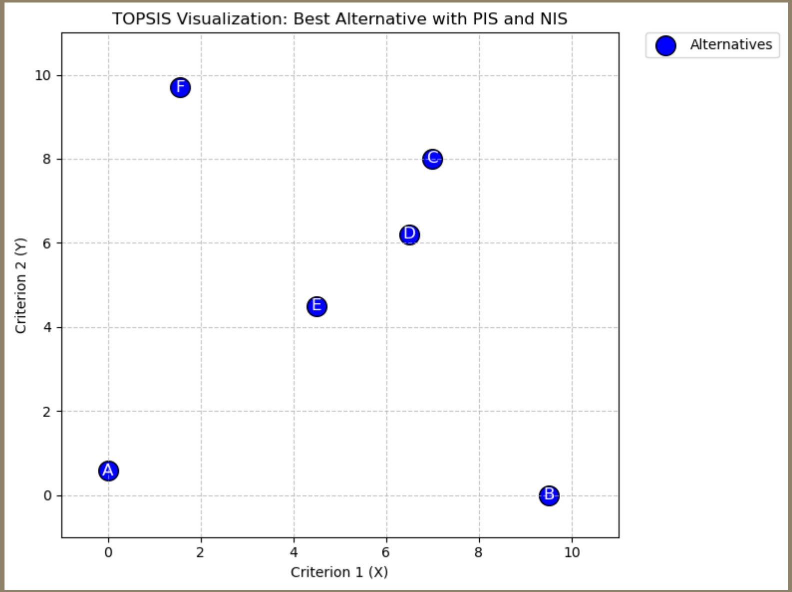
Let’s assume we have 6 alternatives (A, B, … , F) to choose from, considering criteria 1 and 2.

We need to find the option that best compromises between the two criteria, right?

TOPSIS does that by trying to find the best and worst scenarios.

The option that minimizes the distance between the best scenario while maximizing the distance to the worst scenario should be the safest bet.

Using the graph, can you think about where the best and worst scenarios would be?



INTUITION

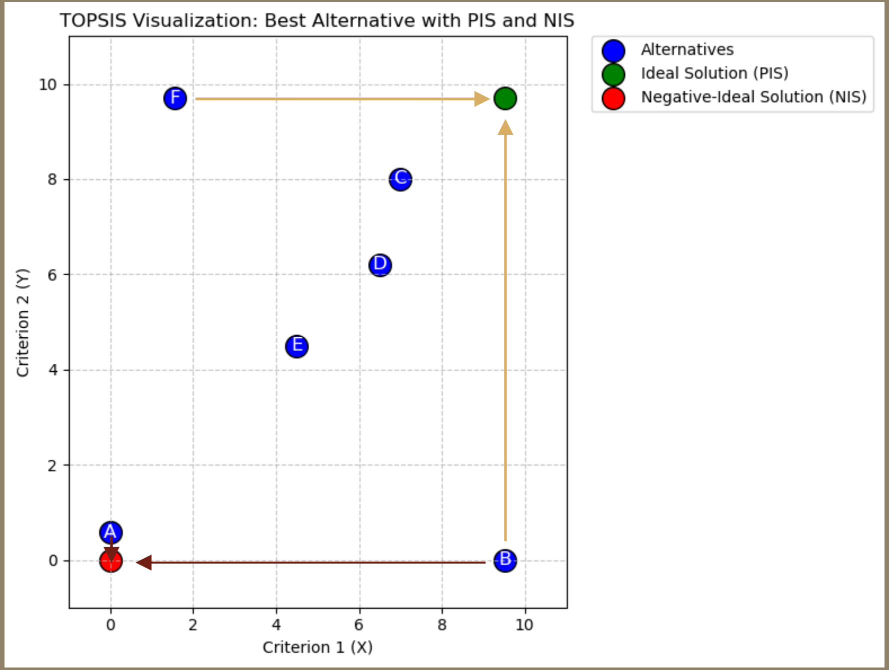
The best and worst scenarios are fictitious and do not match real alternatives.

But they are built from data stemming from the alternatives.

1. Get the best values for both criteria: this should give you the best scenario (i.e., Positive Ideal Solution or PIS).

2. Get the worst values for both criteria: this should give you the worst scenario (i.e., Negative Ideal Solution or NIS).

3. Try to find the option that gets closer to best, but also getting the farthest from the worst.



INTUITION

Let's take A: it performs horribly on both criteria – clearly doesn't work.

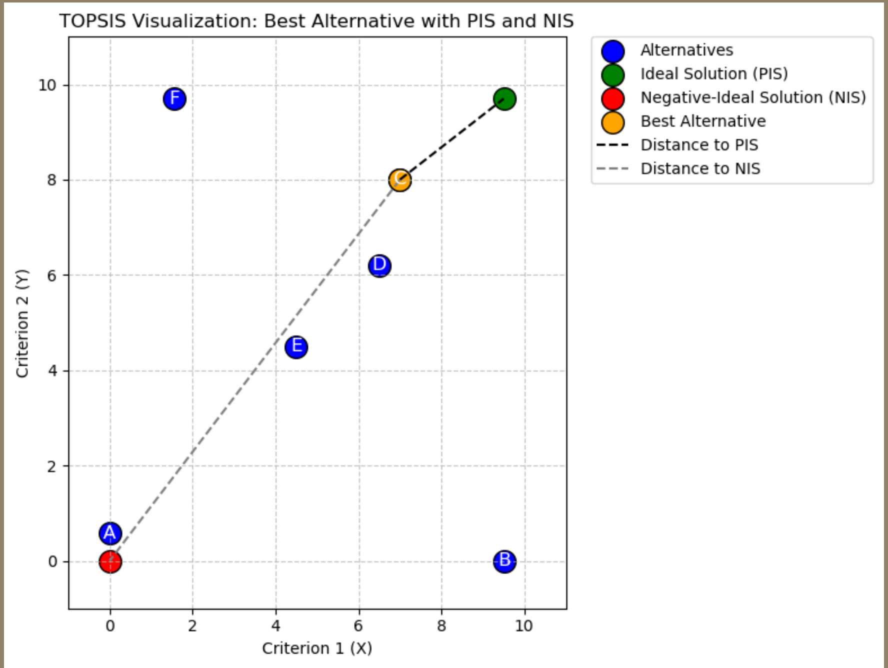
Let's take F and B, they both perform well one one criterion, but not the other.

Here the winner is easy to see (because I chose the position of points!) – it is C.

C does two things at the same time:

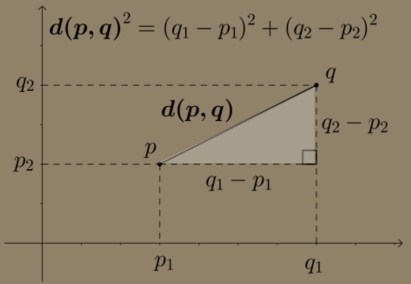
- Minimizes the distance to the Positive Ideal Solution (PIS)
- Maximizes the distance to the Negative Ideal Solution (NIS).

TOPSIS does this using Euclidean distance.

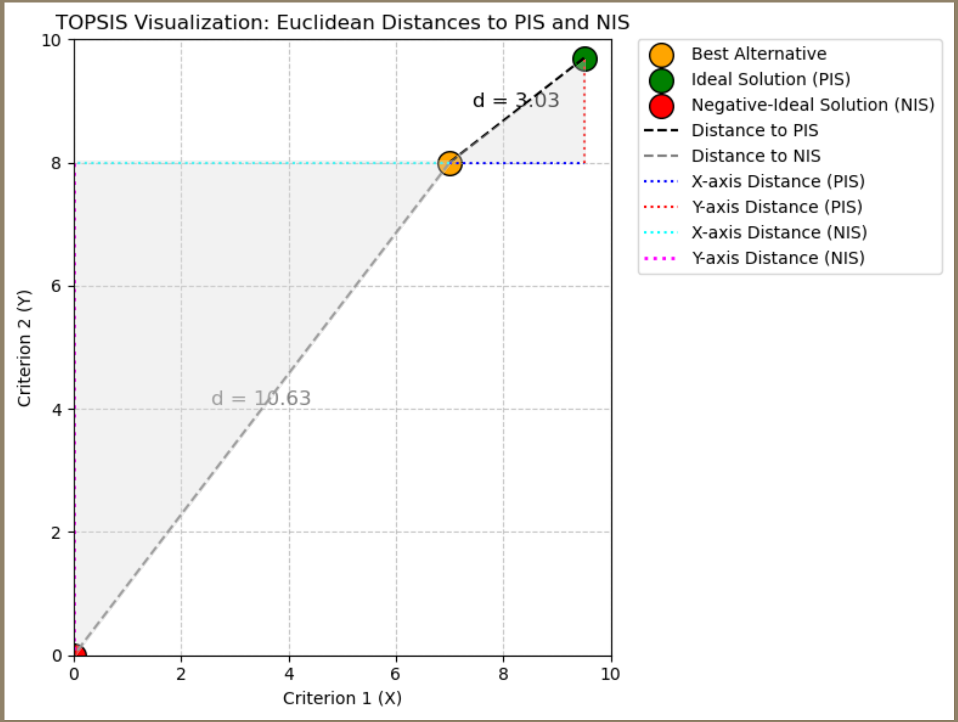


INTUTION

Maybe you don’t remember Euclidean distance from high school, so here it is again:



$$d(p,q) = \sqrt{(q_1 - p_1)^2 - (q_2 + p_2)^2}$$



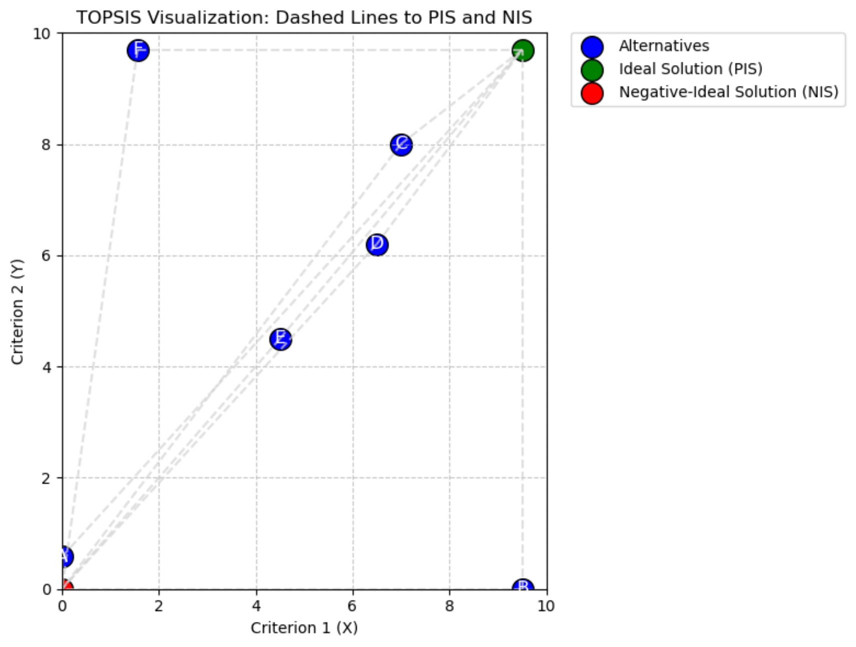
INTUITION

Now we compute the distance between all alternative to both PIS / NIS.

We use a relative closeness calculation to extract ratios.

We use this ratio to rank alternatives:

A_i	C^*	Rank
C	0.876	1
D	0.789	2
E	0.654	3
F	0.543	4
B	0.321	5
A	0.000	6



INTUITION

However, if we had only 2 criteria, it would be too easy.

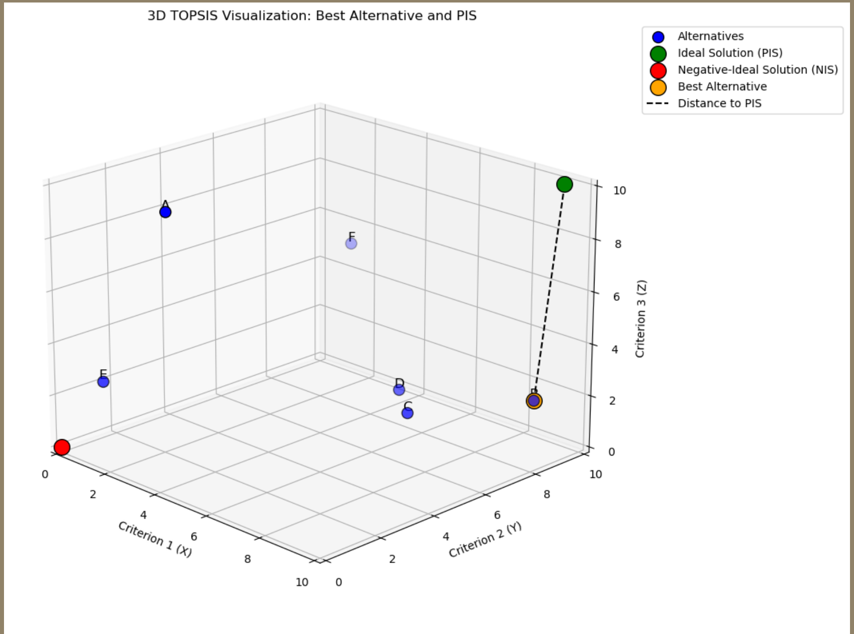
If we have 3 criteria, it becomes more complex.

Larger matrices will result in more dimensions (and we can't really visualize more than 3 dimensions...).

Also, adding more criteria alters the calculation, and another alternative can take the lead (because of the effect of the additional dimensions).

We can't visualize, but that does not stop us from computing and reaching a decision.

Is there anything missing, if we compare to AHP or Fuzzy AHP?



Does TOPSIS have a scale like in AHP?

⚠️ **TOPSIS does not have a pre-built scale as in AHP.**

TOPSIS works on raw or assigned data. For instance, take the following criteria for decision of a car purchase:

- **Price:** raw data in $
- **Fuel Efficiency:** raw data in km/l
- **Safety:** assigned mark (as in 1~10), raw data as grades in a test (whatever the scale in the test), or qualitative / ordinal converted to a scale (A, B, C → 3, 2, 1), etc.

⚠️ Assigned data is arbitrary, obviously

Either way, TOPSIS is made to be simple….

Step 1

Let’s create a decision matrix D consisting of m alternatives $(A_1, \dots, A_m)$ and n criteria $(X_1, \dots, X_n)$.

Each element x_{ij} represents the score of alternative A_i for criterion X_i.

$$
\boldsymbol{D} = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_m \end{array}
\begin{array}{c}
\begin{array}{cccccc} X_1 & X_2 & \cdots & X_j & \cdots & X_n \end{array} \\
\left[\begin{array}{cccccc}
x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1n} \\
x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2n} \\
\vdots & \vdots & \dots & \vdots & \dots & \vdots \\
x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{in} \\
\vdots & \vdots & \dots & \vdots & \dots & \vdots \\
x_{m1} & x_{m2} & \cdots & x_{mj} & \cdots & x_{mn}
\end{array} \right]_{m \times n}
\end{array}
$$

It is basically a spreadsheet with alternatives as rows, and criteria as columns.

Step 2

We transform the decision matrix D into a **normalized decision matrix** R.

- Normalization ensures that all criteria are on the same scale, preventing any single criterion from dominating the decision due to differences in units or magnitude.

Each element r_{ij} of the normalized matrix R is calculated using the formula:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$$

where:

- x_{ij} is the original value in the decision matrix D.
- The resulting value r_{ij} is a dimensionless number between 0 and 1, preserving the relative importance of each criterion.

Step 2

In the normalization formula:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$$

the denominator $\sqrt{\sum_{i=1}^{m} x_{ij}^2}$

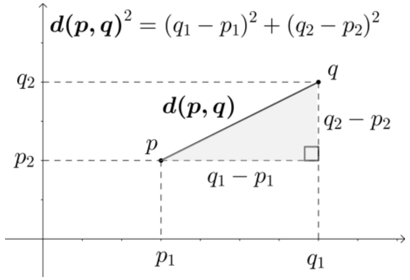
is the Euclidean norm (or magnitude) of the column j. This means:

1. Take all the values in column j (i.e., all x_{ij} values for different alternatives $A_1, \ldots, A_m$).
2. Square each value.
3. Sum up all the squared values.
4. Take the square root of the sum.

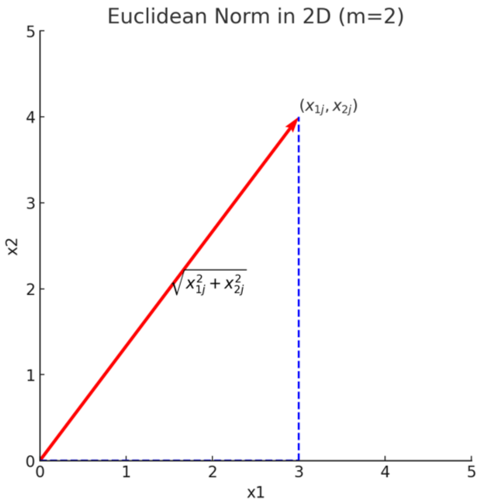
This gives the length of the vector formed by the column j, treating it as a point in an m-dimensional space.

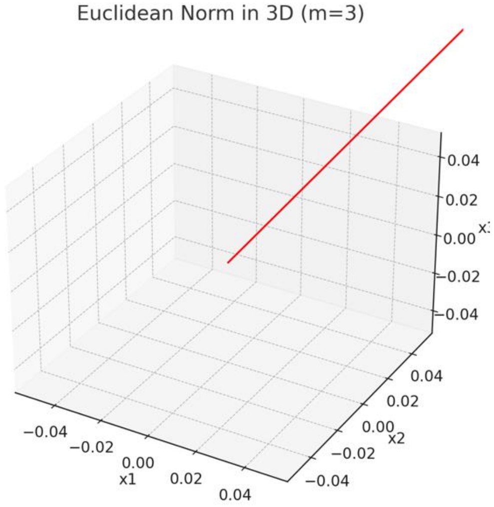
Step 2

You can try and visualize this as a vector, 2D as in two criteria ($m = 2$), 3D as in three criteria ($m = 3$). For larger $m > 3$, this generalizes to an **m-dimensional space**, where the Euclidean norm is the length of the column vector (but we can't visualize it).



$$d(p,q) = \sqrt{(q_1 - p_1)^2 - (q_2 + p_2)^2}$$





Step 2

Let's say we have the following decision matrix D for 3 alternatives and 2 criteria:

$$D = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{cc} X_1 & X_2 \\ \end{array}$$

$$D = \begin{array}{ccc} & X_1 & X_2 \\ A_1 & 20 & 30 \\ A_2 & 25 & 35 \\ A_3 & 22 & 32 \end{array}$$

To normalize D, we calculate r_{ij} for each value:

For X_1: $\sqrt{\sum_{i=1}^{3} x_{i1}^2} = \sqrt{20^2 + 25^2 + 22^2} = \sqrt{400 + 625 + 484} = \sqrt{1509} \approx 38.84$

For X_2: $\sqrt{\sum_{i=1}^{3} x_{i2}^2} = \sqrt{30^2 + 35^2 + 32^2} = \sqrt{900 + 1225 + 1024} = \sqrt{3149} \approx 56.12$

(square → sum → root)

Step 2

Compute the Normalized Values:

For X_1: $r_{11} = \frac{20}{38.84} \approx 0.515, \quad r_{21} = \frac{25}{38.84} \approx 0.644, \quad r_{31} = \frac{22}{38.84} \approx 0.566$

For X_2: $r_{12} = \frac{30}{56.12} \approx 0.535, \quad r_{22} = \frac{35}{56.12} \approx 0.624, \quad r_{32} = \frac{32}{56.12} \approx 0.570$

Normalized Decision Matrix R:

$$R = \begin{bmatrix} 0.515 & 0.535 \\ 0.644 & 0.624 \\ 0.566 & 0.570 \end{bmatrix}$$

Step 3

Step 3 is all about getting weights and multiplying them by r_{ij}.

⚠️ The classical method uses assigned weights (just as we would assign values to the alternatives x criteria):

1. List all criteria $(X_1, X_2, \dots, X_n)$
2. Assign a weight w_j to each criterion X_j, reflecting its relative importance.
3. Normalize the weights so that they sum to 1:

$$w_j = \frac{Assigned\ weight\ for\ X_j}{\sum_{j=1}^{n} Assigned\ weight\ for\ X_j}$$

Suppose you have 3 criteria:
X_1: Price (assigned weight = 5)
X_2: Fuel Efficiency (assigned weight = 3)
X_3: Safety (assigned weight = 2)

$$w_1 = \frac{5}{5+3+2} = 0.5, \quad w_2 = \frac{3}{10} = 0.3, \quad w_3 = \frac{2}{10} = 0.2$$

Step 3

However this is as arbitrary as assigning values as in alternatives x criteria.

Another common option is using simple geometric mean AHP (compute the geometric mean for each row; normalize the means to get the weights – **see 03 – AHP**).

A third, less common way is by calculating entropy. As it is a bit more complicated, I will leave an example in the Appendix.

1. Assign arbitrary weights, normalized.
2. Use AHP as an intermediary to elicit weights
3. Use entropy calculation.
4. Hidden secret option: no weights! Each criteria receives $1/n$, n being the number of criteria.

Either way, make sure the weights sum up to 1.

Among these, most people follow the first strategy as it fits the rest of the easy steps in classical TOPSIS.

Step 3

Given you chose a weight scheme, the normalized decision matrix R is multiplied by the weight w_j assigned to each criterion X_j.

The weights reflect the relative importance of each criterion, and they must satisfy:

$\sum_{j=1}^{n} w_j = 1$ and $w_j > 0$ for all $j = 1, \dots, n$

Each element v_{ij} in the new matrix V is computed as $v_{ij} = w_j \cdot r_{ij}$.

- r_{ij} comes from the normalized decision matrix R.
- w_j is the weight assigned to criterion j.

(just multiply r_{ij} by the weight of the criterion)

$$V = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccccc} X_1 & X_2 & \cdots & X_j & \cdots & X_n \end{array} \\ \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1j} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2j} & \cdots & v_{2n} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ v_{i1} & v_{i2} & \cdots & v_{ij} & \cdots & v_{in} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mj} & \cdots & v_{mn} \end{bmatrix}_{m \times n} \end{array}$$

Step 4

In this step, we determine the **Positive Ideal Solution (PIS)** and **Negative Ideal Solution (NIS)** based on the **weighted normalized decision matrix** V. These solutions represent the **best** and **worst** possible alternatives, respectively.

Be mindful of the variation in the notation across papers:

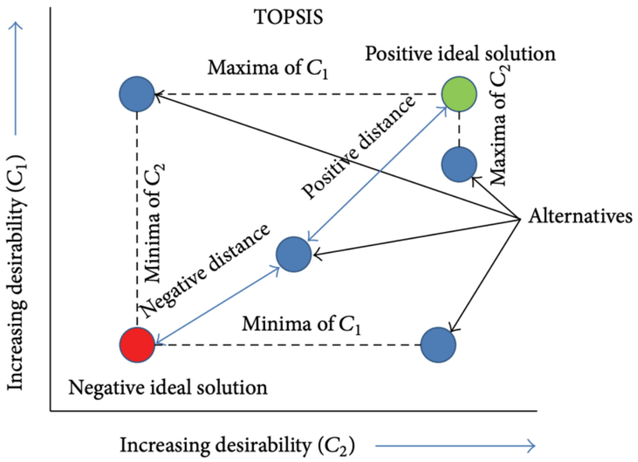
- **PIS:** A^+, A^*, V^+
- **NIS:** A^-, V^-

- **PIS (The best possible values for each criterion):**

$$V^+ = \{v_1^+, v_2^+, \ldots, v_n^+\} = \left\{\max_i v_{ij} \mid j \in J\right\}, \left\{\min_i v_{ij} \mid j \in J\prime\right\}$$

- **NIS (The worst possible values for each criterion.):**

$$V^- = \{v_1^-, v_2^-, \ldots, v_n^-\} = \left\{\min_i v_{ij} \mid j \in J\right\}, \left\{\max_i v_{ij} \mid j \in J\prime\right\}$$



Step 4

Example:

If we have a weighted normalized decision matrix V with three alternatives (A_1, A_2, A_3) and 2 criteria (X_1, X_2):

$$V = \begin{bmatrix} 0.400 & 0.500 \\ 0.300 & 0.700 \\ 0.450 & 0.600 \end{bmatrix}$$

Assuming:

- X_1 is a benefit criterion (J) – i.e., a criterion we want to **maximize**
- X_2 is a cost criterion (J') – i.e., a criterion we want to **minimize**

Then:

$$V^+ = \{\max(0.400, 0.300, 0.450), \min(0.500, 0.700, 0.600)\} = \{0.450, 0.500\}$$

$$V^- = \{\min(0.400, 0.300, 0.450), \max(0.500, 0.700, 0.600)\} = \{0.300, 0.700\}$$

Step 5

Now we use this to compute the distance of each alternative from both the Positive Ideal Solution (PIS) and the Negative Ideal Solution (NIS) using the Euclidean distance.

For each alternative A_i:

1. Separation from the Positive Ideal Solution (PIS):

$$S_i^+ = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^+)^2}$$

2. Separation from the Negative Ideal Solution (NIS):

$$S_i^- = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^-)^2}$$

where:

- v_{ij} is the weighted normalized value of alternative A_i for criterion X_j.
- ${v_{ij}}^+$is the best value for criterion X_j (from V^+).
- ${v_{ij}}^-$ is the worst value for criterion X_j (from V^-).

Step 5

Let's assume we have three alternatives (A_1, A_2, A_3) and two criteria (X_1, X_2) , with the following **weighted normalized decision matrix** V:

$$V = \begin{bmatrix} 0.400 & 0.500 \\ 0.300 & 0.700 \\ 0.450 & 0.600 \end{bmatrix}$$

Then for alternative $A_1 = (0.400, 0.500)$:

$$S_1^+ = \sqrt{(0.400 - 0.450)^2 + (0.500 - 0.500)^2}$$
$$S_1^+ = \sqrt{(-0.050)^2 + (0.000)^2} = \sqrt{0.0025} \approx 0.050$$

$$S_1^- = \sqrt{(0.400 - 0.300)^2 + (0.500 - 0.700)^2}$$
$$S_1^- = \sqrt{(0.100)^2 + (-0.200)^2} = \sqrt{0.0100 + 0.0400} = \sqrt{0.0500} \approx 0.224$$

Step 6

Now we need to use those two distances to figure out ranking!

For each alternative A_i, the **relative closeness coefficient** is given by:

$$C_i^* = \frac{S_i^-}{S_i^+ + S_i^-}$$

where:
- S_i^+ is the distance to the positive ideal solution (PIS).
- S_i^- is the distance to the negative ideal solution (NIS).
- C_i^* represents how close an alternative is to the best solution relative to the worst.

C_i^* is a value between 0 and 1.
- A higher C_i^* means the alternative is closer to the best solution.
- A lower C_i^* means the alternative is closer to the worst solution.
- The alternative with the highest C_i^* is ranked as the best choice.

Step 7

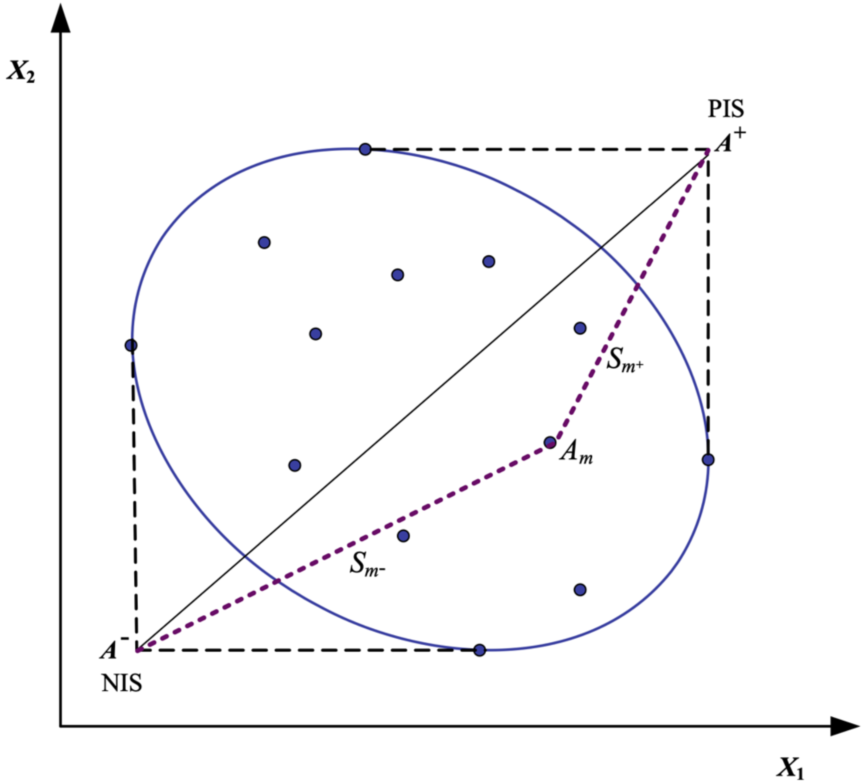
Rank the alternatives in descending order:

Alternative	$\mathbf{S_i}^{+}$	$\mathbf{S_i}^{-}$	$\mathbf{C_i}^{*}$	**Rank**
A_1	0.050	0.224	0.818	1st
A_2	0.100	0.180	0.643	2nd
A_3	0.120	0.160	0.571	3rd

Classical visualization

In the classical paper (Hwang and Yoon 1981), the round form in the middle represents the probability of real values.

- All values are possible, although some are useless like $A_1 = (0,0)$,
- Values outside this circular area are unlikely to happen (but not impossible), including the so-called "perfect solutions" (almost perfectly high values across all criteria), as they usually do not exist in the real world.
- Some points in the visualizations I made in the intuition section (A, B, F) were made for illustrating TOPSIS, but probably would not match real decisions.



Choice behavior

The curves (hyperbolas) represent points with the same relative closeness C_i^*.

When $C_i^* > 0.5$ (Closer to PIS), the indifference curves are convex towards PIS.

DMs prioritize balanced alternatives:

• They prefer an alternative that is well-rounded across criteria rather than one that excels in one criterion but fails in another.

• Example: A student selection committee might prefer a candidate with grades (B, B) over another with (A, F).

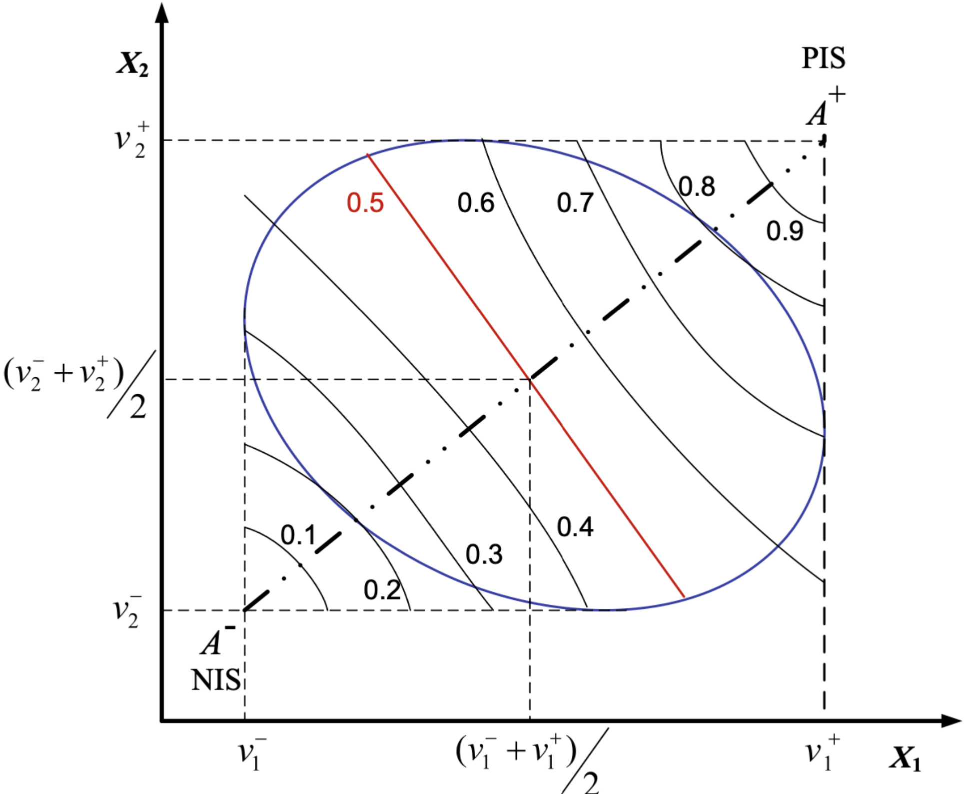
When $C_i^* < 0.5$ (Closer to NIS), the indifference curves are concave towards NIS.

DMs exhibit risk-prone behavior:

• They might prefer an extreme alternative, which is strong in one criterion and weak in another.

• Example: A committee might prefer an applicant with (A, F) rather than (C, C), believing the high strength in one criterion compensates for the weakness in another.

→ *compare to prospect theory (Kahneman and Tversky 1979)*



Group decision

What if we have more than one decision-maker (DM)?

Do the same for all DMs, but when we get to the S_i^+/ S_i^-, we average them.

As such, d_i^+ / d_i^- assume the average of all S_i^+/ S_i^- and they-re calculated from the average responses in K decision-makers:

- Arithmetic mean $d_i^{*+} = \frac{\sum_{k=1}^{K} d_i^{k+}}{K}$ and $d_i^{*-} = \frac{\sum_{k=1}^{K} d_i^{k-}}{K}$

- Geometric mean $d_i^{*+} = \sqrt[K]{\prod_{k=1}^{K} d_i^{k+}}$ and $d_i^{*-} = \sqrt[K]{\prod_{k=1}^{K} d_i^{k-}}$

Hands-on approach

- Let's try this on Google Sheets (same spreadsheet as before):

TOPSIS (Hwang & Yoon 1981)

1 Create a decision matrix Ai x Ci
1.1 Remeber to mark well the cost (negative) and benefit (positive) criteria:
Cost criteria impact negatively your model (risk, cost, danger, etc), benefit criteria impact positively (c

	Cost Criteria		Benefit Criteria		
Alternatives	C1 Cost	C2 Risk	C3 Flexibility	C4 Performance	C5 Reliability
A1	375	59	9	100	92
A2	354	64	2	84	89
A3	338	77	11	89	93
A4	217	67	20	85	84
W	0.3	0.2	0.05	0.3	0.15

1.2 Normalize the matrix using r_ij

	Cost Criteria		Benefit Criteria		
r_ij	C1 Cost	C2 Risk	C3 Flexibility	C4 Performance	C5 Reliability
A1	0.574	0.440	0.366	0.557	0.514
A2	0.542	0.477	0.081	0.468	0.497
A3	0.517	0.574	0.447	0.496	0.519
A4	0.332	0.499	0.812	0.474	0.469

2 Apply weights

	Cost Criteria		Benefit Criteria		
v_ij	C1 Cost	C2 Risk	C3 Flexibility	C4 Performance	C5 Reliability
A1	0.172	0.088	0.018	0.167	0.077
A2	0.162	0.095	0.004	0.140	0.075
A3	0.155	0.115	0.022	0.149	0.078
A4	0.100	0.100	0.041	0.142	0.070

Hands-on approach

- This is a free online TOPSIS tool
- **https://decision-radar.com/topsis**

Result

1. **A4** with score 0.74
2. **A1** with score 0.35
3. **A3** with score 0.29
4. **A2** with score 0.22

Show less

Normalized decision matrix: $\begin{pmatrix} 0.17 & 0.09 & 0.02 & 0.17 & 0.08 \\ 0.16 & 0.10 & 0.00 & 0.14 & 0.07 \\ 0.16 & 0.11 & 0.02 & 0.15 & 0.08 \\ 0.10 & 0.10 & 0.04 & 0.14 & 0.07 \end{pmatrix}$

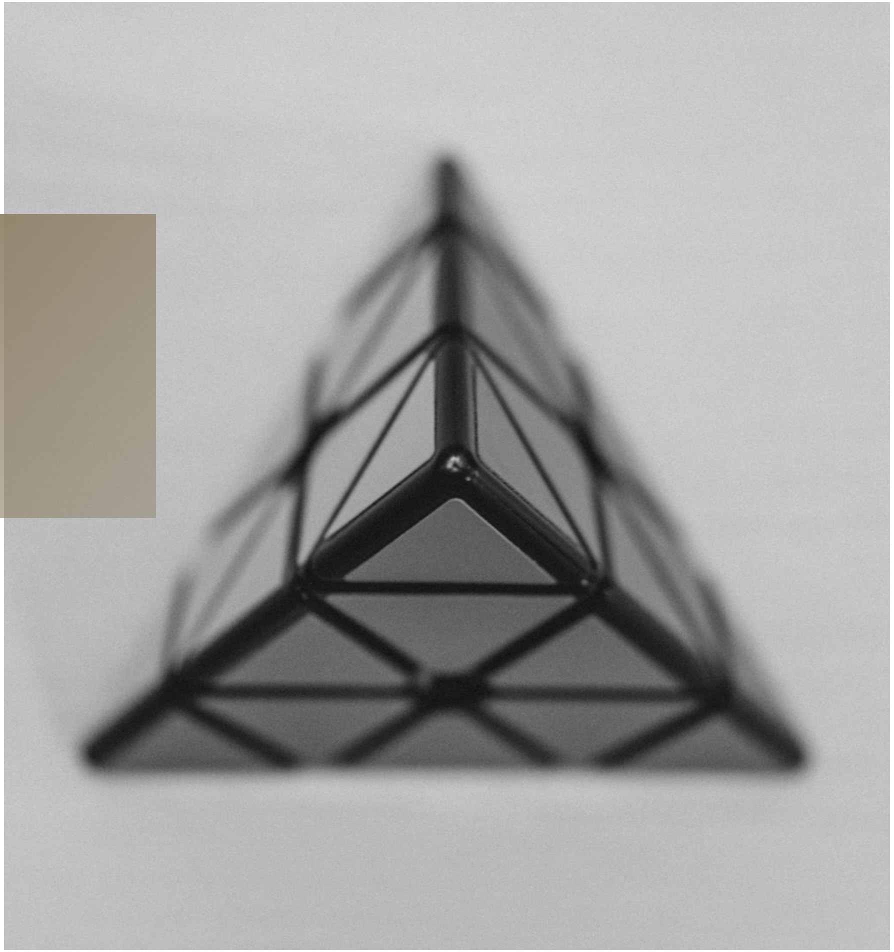
Best answer vector: $(0.10 \quad 0.09 \quad 0.04 \quad 0.17 \quad 0.08)$

Choices distance from best vector: $(0.08 \quad 0.08 \quad 0.07 \quad 0.03)$

Worst answer vector: $(0.17 \quad 0.11 \quad 0.00 \quad 0.14 \quad 0.07)$

Choices distance from worst vector: $(0.04 \quad 0.02 \quad 0.03 \quad 0.08)$

Closeness vector of each choices: $(0.35 \quad 0.22 \quad 0.29 \quad 0.74)$



FUZZY TOPSIS

Now it gets interesting

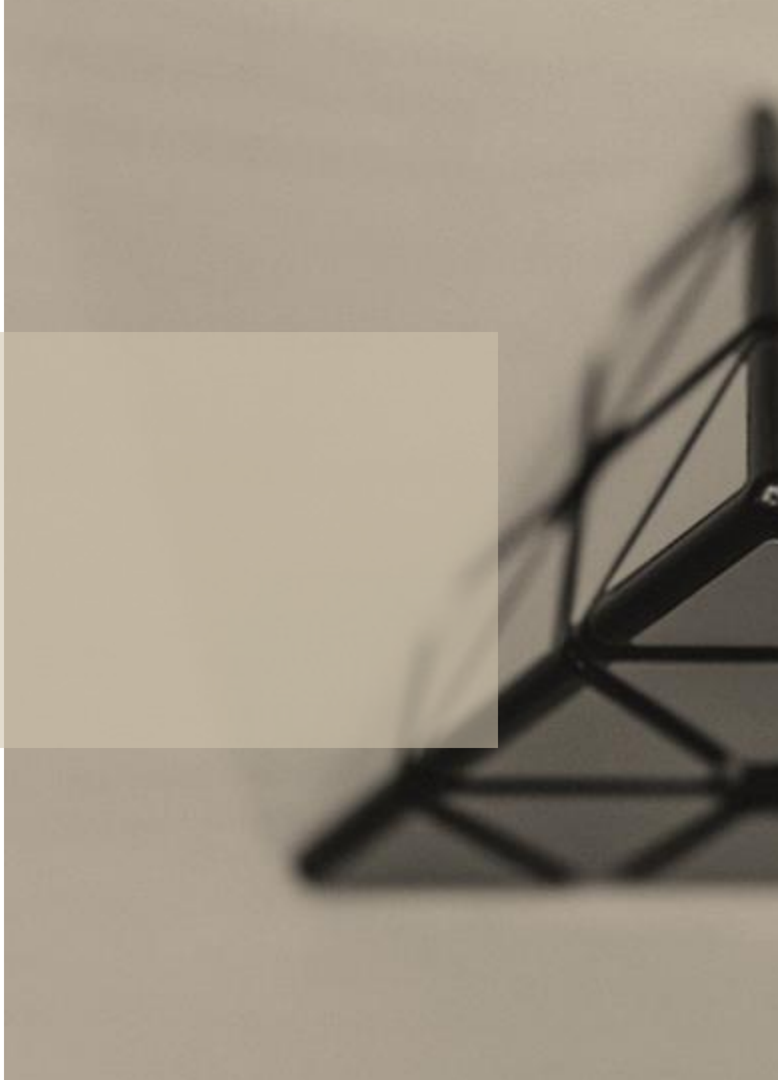
Fuzzy number + TOPSIS?

TOPSIS works best on raw or assigned data.

However, what if we need it to work on perceptions or qualitative data? Or data that we know are vague or imprecise?

Here’s where fuzzy numbers come to the rescue again.

I hope you still remember the basics of fuzzy numbers and operations from **04 Fuzzy AHP**.



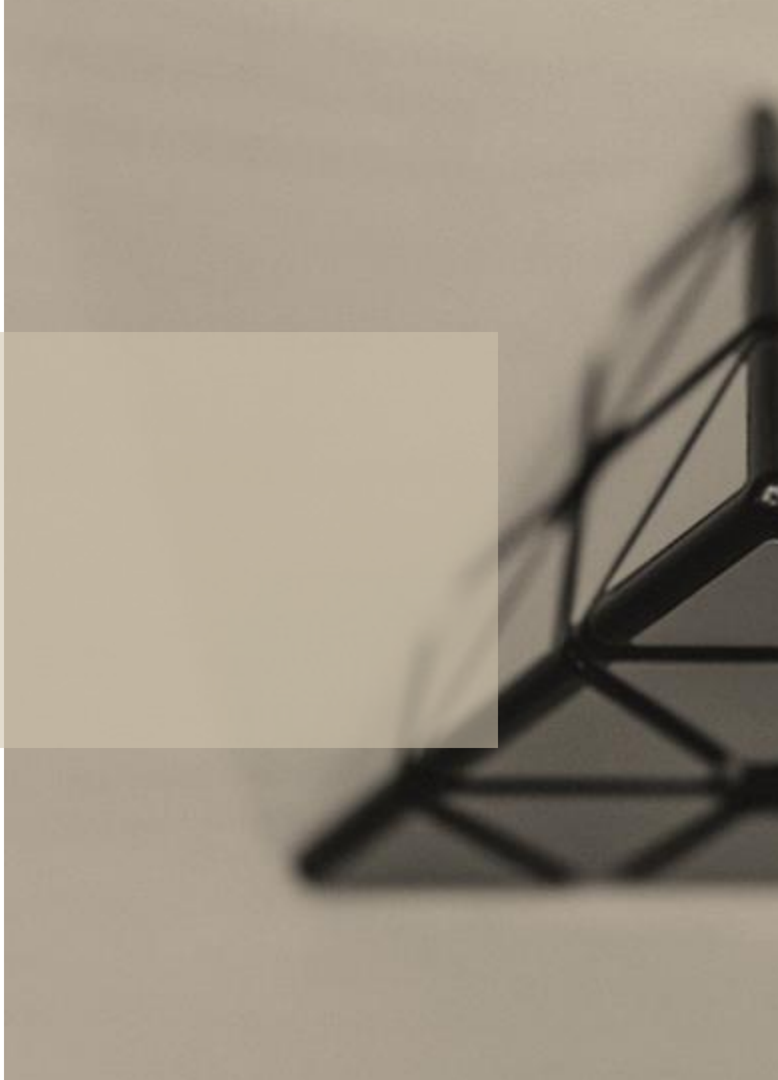
Fuzzy number + TOPSIS?

The implementation we are going to work with comes from Chen (2000).

In this implementation we use linguistic variables, translated to scales.

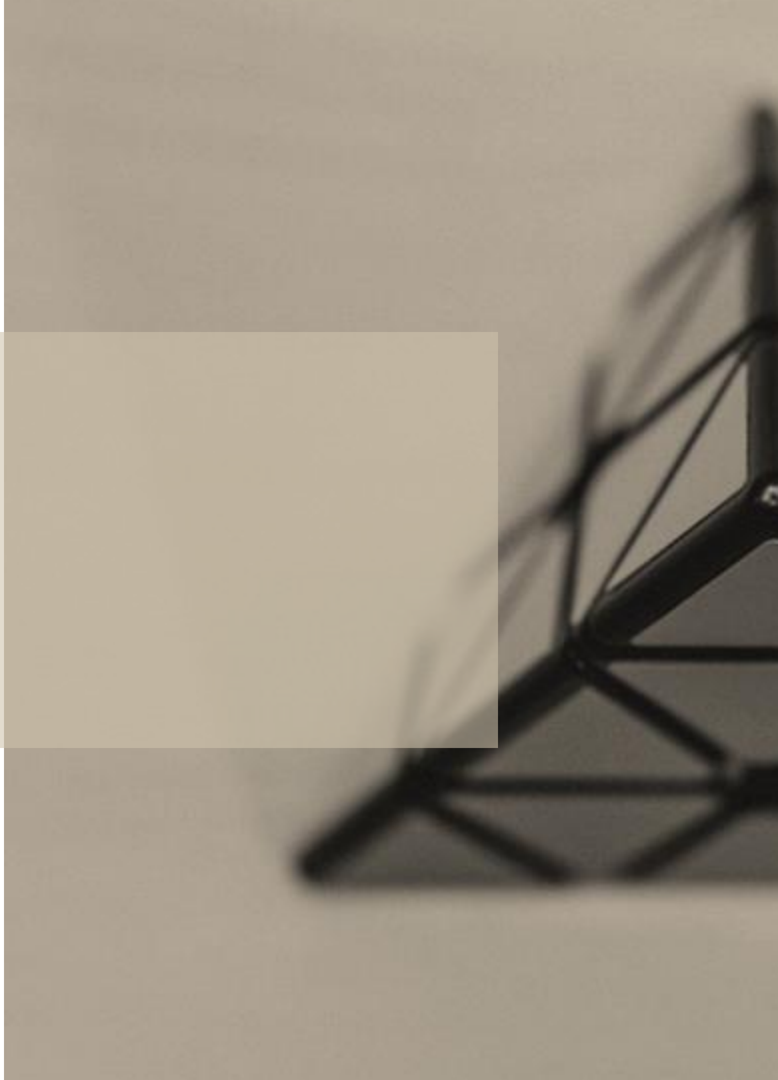
So now, we go back to restricting weights and importances to fixed scales (there are many alterations and refinements but we will stick to the one in Chen, 2000):

- **Linguistic Variables for Criteria Weights**
- **Linguistic Variables for Alternative Ratings**



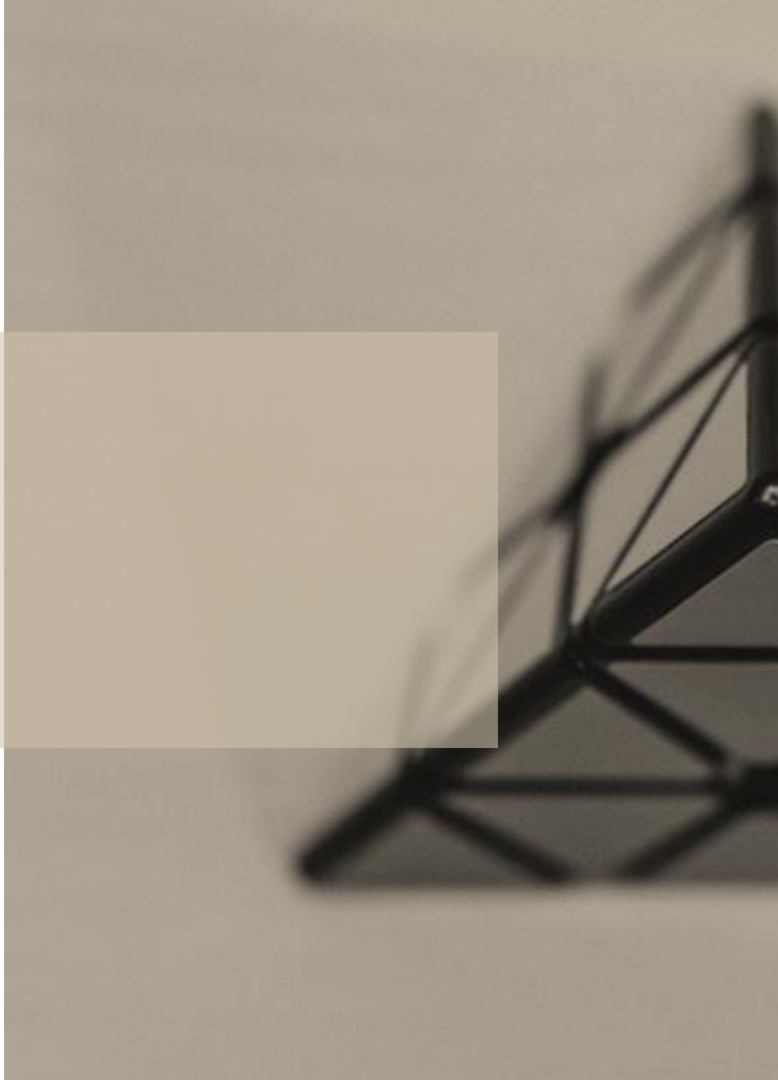
Criteria Weights

Linguistic Term	TFN Representation (Low, Medium, High)
Very Low (VL)	(0.0, 0.0, 0.1)
Low (L)	(0.0, 0.1, 0.3)
Medium Low (ML)	(0.1, 0.3, 0.5)
Medium (M)	(0.3, 0.5, 0.7)
Medium High (MH)	(0.5, 0.7, 0.9)
High (H)	(0.7, 0.9, 1.0)
Very High (VH)	(0.9, 1.0, 1.0)



Alternative Ratings

Linguistic Term	TFN Representation
Very Poor (VP)	(0, 0, 1)
Poor (P)	(0, 1, 3)
Medium Poor (MP)	(1, 3, 5)
Fair (F)	(3, 5, 7)
Medium Good (MG)	(5, 7, 9)
Good (G)	(7, 9, 10)
Very Good (VG)	(9, 10, 10)



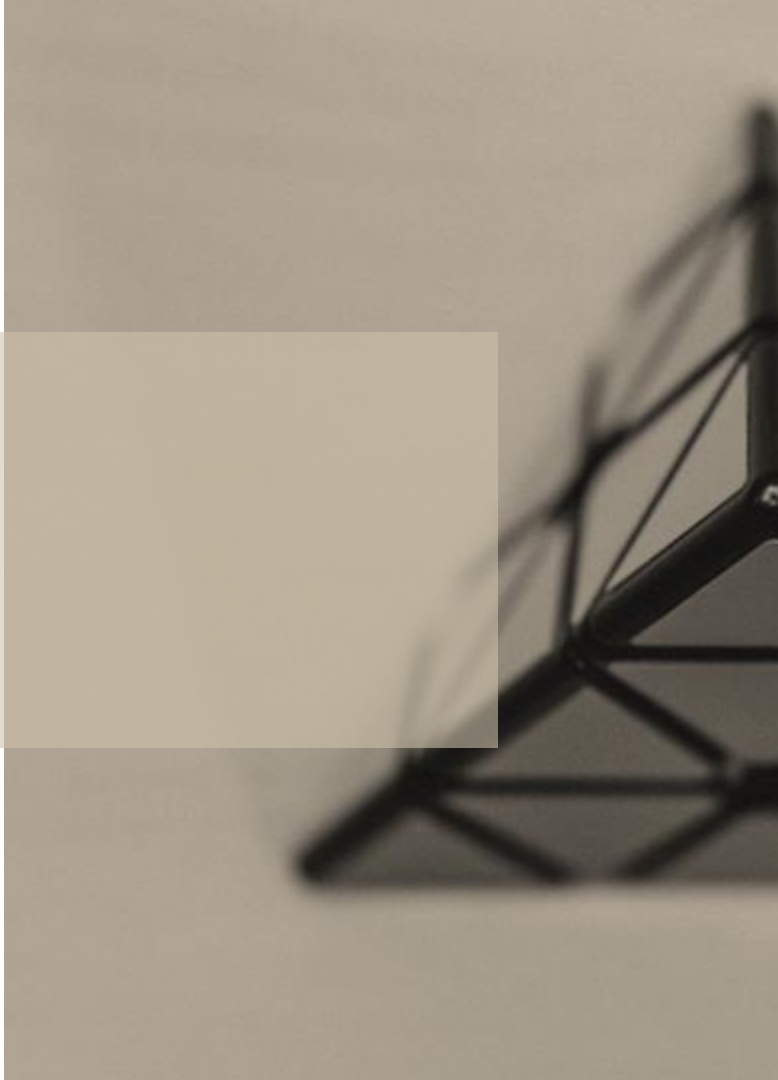
Alternative Ratings

TOPSIS works best on raw or assigned data.

However, what if we need it to work on perceptions or qualitative data? Or data that we know are vague or imprecise?

Here's where fuzzy numbers come to the rescue again.

I hope you still remember the basics of fuzzy numbers and operations from **04 Fuzzy AHP**.



Step 1

Build the decision matrix.

Remember that now any linguistic values are turned to Triangular Fuzzy Numbers (TFNs) according to the scale.

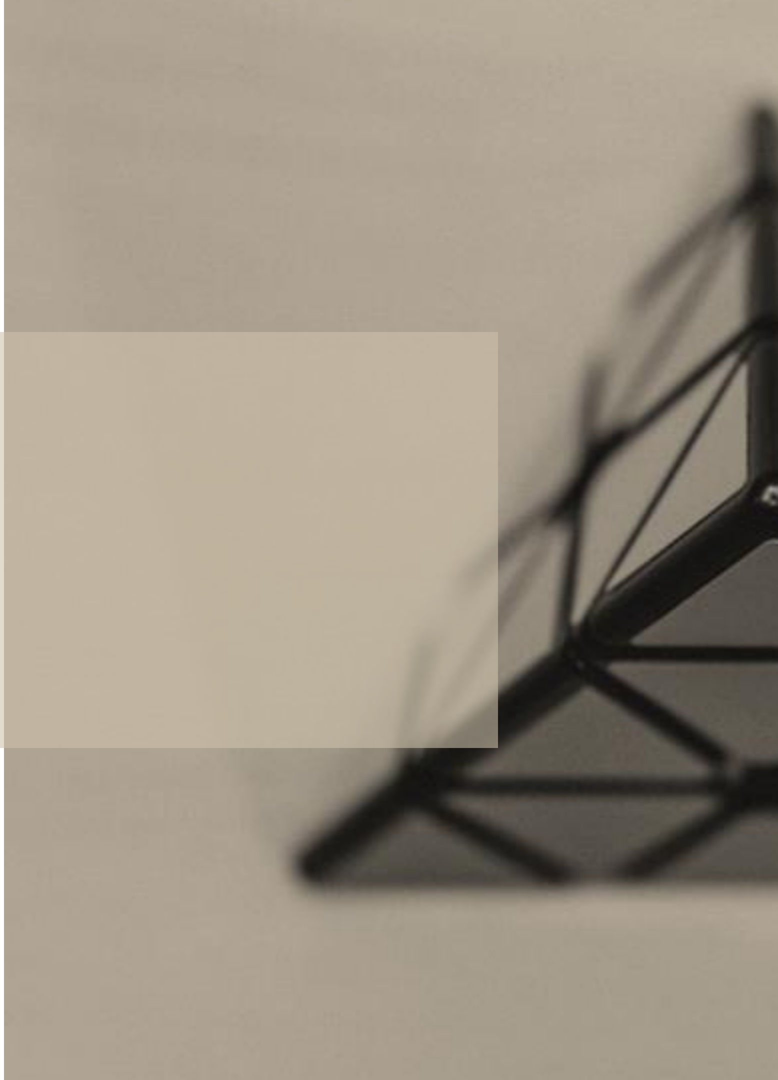
⚠️ Differently from crisp TOPSIS, the group decision aggregation happens now:

1. Build individual decision matrices
2. Average each pairwise $A_i \times X_i$ for all K decision-makers:

$$\tilde{x}ij = \frac{1}{K} \sum k = 1^K \tilde{x}_{ij}^{(k)}$$

3. Remember the operation:

$$\frac{1}{K}((a, b, c) \oplus (d, e, f)) = \frac{1}{K}(a + d, b + e\ , c + f)$$



Step 1

Build the decision matrix.

Remember that now any linguistic values are turned to Triangular Fuzzy Numbers (TFNs) according to the scale.

⚠️ Differently from crisp TOPSIS, the group decision aggregation happens now:

1. Build individual decision matrices
2. Average each pairwise $A_i \times X_i$ for all K decision-makers:

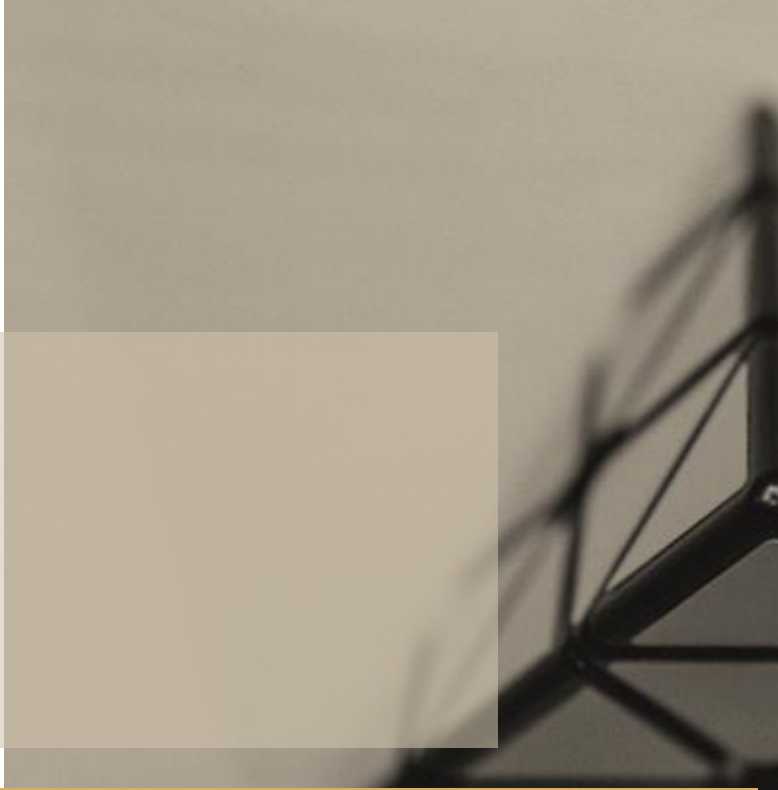
$$\tilde{x}ij = \frac{1}{K} \sum k = 1^K \tilde{x}_{ij}^{(k)}$$

alternative formula (not Chen's)

$$a_{ij} = \min_k \{a_{ij}^k\}, \quad b_{ij} = \frac{1}{K} \sum_{k=1}^{K} b_{ij}^k, \quad c_{ij} = \max_k \{c_{ij}^k\}$$

3. Remember the operation:

$$\frac{1}{K}((a, b, c) \oplus (d, e, f)) = \frac{1}{K}(a + d, b + e\,, c + f)$$



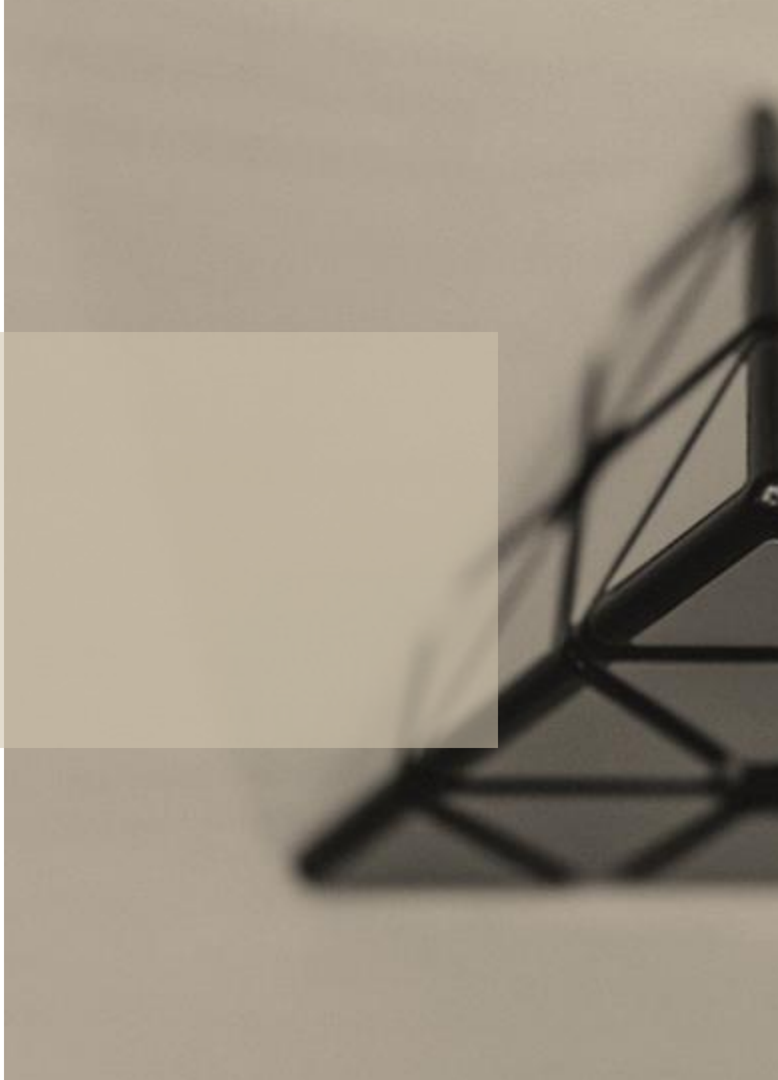
Step 2

Remember that now weights also obey a scale.

Given K DMs, the fuzzy weight $\widetilde{w}_j$ for each criterion is:

$$\tilde{w}j = \frac{1}{K}\sum k = 1^K \tilde{w}_j^{(k)}$$

1. Same thing, just remember how fuzzy addition works and you'll be fine



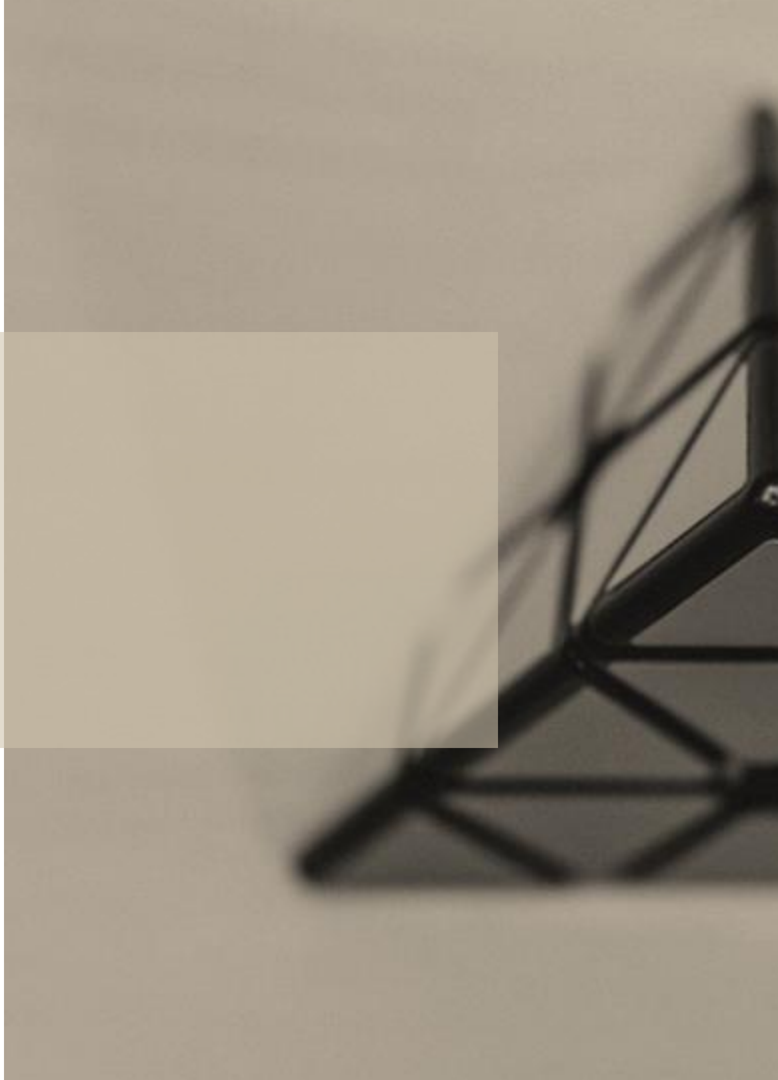
Step 3

Normalization here works in “mysterious ways” – we will Simply follow Chen’s algorithm...

1. Given a $\widetilde{D}$ decision matrix, each $\tilde{x}_{ij}$ is a TFN:

$$\tilde{D} = \begin{bmatrix} \tilde{x}11 & \tilde{x}12 & \cdots & \tilde{x}1n \\ \tilde{x}21 & \tilde{x}22 & \cdots & \tilde{x}2n \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{x}m1 & \tilde{x}m2 & \cdots & \tilde{x}_{mn} \end{bmatrix}$$

2. Normalization has two different formulae, one for benefit criteria (i.e., criteria that positively affect the decision), and another for cost criteria (the ones we need to minimize).



Step 3

The normalized fuzzy decision matrix $\tilde{R}$ is computed as:

1. For benefit criteria:

$$\tilde{r}_{ij} = \left(\frac{a_{ij}}{c_j^*}, \frac{b_{ij}}{c_j^*}, \frac{c_{ij}}{c_j^*} \right)$$

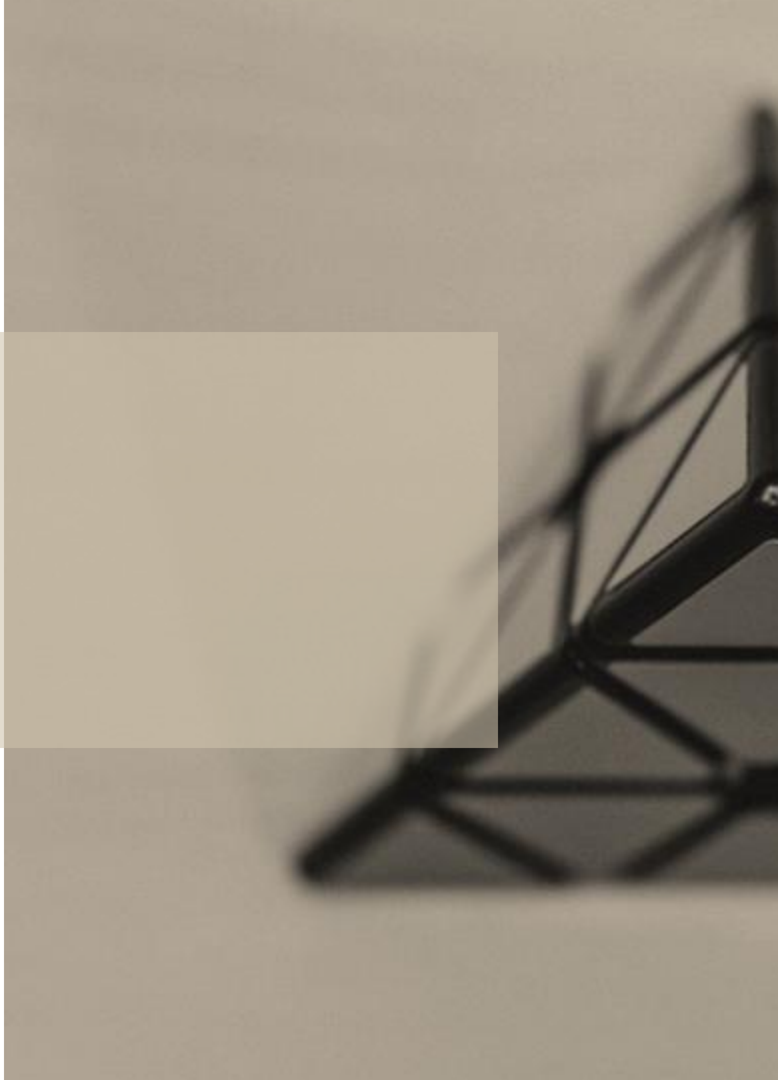
1. For cost criteria:

$$\tilde{r}_{ij} = \left(\frac{a_j^-}{c_{ij}}, \frac{a_j^-}{b_{ij}}, \frac{a_j^-}{a_{ij}} \right)$$

2. Where:

   1. $c_j^* = \max_i c_{ij}$ for benefit criteria

   2. $a_j^- = \min_i a_{ij}$ for cost criteria



Step 3

This is probably a good point to stop and remember that Chen uses a non-standard notation for TFN points:

- Most papers note TFNs as (l, m, u)
- Chen's paper uses (a, b, c)

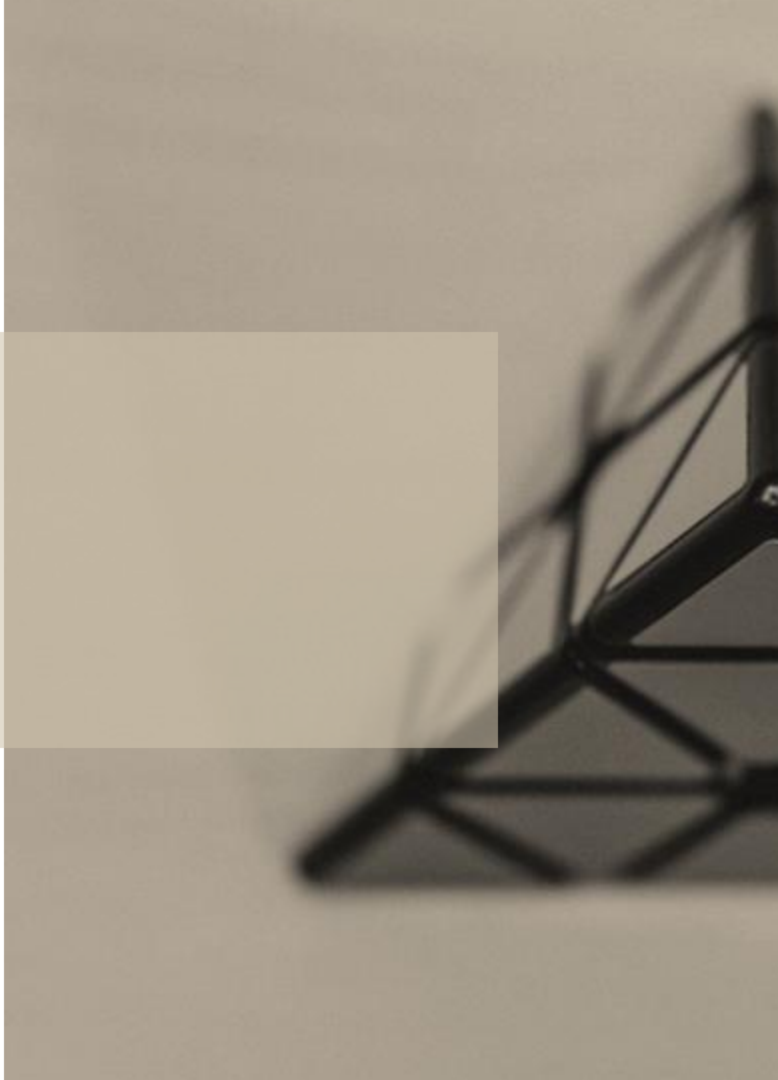
This means if we use the notation we saw in **04 Fuzzy AHP** is:

$$\tilde{r}ij = \left(\frac{lij}{c_j}, \frac{m_{ij}}{c_j}, \frac{u_{ij}}{c_j*} \right)$$

$$c_j^* = \max_i u_{ij}$$

$$\tilde{r}ij = \left(\frac{l_j^-}{uij}, \frac{l_j^-}{m_{ij}}, \frac{l_j^-}{l_{ij}} \right)$$

$$l_j^- = \min_i l_{ij}$$



Step 3

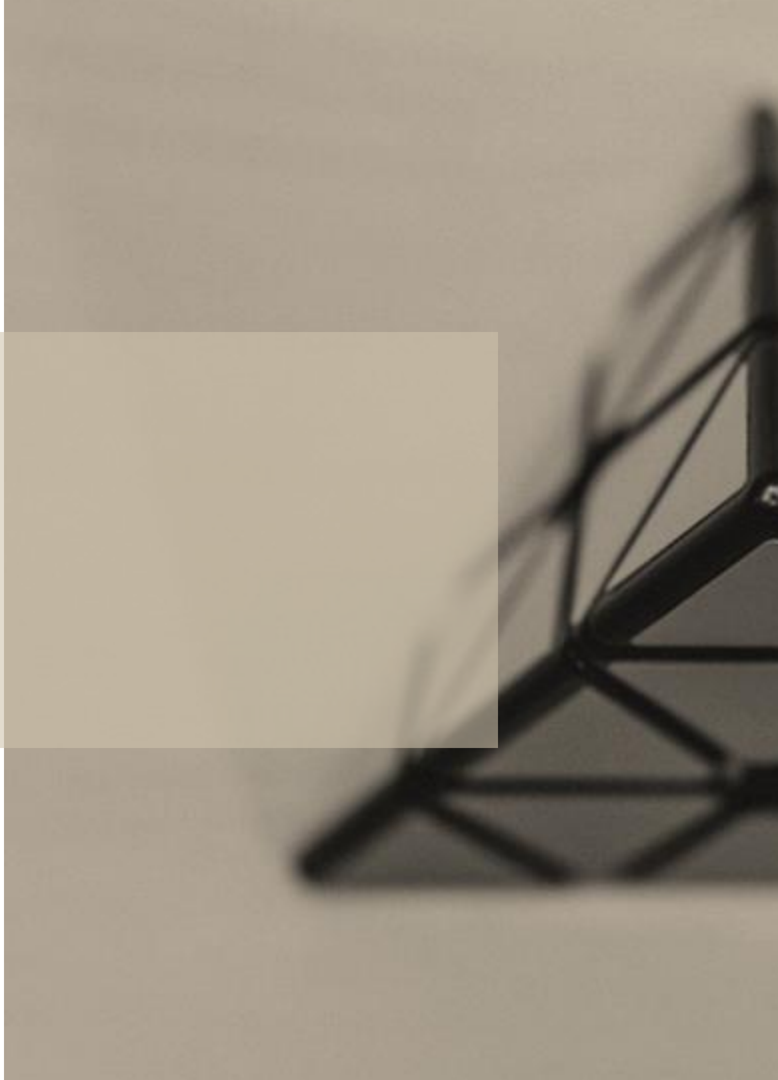
Example: Let’s take the following matrix, with (A_1, A_2, A_3) and (C_1, C_2):

$$\begin{array}{c} \qquad\qquad c_j^* \qquad\quad l_j^- \\ D = \begin{bmatrix} (3,5,\boxed{7}) & (\boxed{4},6,8) \\ (2,4,\boxed{6}) & (\boxed{3},5,7) \\ (5,7,\boxed{9}) & (\boxed{2},4,6) \end{bmatrix} \end{array}$$

We compute:

- For benefit criteria (C_1, C_2): Use maximum upper bound c_j^*.
- For cost criteria (*not present in this example*): Use minimum lower bound l_j^-.

- For C_1 : $c_1^* = \max(7,6,9) = 9$
- For C_2 : $c_2^* = \max(8,7,6) = 8$



Step 3

Applying the normalization formula:

$$\tilde{r}ij = \left(\frac{lij}{c_j^*}, \frac{m_{ij}}{c_j^*}, \frac{u_{ij}}{c_j^*}\right)$$

$$\tilde{r}ij = \left(\frac{lij}{\max_i u_{ij}}, \frac{m_{ij}}{\max_i u_{ij}}, \frac{u_{ij}}{\max_i u_{ij}}\right)$$

$$\begin{array}{cc} & \quad c_j^* \qquad\quad l_j^- \\ (l,m,u) & \\ D = & \begin{bmatrix} (3,5,7) & (4,6,8) \\ (2,4,6) & (3,5,7) \\ (5,7,9) & (2,4,6) \end{bmatrix} \end{array}$$

$$\tilde{r}ij = \left(\frac{lij}{c_j}, \frac{m_{ij}}{c_j}, \frac{u_{ij}}{c_j*}\right)$$

$$\tilde{r}ij = \left(\frac{l_j^-}{uij}, \frac{l_j^-}{m_{ij}}, \frac{l_j^-}{l_{ij}}\right)$$

For A_1 under C_1:

$$\tilde{r}_{11} = \left(\frac{3}{9}, \frac{5}{9}, \frac{7}{9}\right) = (0.33, 0.56, 0.78)$$

For A_2 under C_1:

$$\tilde{r}_{21} = \left(\frac{2}{9}, \frac{4}{9}, \frac{6}{9}\right) = (0.22, 0.44, 0.67)$$

For A_3 under C_1:

$$\tilde{r}_{31} = \left(\frac{5}{9}, \frac{7}{9}, \frac{9}{9}\right) = (0.56, 0.78, 1.00)$$

For A_1 under C_2:

$$\tilde{r}_{12} = \left(\frac{4}{8}, \frac{6}{8}, \frac{8}{8}\right) = (0.50, 0.75, 1.00)$$

For A_2 under C_2:

$$\tilde{r}_{22} = \left(\frac{3}{8}, \frac{5}{8}, \frac{7}{8}\right) = (0.38, 0.63, 0.88)$$

For A_3 under C_2:

$$\tilde{r}_{32} = \left(\frac{2}{8}, \frac{4}{8}, \frac{6}{8}\right) = (0.25, 0.50, 0.75)$$

Step 4

Compute the Weighted Normalized Matrix

Joining it all together:

$$R = \begin{bmatrix} (0.33, 0.56, 0.78) & (0.50, 0.75, 1.00) \\ (0.22, 0.44, 0.67) & (0.38, 0.63, 0.88) \\ (0.56, 0.78, 1.00) & (0.25, 0.50, 0.75) \end{bmatrix}$$

Notice now that all TFNs are monotonically ordered, and within the bounds of 0 ~ 1.

The weighted normalized fuzzy decision matrix $\tilde{V}$ is $\tilde{V} = [\tilde{v}_{ij}]$ where $\tilde{v}_{ij} = \tilde{r}_{ij} \cdot \tilde{w}_{ij}$.

That is, multiply this matrix by the aggregated weights….

Step 5

Now we need to find the best and worst scenarios.

Chen uses A^* and A^- (instead of V^+ or V^-) for the Fuzzy PIS and NIS (or FPIS and FNIS).

Here, the rule is much, much easier than in crisp TOPSIS (at least something had to right?):

- FPIS: $A^* = (1, 1, 1)$
- NPIS: $A^- = (0, 0, 0)$

Given A^* and A^- (or FPIS and FNIS), we need to compute the distance between two TFNs.

We use the following formula, to calculate the distance between any two TFNs:

$$d(\tilde{m}, \tilde{n}) = \sqrt{\frac{1}{3}\sum_{i=1}^{3}(m_i - n_i)^2} \quad \rightarrow \quad d(\tilde{m}, \tilde{n}) = \sqrt{\frac{1}{3}\left((l_m - l_n)^2 + (m_m - m_n)^2 + (u_m - u_n)^2\right)}$$

Step 6

The distance of each alternative A_i from the Fuzzy Positive Ideal Solution (FPIS) is:

$$d_i^* = \sum_{j=1}^{n} d(\tilde{v}_{ij}, \tilde{v}_j^*)$$

where:

- $d(\tilde{v}_{ij}, \tilde{v}j^*)$ is the distance between the weighted normalized fuzzy number $\tilde{v}ij$ and the FPIS value $\tilde{v}_j^*$.

- The FPIS is typically defined as $\tilde{v}_j^* = (1, 1, 1)$ for all criteria.

So for each alternative A_i :

1. Compute the distance from each fuzzy number $\tilde{v}_j^* = (1, 1, 1)$.
2. Sum up the distances over all criteria j.

Step 6

The distance of each alternative A_i from the Fuzzy Negative Ideal Solution (FNIS) is:

$$d_i^- = \sum_{j=1}^{n} d(\tilde{v}_{ij}, \tilde{v}_j^-)$$

where:

- $d(\tilde{v}_{ij}, \tilde{v}j^-)$ is the distance between the weighted normalized fuzzy number $\tilde{v}ij$ and the FNIS value $\tilde{v}_j^-$.
- The FNIS is typically defined as $\tilde{v}_j^- = (0, 0, 0)$ for all criteria.

So for each alternative A_i :

1. Compute the distance from each fuzzy number $\tilde{v}_j^- = (0, 0, 0)$.
2. Sum up the distances over all criteria j.

Step 6

Example: suppose we have one alternative A_1 and criteria (C_1, C_2), and that FPIS = (1,1,1):

$$V = \left[(0.5, 0.7, 0.9) \quad (0.3, 0.5, 0.7)\right]$$

For C_1: $d((0.5,0.7,0.9),(1,1,1)) = \sqrt{\frac{1}{3}((0.5-1)^2+(0.7-1)^2+(0.9-1)^2)} = \sqrt{\frac{1}{3}(0.25+0.09+0.01)} = \sqrt{\frac{0.35}{3}} \approx 0.34$

For C_2: $d((0.3,0.5,0.7),(1,1,1)) = \sqrt{\frac{1}{3}((0.3-1)^2+(0.5-1)^2+(0.7-1)^2)} = \sqrt{\frac{1}{3}(0.49+0.25+0.09)} = \sqrt{\frac{0.83}{3}} \approx 0.52$

Now, summing the distances, we obtain FPIS:

$$d_1^* = 0.34 + 0.52 = 0.86$$

Step 6

Example: suppose we have one alternative A_1 and criteria (C_1, C_2), and that NPIS = $(0,0,0)$:

For C_1: $d((0.5, 0.7, 0.9), (0, 0, 0)) = \sqrt{\frac{1}{3}((0.5-0)^2 + (0.7-0)^2 + (0.9-0)^2)} = \sqrt{\frac{1}{3}(0.25 + 0.49 + 0.81)} = \sqrt{\frac{1.55}{3}} \approx 0.72$

For C_2: $d((0.3, 0.5, 0.7), (0, 0, 0)) = \sqrt{\frac{1}{3}((0.3-0)^2 + (0.5-0)^2 + (0.7-0)^2)} = \sqrt{\frac{1}{3}(0.09 + 0.25 + 0.49)} = \sqrt{\frac{0.83}{3}} \approx 0.52$

Now, summing the distances, we obtain FNIS :

$$d_1^- = 0.72 + 0.52 = 1.24$$

Step 7

Now all we are left to do is compute the closeness coefficient that will gives the ranking.

$$CC_i = \frac{d_i^-}{d_i^+ + d_i^-}$$

Substituting the values:

$$CC_i = \frac{1.24}{0.86 + 1.24} \qquad CC_i = \frac{1.24}{2.10} \qquad CC_i \approx 0.590$$

Do the same for all values, order all CC_i and rank them.

Entropy method to elicit weights

The entropy method is a data-driven approach that uses the variability in the decision matrix to determine weights. It is useful when subjective judgments are not a

- 1. **Normalize the decision matrix.**
- 2. **Compute the entropy** E_j for each criterion X_j :

$$E_j = -k \sum_{i=1}^{m} p_{ij} \ln(p_{ij})$$

- where:

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}}, \quad k = \frac{1}{\ln(m)}$$

- 3. **Compute the degree of divergence**:

$$d_j = 1 - E_j$$

- 4. **Normalize the divergence values to obtain the weights**:

$$w_j = \frac{d_j}{\sum_{j=1}^{n} d_j}$$

APPENDIX 1 - ENTROPY

Entropy method to elicit weights

Example:

If the entropy values for three criteria are $E_1 = 0.2, E_2 = 0.5$ and $E_3 = 0.3$, the weights are computed as:

$$w_1 = \frac{1-0.2}{(1-0.2)+(1-0.5)+(1-0.3)} = \frac{0.8}{2.0} = 0.4$$

$$w_2 = \frac{0.5}{2.0} = 0.25, \quad w_3 = \frac{0.7}{2.0} = 0.35$$

(Now, you will probably go back to assigning arbitrary weights, right?)

APPENDIX 1 - ENTROPY

TIMELINE

Subject to change

03
Analytical Hierarchical Process (AHP)

04
Fuzzy AHP

05
TOPSIS &
Fuzzy TOPSIS

06
MACBETH

07
PAPRIKA

REFERENCES

REFERENCES

Today's content was mainly based on

- **Goodwin, P., & Wright, G. (2014). Decision analysis for management judgment. John Wiley & Sons. Belton, V., & Stewart, T. (2012). Multiple criteria decision analysis: an integrated approach. Springer Science & Business Media.**
- **Greco, S., Figueira, J., & Ehrgott, M. (Eds.). (2016). Multiple criteria decision analysis: state of the art surveys. New York, Springer.**
- **Shih, H. S., & Olson, D. L. (2022). TOPSIS and its extensions: A distance-based MCDM approach (Vol. 447). Springer Nature.**

REFERENCES

IMAGES

The image for PIS/NIS (TOPSIS Step 4) is from:

- **Chauhan, A., & Vaish, R. (2014). A comparative study on decision making methods with interval data. Journal of Computational Engineering, 2014(1), 793074.**

The images for Classical Visualization / Choice Behavior is from:

- **Shih, H. S., & Olson, D. L. (2022). TOPSIS and its extensions: A distance-based MCDM approach (Vol. 447). Springer Nature.**

THANKS

Does anyone have any questions?
Contact me at:

fellipe.martins@mackenzie.br
+11 95619 0585 (business hours)
fellipemartins.com.br