Difformity Index

What is a index of Difformity

It is a index that signals how much a ξ object of class k differs from the typical $\langle\xi_k\rangle$ object of its own class k. Through this approssimation, this index indicates how the object is difform from its own class.

Assumptions on the object

A ξ object

1. is characterised as a couple of vectors (i, p) suche that
    1. i are observable
    2. each i-trait is associated to a p_i score
        1. p_i are normalised such that $\sum p_{i,\xi} = 1$
2. ξ is assigned to one and only one k class
    1. This relation is fixed over time and it never changes

Assumptions on the typical object of the class

The k class

1. Has a typical object $\langle\xi_k\rangle$
2. $\langle\xi_k\rangle$ can be individued through many methods
    1. It could hold the approximation that $\langle\xi_k\rangle$ is identified in that couple of vectors $(i_{\langle\xi_k\rangle}, p_{\langle\xi_k\rangle})$ where
        1. All i traits observed in ξ elements of class k are $i_{\langle\xi_k\rangle}$ traits of $\langle\xi_k\rangle$

    2. $p_{i,\langle\xi_k\rangle}$ satifies the conditions to be apportioned to a average value of $p_{i,\xi\in k}$
    3. Still holds that $\sum p_{i,\langle\xi_k\rangle} = 1$
2. Or $\langle\xi_k\rangle$ can be defined alternatively as any couple of vectors $(i_{\langle\xi_k\rangle}, p_{\langle\xi_k\rangle})$ as long it holds $\sum p_{i,\langle\xi_k\rangle} = 1$

Under these conditions, to simplify notations, it is assumed that

- $(i_{\langle\xi_k\rangle}, p_{\langle\xi_k\rangle})$ can be recalled directly as (i_k, p_k)

Operation

A ξ object always a k class. So it always exist

- a vector of i_ξ traits of the object
- a vector of i_k traits of the class of the object

A first operation is to identify the set $C_\xi = \cup(i_\xi, i_k)$

Difformity is defined as follow:

$$\sum_{i,j\in C_\xi} (|p_{i,k} - p_{i,\xi}|) \cdot (|p_{j,k} - p_{j,\xi}|) \cdot \frac{d_{i,j}}{2}$$

where $d_{i,j}$ is a measure of dissimilarity between the trait i and the trait j, such that $d_{i,i} = 0$.

Homeomorphism with (Inter)Disciplinarity

- Papers are objects
- Journals are classes
- Disciplines are traits
    - Scores are allocations of disciplinarity

This is not the only possible homemorphism. For example, papers can be still the objects (analytical units), then authors can be the classes.

More radically, instead of assuming that papers are the objects, it is possible to assume that authors are objects and their departments in a specific fixed time point are their class.

Notes on the homeomorphism

- Disciplines are defined as observable traits, not as classes. In reality they act more as labels. The origin of this labelisation (i.e. the classification scheme) is not trivial.
- Canonically, disciplinary labelisation is operated through the reference lists of papers. This operation is understood as an observation of what the paper is intending to cognitively integrate.
  - Nevertheless, a very similar operation can be performed over the semantics of the paper, and it would still catch how the paper is trying to integrate disparate sources of concepts, data, theories, etc.

Example

- Let have a collection of two papers from the same journal. Let call the papers "Paper 1" and "Paper 2".
- The journal is called "International Journal of Pomponomics" or just "IJP".
- According to HAL GPT, a friendly Artificial Intelligence, it is fair to say that a nice representation of what IJP deals with is the following:
  - Pomponomics: 66%
  - Junkology: 25%
  - Enneagrammatics: 9%

```
ex_Journals <- tibble(
  level = "k",
  k_Name = "IJP",
  Discipline =
    c("Pomponomics",
      "Junkology",
      "Enneagrammatics"),
  p = c(.66,.25,.09)
)

ex_Journals
```

```
# A tibble: 3 x 4
  level k_Name Discipline          p
  <chr> <chr>  <chr>           <dbl>
```

```
1 k     IJP    Pomponomics      0.66
2 k     IJP    Junkology        0.25
3 k     IJP    Enneagrammatics  0.09
```

Let's see now how HAL GPT will label the papers:

```
ex_Papers <- tibble(
  level = "xi",
  xi_Name = "Paper 1",
  Discipline =
    c("Pomponomics",
      "Junkology"),
  p = c(.5,.5),
  k_Name = "IJP"
) %>%
  add_row(
    tibble(
      level = "xi",
  xi_Name = "Paper 2",
  Discipline =
    c("Pomponomics",
      "Junkology",
      "Enneagrammatics",
      "Science of Science"),
  p = c(.25,.25,.25,.25),
  k_Name = "IJP"
)
  )

ex_Papers
```

```
# A tibble: 6 x 5
  level xi_Name Discipline             p k_Name
  <chr> <chr>   <chr>              <dbl> <chr>
1 xi    Paper 1 Pomponomics         0.5  IJP
2 xi    Paper 1 Junkology           0.5  IJP
3 xi    Paper 2 Pomponomics         0.25 IJP
4 xi    Paper 2 Junkology           0.25 IJP
5 xi    Paper 2 Enneagrammatics     0.25 IJP
6 xi    Paper 2 Science of Science  0.25 IJP
```

A convenient way to store all this information is the following long format:

```
ex_Papers %>%
  add_row(ex_Journals %>% add_column(xi_Name = "Paper 1")) %>%
  add_row(ex_Journals %>% add_column(xi_Name = "Paper 2")) -> ex_Papers

  ex_Papers %>% arrange(xi_Name)
```

```
# A tibble: 12 x 5
   level xi_Name Discipline             p k_Name
   <chr> <chr>   <chr>              <dbl> <chr>
 1 xi    Paper 1 Pomponomics         0.5  IJP
 2 xi    Paper 1 Junkology           0.5  IJP
 3 k     Paper 1 Pomponomics         0.66 IJP
 4 k     Paper 1 Junkology           0.25 IJP
 5 k     Paper 1 Enneagrammatics     0.09 IJP
 6 xi    Paper 2 Pomponomics         0.25 IJP
 7 xi    Paper 2 Junkology           0.25 IJP
 8 xi    Paper 2 Enneagrammatics     0.25 IJP
 9 xi    Paper 2 Science of Science  0.25 IJP
10 k     Paper 2 Pomponomics         0.66 IJP
11 k     Paper 2 Junkology           0.25 IJP
12 k     Paper 2 Enneagrammatics     0.09 IJP
```

The dataset is now re-formatted again to make it easier a subtraction

```
ex_Papers %>%
  pivot_wider(names_from = level,
              values_from = p,
              values_fill = 0) -> ex_Papers

ex_Papers %>%
  mutate(diff = abs(k - xi)) -> results
results
```

```
# A tibble: 7 x 6
  xi_Name Discipline         k_Name    xi     k  diff
  <chr>   <chr>              <chr>  <dbl> <dbl> <dbl>
1 Paper 1 Pomponomics        IJP     0.5   0.66  0.16
2 Paper 1 Junkology          IJP     0.5   0.25  0.25
3 Paper 2 Pomponomics        IJP     0.25  0.66  0.41
4 Paper 2 Junkology          IJP     0.25  0.25  0
5 Paper 2 Enneagrammatics    IJP     0.25  0.09  0.16
```

```
6 Paper 2 Science of Science IJP     0.25  0     0.25
7 Paper 1 Enneagrammatics    IJP     0     0.09  0.09
```

Final operations

Dissimilarity matrix

In literature, the following matrix of dissimilarities is found (actually not a matrix in this format):

```
d = crossing(i = ex_Papers$Discipline %>% unique(),
             j = ex_Papers$Discipline %>% unique()) %>%
  mutate(d = ifelse(i == j,
                    0,
                    .5
                    ))
d
```

```
# A tibble: 16 x 3
   i                  j                      d
   <chr>              <chr>              <dbl>
 1 Enneagrammatics    Enneagrammatics      0  
 2 Enneagrammatics    Junkology            0.5
 3 Enneagrammatics    Pomponomics          0.5
 4 Enneagrammatics    Science of Science   0.5
 5 Junkology          Enneagrammatics      0.5
 6 Junkology          Junkology            0  
 7 Junkology          Pomponomics          0.5
 8 Junkology          Science of Science   0.5
 9 Pomponomics        Enneagrammatics      0.5
10 Pomponomics        Junkology            0.5
11 Pomponomics        Pomponomics          0  
12 Pomponomics        Science of Science   0.5
13 Science of Science Enneagrammatics      0.5
14 Science of Science Junkology            0.5
15 Science of Science Pomponomics          0.5
16 Science of Science Science of Science   0  
```

Difformity

```r
results %>%
  transmute(xi_Name,
            Discipline) %>%
  group_by(xi_Name) %>%
  summarize(crossed = list(crossing(Discipline, Discipline))) %>%
  unnest(crossed) %>%
  rename(
    i = Discipline...1,
    j = Discipline...2
  ) %>%
  left_join(
    results %>% transmute(
      xi_Name,
      i = Discipline,
      diff_i = diff
    )
  ) %>%
    left_join(
    results %>% transmute(
      xi_Name,
      j = Discipline,
      diff_j = diff
  )
    ) %>%
  left_join(
    d
  ) -> final_tibble
```

```
Joining with `by = join_by(xi_Name, i)`
Joining with `by = join_by(xi_Name, j)`
Joining with `by = join_by(i, j)`
```

```r
final_tibble
```

```
# A tibble: 25 x 6
   xi_Name i               j               diff_i diff_j     d
   <chr>   <chr>           <chr>            <dbl>  <dbl> <dbl>
 1 Paper 1 Enneagrammatics Enneagrammatics   0.09   0.09    0
 2 Paper 1 Enneagrammatics Junkology         0.09   0.25   0.5
 3 Paper 1 Enneagrammatics Pomponomics       0.09   0.16   0.5
 4 Paper 1 Junkology       Enneagrammatics   0.25   0.09   0.5
 5 Paper 1 Junkology       Junkology         0.25   0.25   0
 6 Paper 1 Junkology       Pomponomics       0.25   0.16   0.5
 7 Paper 1 Pomponomics     Enneagrammatics   0.16   0.09   0.5
 8 Paper 1 Pomponomics     Junkology         0.16   0.25   0.5
 9 Paper 1 Pomponomics     Pomponomics       0.16   0.16   0
10 Paper 2 Enneagrammatics Enneagrammatics   0.16   0.16   0
# ... with 15 more rows
```

Finally we can measure the difformity for the two papers!

```
final_tibble %>%
  summarise(
    Difformity = sum(diff_i * diff_j * d/2),
    .by = "xi_Name"
  ) %>%
  mutate(
    True_Difformity =
      1 / (1 - Difformity)
  )
```

```
# A tibble: 2 x 3
  xi_Name Difformity True_Difformity
  <chr>        <dbl>           <dbl>
1 Paper 1     0.0385            1.04
2 Paper 2     0.104             1.12
```

Criticisms

Mathematical concerns

$p \mid \sum p = 1$ makes sense as a noteworthy statistical distribution.

Instead, $|p_k - p_\xi|$ is statistically complex and may need further corrections to be adjusted into a True measure. For example,

- while Rao-Stirling only converges to 1
- Difformity can reach 1 when
  - object and class have no traits in common,

  - all of their traits have $d_{ij} = 1$

- In this case True Difformity is Infinite.

Theoretical concerns

The index is very similar to Rao, however in the Stirling's interpretation of Rao, Rao compunds three factors: Richness, Balance and Disparity.

These factors make no sense for the index of Difformity. Not only in semantics, but directly in the mathematical proprieties of the index.

For example, let's imagine the case for

- Richness being minimal,
- while Balance and Disparity being maximal.

This is the case for ξ holding only 2 traits

- $i_1 = .5$
- $i_2 = .5$
- $d_{i_1,i_2} = 1$

In this case, $D_{RS} = .5$, which means that

- since $\lim RS = 1$
- then Richness counts for half of the whole index. at these conditions

For Difformity, minimising Richness while maximising Disparity requires has 2 interpretations:

- Since richness is minimal, k and ξ have only one trait, and they share it. By imposing $\sum p = 1$, then Balance is necessarily maximal and Difformity will be minimal at 0, independently by Disparity.
- k and ξ have only one trait, but these are two different traits (i and j), and very dissimilar (Disparity is maximal). Balance is again forced to be maximal can be maximal by $p_i = p_j = 1$. In this case, Difformity will be maximal at 1.

So it is demonstrated that in its current definition, Difformity is independent by Richness.

Solution

Index of difformity can be forced to behave exactly as Rao-Stirling with a re-normalisation

$$\phi_{i,(k,\xi)} = \frac{|p_{i,k} - p_{i,\xi}|}{\sum_i |p_{i,k} - p_{i,\xi}|}$$

$$\text{Difformity} = \sum_{i,j \in C_\xi} \phi_i \cdot \phi_j \cdot d_{i,j}$$

Forcing that $\phi = 0$ when $\sum_i |p_{i,k} - p_{i,\xi}| = 0$

```
results %>%
  mutate(phi = diff/sum(diff),
         .by = "xi_Name") -> results

results %>%
  transmute(xi_Name,
            Discipline) %>%
  group_by(xi_Name) %>%
  summarize(crossed = list(crossing(Discipline, Discipline))) %>%
  unnest(crossed) %>%
  rename(
    i = Discipline...1,
    j = Discipline...2
  ) %>%
  left_join(
    results %>% transmute(
      xi_Name,
      i = Discipline,
      diff_i = diff,
      phi_i = phi
    )
  ) %>%
    left_join(
    results %>% transmute(
      xi_Name,
      j = Discipline,
      diff_j = diff,
      phi_j = phi
  )
    ) %>%
  left_join(
    d
  ) -> final_tibble
```

```
Joining with `by = join_by(xi_Name, i)`
Joining with `by = join_by(xi_Name, j)`
Joining with `by = join_by(i, j)`
```

```
final_tibble %>%
  summarise(
    Difformity = sum(diff_i * diff_j * d/2),
    Difformity_alt = sum(phi_i * phi_j * d),
    .by = "xi_Name"
  ) %>%
  mutate(
    True_Difformity =
      1 / (1 - Difformity),
    True_Difformity_alt =
      1 / (1 - Difformity_alt)
  )
```

```
# A tibble: 2 x 5
  xi_Name Difformity Difformity_alt True_Difformity True_Difformity_alt
  <chr>        <dbl>          <dbl>           <dbl>               <dbl>
1 Paper 1     0.0385          0.308            1.04                1.44
2 Paper 2     0.104           0.309            1.12                1.45
```

Methodological concerns

Let's assume the case for a journal characterised by only one discipline: Political Economics.

In this journal there is a paper characterised by only one discipline: Economical Politics.

In reality, the two are the same things. Yet without accounting for Disparity, the index would signal the maximal Difformity. However, if $d = 0$ as it should be, then Difformity is 0.

In other terms, the index is highly sensitive to d.