York University – Glendon Campus

Faculty of Liberal Arts and Professional Studies

Data Mapping: Finding the Relation Between Two Disciplines (Outline)

Peter Gemayel

Thursday August 22, 2019

Abstract: There is no one universal way for finding a relation between two disciplines. This paper aims to test one method on two areas of study taught at York University Glendon Campus, namely International Studies and Philosophy. A detailed scheme that goes through the analysis of how one could attain results and show through running a χ^2 – test how the two disciplines are related by rejecting or accepting the null hypothesis in in favour of the alternate hypothesis.

Index

I – Theory

Interdisciplinarity is the combination of two or more disciplines and crossing the boundaries set by the traditional schools of thought to create a new discipline that better serves the expanding needs of society. Ever since the 18^{th} century, universities at higher levels have put an emphasis on research and academic freedom which has led to phenomenal development in science and society. Since then, professionals of various educational backgrounds come together to combine their perspectives to solve some of the world's most complex problems, such as the global warming epidemic.

Consider how Calculus has significantly changed the field of Business. When Sir Isaac Newton first founded Calculus, it was to answer questions about how objects on the macro scale work. Fast forward to the present and observe how businesses have integrated Calculus into their models to maximize profits. This is just one of the many examples of how professionals worldwide collaborate to make the world we live in simpler.

To the best of my knowledge, this is the first project that attempts to build a universal to find correlations between two disciplines. The aim of this report is to prove that by creating a list of pairs of all possible keywords collected from two disciplines, one can run a χ-Square test to prove whether the two disciplines are dependent or independent of one another.

As my primary goal is to attempt to build a mathematical background for this project and a supporting algorithm, this paper will serve more as an outline for future progress in this field.

II – Procedure

The goal is to bring disciplines taught at York University, Glendon Campus to investigate relations among them to further enhance the student learning experience.

To obtain the results, two disciplines are selected from those taught at the university. After selecting which two disciplines to work with, a list of professors from each of the two departments can be obtained from the university website and then use google scholar and other academic search engines to access the articles by each professor and strip the keywords from those articles.

For this report, the two disciplines selected are International Studies and Philosophy. Since many professors from these fields did not have their published articles uploaded to the university website, many search engines had to be consulted. The articles were tabulated in such a way as to record as much information as possible with the basic structure shown in the table below.

Table 1 – Format used to gather data on Excel.

First Name	Last Name	Department	Article Title	Journal Published	Citations	Keywords	JEL Classifications

Since not all professors have used JEL Classifications in their articles, the last column in table 1 can be ignored for the purpose of this paper.

To get the results, a search engine was used to go through the articles and check how many times a keyword pair from the list generated would occur. One search engine that is accessible with ease is https://www.aminer.org/ which contains large packages of approximately 270 million publications.

A script for the above procedure has been written in Python and is shown in the appendix and an excel file for the raw data is attached separately.

III – Results

Let the keywords obtained from articles by professors in the International Studies department be determined by

$$\text{Keyword \#1} = \{Y_1, Y_2, \ldots, Y_n\}$$

and those for Philosophy by

$$\text{Keyword \#2} = \{X_1, X_2, \ldots, X_k\}.$$

Then pair one element in Keyword #1 with all elements in Keyword #2 to create a list of all possible keyword pairings which can then be displayed in a matrix format. This table is to be filled by searching every pair of keywords in the academic search engine.

Table 2 – Table showing all pairs of keywords and the number of articles related to them (N_{i_j}).

	Y_1	Y_2	Y_3	…	Y_n
X_1	N_{1_1}	N_{1_2}	N_{1_3}	…	N_{1_n}
X_2	N_{2_1}	N_{2_2}	N_{2_3}	…	N_{2_n}
X_3	N_{3_1}	N_{3_2}	N_{3_3}	…	N_{3_n}
…	…	…	…	…	…
X_k	N_{k_1}	N_{k_2}	N_{k_3}	…	N_{k_n}

One way to analyze categorical data is to use a χ^2- test to check whether the variables are dependent or independent.

$$\chi^2 = \Sigma_i \frac{(O_i - E_i)^2}{E_i} \text{ for i = 1,2,3…} \quad (1)$$

where O_i is the observed quantity and E_i is the expected quantity which is calculated in the following way

$$E_1 = \frac{\Sigma_j Y_j X_1 \times \Sigma_i X_i Y_1}{\Sigma_{ij} X_i Y_j} \text{ for i,j = 1,2,3…} \quad (2)$$

Similarly, for E_2, E_3, ...

The degrees of freedom (DoF)

$$DoF = (\#rows - 1) \times (\#columns - 1) \qquad (3)$$

which represent the least number of independent coordinates that can specify the position of the system completely is found to be 4004. The significance level is set at $\alpha = 0.05$ means that there is only a 5 % chance of making a Type I error.

Let H_0 be the null hypothesis that and H_A be the alternative hypothesis, such that

H_0 = the two disciplines are independent.

H_A = the two disciplines are dependent.

The p-value is then determined from a statistics table after finding the χ^2 value.

If the p-value,

i) Is less than or equal to α then H_A is favoured.
ii) Is greater than α then H_0 is favoured.

Due to time limit, difficulty accessing publications and limited access to powerful computers, the analysis has not been conducted on concrete data.

IV – Conclusion

Once the results are obtained, professionals from both areas of study will come together and explore the commonalities based on the results generated from this process.

The specific areas to search for commonalities can also arise in the pattern in which keyword-pairs occurrences happen in subfields of the disciplines in question.

If the above method proves to be precise, then this could be applied to every other discipline.

V – Appendix

Code to generate formatted tables and get results (written in Python):

```
import pandas as pd
import numpy as np
import ijson
from io import StringIO
def convert_excel_file_to_list(filename):
    """
    Takes in a file name.
    Outputs a list of pairs, containing every pair of "International" and "Philosophy" keywords
    """
    df = pd.read_csv(filename, header=0, keep_default_na=False)
    column_status = "Department"        # column that has International/Philosophy tag
    column_prefix_keyword = "Key word"  # column prefixes that contain keywords
    set_international = set()           # will contain all international keywords
    set_philosophy = set()              # will contain all philosophy keywords
    # get all column names for columns that represent keywords: e.g. Key word #1, Key word #2, ...
    keyword_columns = [x for x in df.columns.values if "Key word" in x]
    # loop through table
    for index, row in df.iterrows():
        # get the status: International studies or Philosophy
        status = row[column_status]
        # go through all key word columns and collect key words
        for kc in keyword_columns:
            # skip empty keywords
            if str(row[kc]).strip() == "_":
                continue
            # if keyword belongs to international studies, put it in that set
            if status == "International Studies":
                set_international.add(row[kc])
            # otherwise, put it in the philosophy set
            else:
                set_philosophy.add(row[kc])
    # for EACH international key word, go through all philosophy keywrods and put the two as a pair in the list
    list_pairs = [[i, j] for i in set_international for j in set_philosophy]
    # E.g. list_pairs[0] is the first pair, list_pairs[1] is the second pair, etc..
    # list_pairs[0][0] is the international keyword of the first pair
    # list_pairs[0][1] is the philosophy keyword of the first pair
    # etc...
    return list_pairs
```

```
data_lists = convert_excel_file_to_list("data.csv")
# print all pairs separated by a comma
for pair in data_lists:
   print ("{}, {}".format(pair[0], pair[1]))
# put data list in a formatted table
df = pd.DataFrame(data_lists)
df.columns = ["International", "Philosophy"]
column_names = sorted(list(set(df["International"])))
row_names = sorted(list(set(df["Philosophy"])))
print (len(column_names), len(row_names))
# ;limit
column_names = column_names
row_names = row_names
df = pd.DataFrame(columns=column_names, index=row_names)
df.head()
def load_json_2(filename):
    # call load_json_2 and it will give you a list of all the keywords per ar
ticle
    with open(filename, "r") as f:
        keywords = list()
        for line in f:
            stream = StringIO(line)
            parser = ijson.parse(stream)
            for prefix, event, value in parser:
                if (prefix, event) == ("keywords", "start_array"):
                    keywords.append(list())  # create new list for article
                if (prefix, event) == ("keywords.item", "string"):
                    keywords[-1].append(value)
    return keywords
keywords = load_json_2(r"File location")
print(keywords[0])  # print keywrods of first article
print(keywords[1])  # print keywords of second article
print("done")
# keywords is a list
# keywords[0] is the list of keywords for the first article
# keywords[1] is the list of keywords for the second article
# ...
# keywords[0][0] is the first keyword of the first article
# keywrods[0][1] is the second keyword of the first article
# etc...
for i in range(len(keywords)):
    for k in range(len(data_lists)):
        c = keywords[i].count(data_lists[k])
        print(c)
df.replace(np.nan, c, inplace=True)
df.to_csv('file.csv', index=True)
```

```
from scipy.stats import chi2_contingency
from scipy.stats import chi2
stat, p, dof, expected = chi2_contingency(df)
print('dof=%d' % dof)
print(expected)
# interpret test-statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
# interpret p-value
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```