`tectr`: Paket zur Konstruktion eines Indikatorsystems

Samuel Lippl

2018-01-21

Einführung: Fragilität von Staaten

Die *Fragilität* von Staaten beschäftigt sich mit der strukturellen Analyse von Staaten im Hinblick auf die Krisenanfälligkeit. Das Indikatormodell SPIDER[1] der Bundeswehr hat das Ziel, diese Fragilität zu erfassen. Im Rahmen meiner Stelle als studentische Hilfskraft bei Prof. Dr. Manfred Sargl wirke ich statistisch an diesem System mit. Insbesondere bestehen meine Aufgaben daraus, gemeinsam mit meinem Kommilitonen Fabian Obster neue Indikatoren zu finden und auf ihre Tauglichkeit zu prüfen, statistische Methoden zur Vorhersage anzuwenden und die Ergebnisse zugänglich zu machen.
Da Krisen sich oft in Bürgerkriegen und anderen bewaffneten Konflikten manifestieren, ist dise Arbeit eng verwandt mit der Konfliktforschung. Diese Disziplin arbeitet zum großen Teil mit einfachen Modellen und wenigen prädiktiven Variablen. Diese Methodik wird sogar empfohlen (Schrodt 2014) und leitet sich vom Ziel der Konfliktforschung ab. Vorhersagen werden verwendet, um den Effekt einzelner Variablen abzuschätzen (Hegre et al. 2017). Je sparsamer das Modell[2], desto einfacher dessen Analyse.
Es ist daher nicht sonderlich kompliziert, in diesem Rahmen eine Arbeitsweise zu etablieren, die Overfitting vorbeugt und eine schrittweise Verbesserung des Modells erlaubt[3].
Anders steht es damit für SPIDER und den Versuch einer Vorhersage. Während die Konfliktforschung den Aufbau einer Theorie verfolgt, wie Konflikte entstehen und eskalieren, sollte SPIDER primär in der Anwendung funktionieren, d. h. so gute Vorhersagen wie möglich liefern. Hierzu haben Fabian und ich festgestellt, dass eine abstrakte Herangehensweise aus mehreren Gründen, die allgemein für programmatische Abstrahierung sprechen, sinnvoll ist. Unser Ansatz war ein Package in R. Seinem angedachten Nutzen, einen Bauplan für die Konstruktion eines Indikatorsystems zu bieten, entsprang der Arbeitstitel (archi)`tectr`.
Das grundlegende Konzept hinter dem Package und eine Auswahl seiner angedachten Vorteile will ich am Beispiel meiner konkreten Anwendung auf die Fragilität vorstellen (natürlich stark vereinfacht). Sofern ich die Funktionen bereits geschrieben habe, werde ich meine Erklärungen mit diesen illustrieren, sonst hoffe ich, die Ideen hinter dem Package mit Pseudocode veranschaulichen zu können. Bevor ich dazu übergehe, will ich aber noch zwei organisatorische Punkte ansprechen.

"Disclaimer"

Arbeitsteilung

Bisher ist nicht wirklich klargeworden, wie sich meine Aufgaben und die meines Kollegen unterscheiden, weshalb ich dies in einem kurzen Abschnitt klarstellen will.
Während wir gemeinsam an SPIDER arbeiten, habe ich bislang die Implementierung in R bis auf wenige Ausnahmen übernommen, weil ich mehr Erfahrung darin habe. Das Konzept, das ich im Folgenden darlege, habe ich mir überlegt und anschließend in Diskussion mit Fabian verfeinert. Insbesondere stammt der gesamte bisherige Code in `tectr` von mir und in der nächsten Zeit wird diese Arbeitsteilung sich auch so fortsetzen. Sollte es nötig sein, sie bis zum Ende der Bachelorarbeit beizubehalten, wäre dies für uns beide kein Problem. In jedem Fall wäre durch die Versionskontrolle mit Git und Github eine transparente Einsicht möglich, wer was geschrieben hat.

[1] siehe die Projektbeschreibung
[2] sofern kritische Variablen wie Population oder BIP pro Kopf berücksichtigt werden.
[3] ein Beispiel hierfür wäre Box's Loop (Colaresi and Mahmood 2017)

Einschreibung

Ich bin eigentlich Student der Mathematik und habe im letzten Wintersemester einen Bachelor in Statistik als Doppelstudium begonnen. In Rücksprache mit Prof. Augustin und Prof. Schmid sollte ich dem Doppelstudium eine einjährige Testphase geben, um zu sehen, ob ich damit klarkomme. Inzwischen habe ich 84 ECTS-Punkte gesammelt und plane, das Studium im nächsten Sommersemester abzuschließen. In diesem Sinne werde ich mich erst zum nächstmöglichen Zeitpunkt, also Ende Februar, offiziell einschreiben. Hierfür habe ich nochmal Rücksprache mit Prof. Schmid gehalten. Es wäre erst danach möglich, mit der Bachelorarbeit zu beginnen.

Implementierung

Content

Am Anfang von `tectr` steht ein Wissenschaftler der Anwendungsdisziplin. Im Zentrum seiner Arbeit steht der **Content**, der die statistische Arbeit formen soll. In meinem Fall handelte es sich hierbei um die Fragilität, für die zwei wichtige Dimensionen Kapazität und Autorität sind.

```
devtools::load_all()
#> Loading tectr
inds <- Content(c("Fragilität", "Kapazität", "Autorität"),
                c("Krisenanfälligkeit des Staates",
                  "Wie gut versorgt der Staat die Bürger mit Gütern?",
                  "Wie gut wahrt der Staat sein Gewaltmonopol?"))
inds
#>         name                                        definition
#> 1 Fragilität                    Krisenanfälligkeit des Staates
#> 2  Kapazität Wie gut versorgt der Staat die Bürger mit Gütern?
#> 3  Autorität       Wie gut wahrt der Staat sein Gewaltmonopol?
```

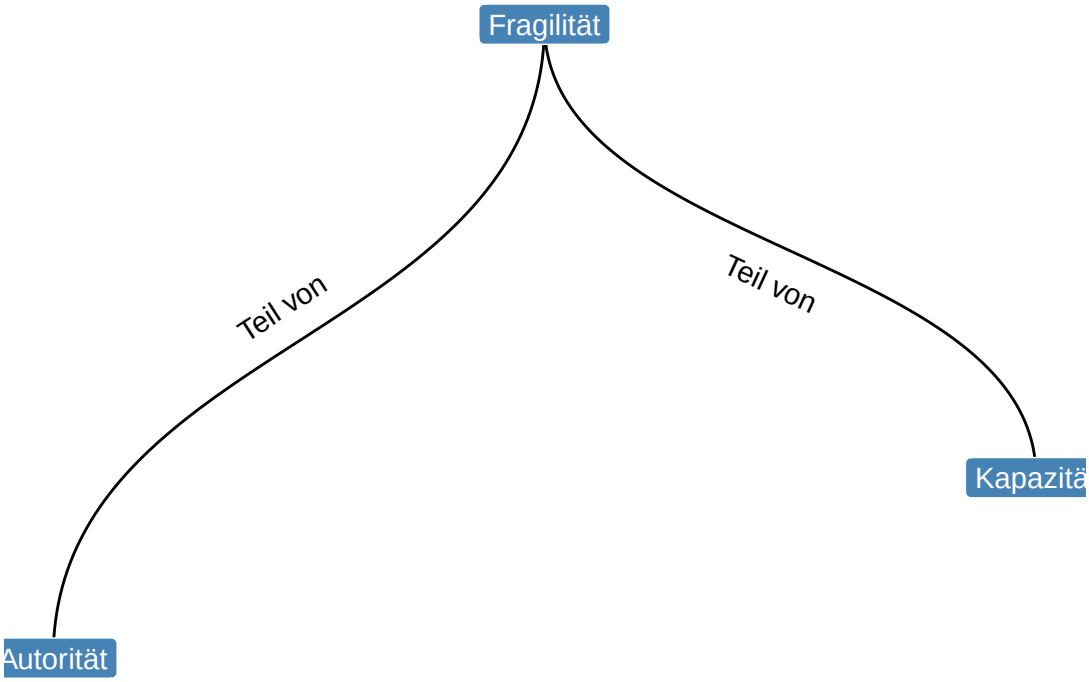
Am Ende der Beschreibung dieses Contents sollten alle Inhalte von einem Statistiker operationalisiert werden können. Es müssen also entweder externe Quellen vorliegen oder eine Anleitung, wie sich der Content aus anderen Contents zusammensetzt. Hierfür sind **C_Relations**, spezielle **C**ontents, entscheidend.

```
rels <- C_Relation(c("Teil von"))
rels
#>       name definition
#> 1 Teil von
```

Contents und C_Relations lassen sich zu **Structures** zusammensetzen, mit denen die Beziehung zwischen den Contents klar wird (die Visualisierung ist noch sehr archaisch):

```
frag <- Structure(variables = inds,
                  relations = rels,
                  edges = data.frame(from = c("Kapazität", "Autorität"),
                                     to = c("Fragilität", "Fragilität"),
                                     name = c("Teil von", "Teil von")))
frag
#> # A tbl_graph: 3 nodes and 2 edges
#> #
#> # A rooted tree
#> #
#> # Node Data: 3 x 2 (active)
#>   name       definition
#>   <fctr>     <fctr>
```

```
#> 1 Fragilität Krisenanfälligkeit des Staates
#> 2 Kapazität  Wie gut versorgt der Staat die Bürger mit Gütern?
#> 3 Autorität  Wie gut wahrt der Staat sein Gewaltmonopol?
#> #
#> # Edge Data: 2 x 3
#>    from    to  name
#>   <int> <int> <int>
#> 1     2     1     1
#> 2     3     1     1
visualize_struct(frag)
#> Using `nicely` as default layout
```



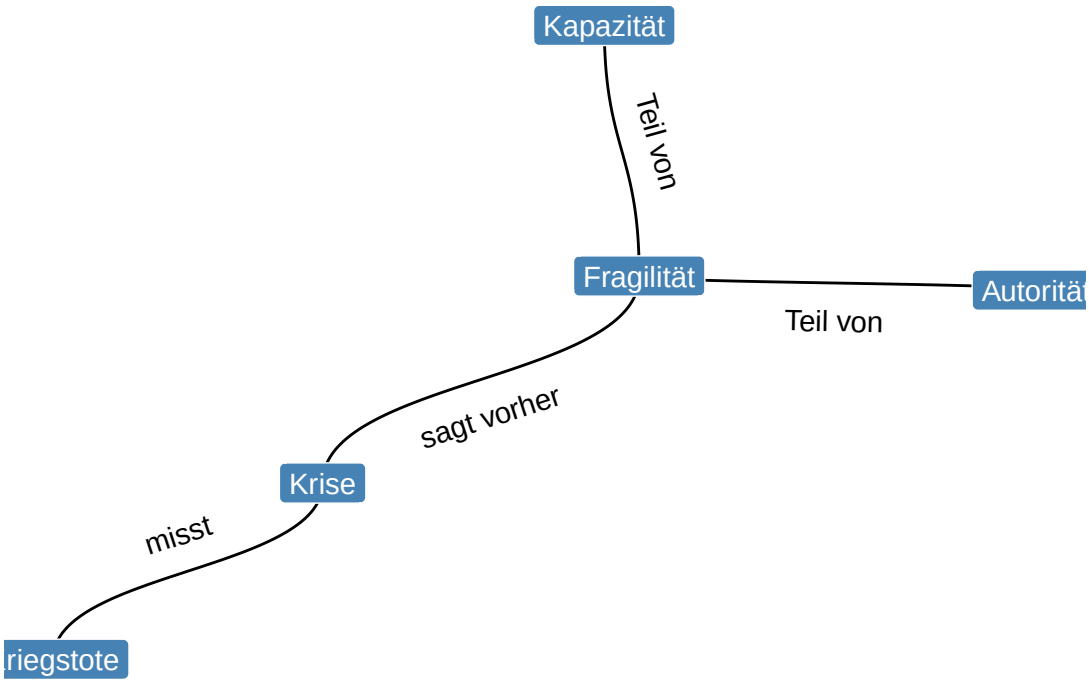
Weil aber diese Contents nicht messbar sind, muss der Wissenschaftler genauer angeben, was er wünscht. Um mehrere Arbeitsschritte zu vereinfachen und zu kürzen, wäre beispielsweise die Zahl der Toten in einem bewaffneten Konflikt[4] eine geeignete Operationalisierung von Krise.

```
library(dplyr)
#>
#> Attaching package: 'dplyr'
#> The following objects are masked from 'package:stats':
#>
#>     filter, lag
#> The following objects are masked from 'package:base':
#>
#>     intersect, setdiff, setequal, union
```

[4] siehe die Seite der UCDP

```r
frag <- frag %>%
  extend_variables(Content(c("Krise", "Kriegstote"),
                           c("In einem Staat gibt es große Probleme.",
                             "UCDP-Messung"))) %>%
  extend_relations(C_Relation(c("misst", "sagt vorher"))) %>%
  extend_edges(data.frame(from = c("Kriegstote", "Fragilität"),
                          to = c("Krise", "Krise"),
                          name = c("misst", "sagt vorher")))

frag
#> # A tbl_graph: 5 nodes and 4 edges
#> #
#> # A rooted tree
#> #
#> # Node Data: 5 x 2 (active)
#>   name       definition
#>   <chr>      <chr>
#> 1 Fragilität Krisenanfälligkeit des Staates
#> 2 Kapazität  Wie gut versorgt der Staat die Bürger mit Gütern?
#> 3 Autorität  Wie gut wahrt der Staat sein Gewaltmonopol?
#> 4 Krise      In einem Staat gibt es große Probleme.
#> 5 Kriegstote UCDP-Messung
#> #
#> # Edge Data: 4 x 3
#>    from    to  name
#>   <int> <int> <int>
#> 1     2     1     1
#> 2     3     1     1
#> 3     5     4     2
#> # ... with 1 more row
visualize_struct(frag)
#> Using `nicely` as default layout
```



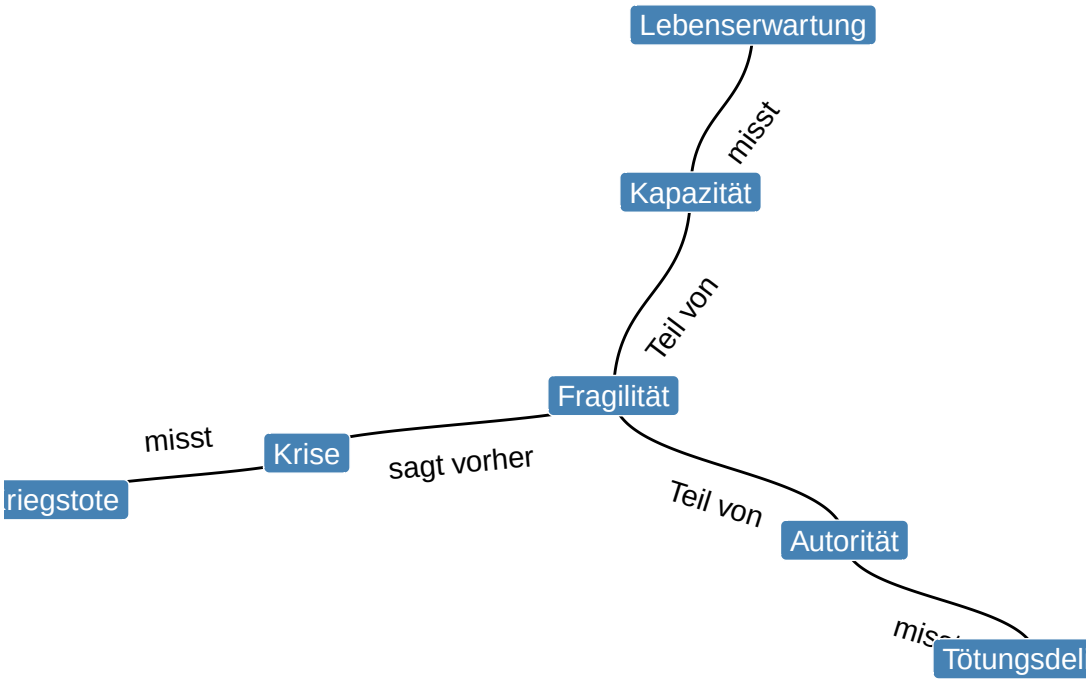
In weiteren Arbeitsschritten müssten Autorität und Kapazität näher charakterisiert werden. In diesem Sinne lässt sich eine anständige Beschreibung des inhaltlichen Modells finden, die die Operationalisierung leicht macht. Die Struktur soll insbesondere den Zusammenhang zwischen den inhaltlichen Überlegungen und der statistischen Modellierung herausstellen. Stark vereinfacht hätten wir mit der vorliegenden Spezifizierung als Resultat ein Modell, das mit Tötungsdelikten und Lebenserwartung die Kriegstoten in einem Staat vorhersagt.

```
frag <- frag %>%
  extend_variables(Content(c("Lebenserwartung", "Tötungsdelikte"),
                           c("Weltbank-Datenbank Lebenserwartung",
                             "Weltbank-Datenbank Tötungsdelikte"))) %>%
  extend_edges(data.frame(from = c("Lebenserwartung", "Tötungsdelikte"),
                          to = c("Kapazität", "Autorität"),
                          name = c("misst", "misst")))

frag
#> # A tbl_graph: 7 nodes and 6 edges
#> #
#> # A rooted tree
#> #
#> # Node Data: 7 x 2 (active)
#>   name            definition
#>   <chr>           <chr>
#> 1 Fragilität      Krisenanfälligkeit des Staates
#> 2 Kapazität       Wie gut versorgt der Staat die Bürger mit Gütern?
#> 3 Autorität       Wie gut wahrt der Staat sein Gewaltmonopol?
#> 4 Krise           In einem Staat gibt es große Probleme.
#> 5 Kriegstote      UCDP-Messung
```

```
#> 6 Lebenserwartung Weltbank-Datenbank Lebenserwartung
#> # ... with 1 more row
#> #
#> # Edge Data: 6 x 3
#>    from    to  name
#>   <int> <int> <int>
#> 1     2     1     1
#> 2     3     1     1
#> 3     5     4     2
#> # ... with 3 more rows

visualize_struct(frag)
#> Using `nicely` as default layout
```



Compounds

Zum momentanen Zeitpunkt ist noch nicht klar, über was diese Contents eine Aussage treffen. Im Fall der Fragilität von Staaten ist dies eine Region auf der Weltkarte, angegeben entweder durch Koordinaten oder ein Polygon aus Koordinaten. Warum ist es nicht sinnvoll, diese einfach als weitere Variable zu betrachten? Aus meiner Sicht sind drei Gründe entscheidend:

1. Oftmals sind beliebig viele Instanzen dieser Objekte vorhanden. Beispielsweise könnten wir hier ein beliebiges Koordinatenpolygon angeben. Das bietet natürlich keine Mehrinformation. Vielmehr ist die Zuordnung der anderen Variablen zu einer bestimmten Region ein Mehrwert an Information.
2. Es kann oftmals hilfreich sein, verschiedene Instanzen vereinigt zu betrachten, z. B. könnten wir

Informationen über ein Land für einzelne Provinzen gegeben haben, die wir dann zusammenfassen müssen. Für die anderen Variablen ist eine solche Vorstellung nicht sinnvoll.

3. Koordinatenpolygone beispielsweise verbrauchen sehr viel Speicher und Operationen sind komputationell teuer. Weil wir oft mit den gleichen Polygonen (oder anderen Instanzen) arbeiten, könnten nicht evaluierte Ausdrücke und die Identifikation der Polygone mit IDs (oder Worten wie "Afghanistan") bessere Leistung bedeuten.

Mein Ansatz für diese Themen sind **Compounds**. Ein Compound $\mathcal{C}$ besteht aus seinem **Universum** C und weiteren Mitgliedern, sodass $\bigcup \mathcal{C} = C$, d. h. alle Element des Compounds sind eine Teilmenge von C. Allgemein sind die Elemente von $\mathcal{C}$ diejenigen Instanzen, für die wir Werte für unsere Variablen zu berechnen haben. Wieso vereinfacht das die oben angesprochenen Probleme? Um dies zu beantworten, würde ich noch ein zweites spezielles Element **Void** $C^0 \in \mathcal{C}$ einführen, was genau jenem Teil von C entspricht, für den wir keine Wert gegeben haben. Betrachten wir beispielsweise Staaten, so könnten wir für alle Kontinente Daten gegeben haben, aber für die Ozeane nicht. Es wäre sinnvoll als Universum die ganze Weltkarte in Form von Koordinaten $C := [-180, 180) \times [0, 180]$ zu definieren. In diesem Fall wäre C^0 beispielsweise das Gebiet, das keinem Staat gehört.
Wir nennen eine Teilmenge $A \subseteq \mathcal{C}$ **Partition** von $\mathcal{C}$, wenn sie eine Partition von $C \setminus C^0$ darstellt, d. h. alle Elemente $a \in A$ sind disjunkt und $\bigcup A = C \setminus C^0$. Beispielsweise wären alle Staaten eine Partition, alle Kontinente (solange wir diese als Vereinigung von Staaten betrachten) aber auch. Insbesondere sind die Kontinente **gröber** als die Staaten. Wir nennen A **atomar**, wenn alle anderen Partitionen gröber sind.
Wie hilft uns das?
Erstellen wir ein Compound `Compound(universe = universe, members = members)`! Als ersten Member enthält diese z. B. China als Polygon. China wird entweder mit einer id `1` oder mit einem Namen `chn` identifiziert - sagen wir zweiteres. Falls wir nun als zweiten Member das Territorium von Indien `ind` haben, könnte es sein, dass unsere Quelle (die für den zweiten Member eine andere ist als für den ersten) Kashmir als gänzlich zu Indien zugehörig betrachtet. Da einige Territorien in Kashmir (obgleich umstritten) eigentlich zu China gehören, überschneiden sich die beiden Regionen. Interessieren wir uns nun im dritten Member für die chinesischen Regionen von Kashmir `chn_kas`, so müssten wir diese nicht getrennt speichern, sondern könnten `chn_kas = chn:ind` schreiben. Dann würden die Compounds aus einem Vektor von Polygonenlevels `c("chn" = , "ind" = )` bestehen. Falls uns die Bezeichnung `chn_kas` statt `chn:ind` wichtig ist, könnten wir auch diese festlegen.
Hier wären natürlich verschiedene Implementierungen denkbar (etwa die Herleitung aller Atome), aber das Prinzip besteht darin, die vielen Compounds durch die Operationen `+` (Vereinigung), `-` (ohne) und `:` (Schnitt) kompakter und übersichtlicher zu gestalten. Erst, wenn wir hierfür auch Variablen gegeben haben, fügen wir dabei eine bestimmte Instanz zu den Compounds hinzu. Das heißt aber nicht, dass uns unsere Variablen ansonsten keine Information über einen Compound verschaffen (s. u.).

Operatoren

Operatoren sind das Herzstück von `tectr` und in ihrer Essenz Variablen. Ihr Ziel ist, Aktualisierung des Modells (auch der Regression) flexibel zu gestalten und das Modell gleichzeitig für Nicht-Statistiker möglichst anschaulich zu visualisieren. Operatoren sind Funktionen von der Menge der Compounds in eine Menge V, die einen Unbekannt-Wert NA enthalten muss:

$$Op : \mathcal{C} \to V, V(C^0) = NA$$

Definiert werden sie rekursiv. Wir beginnen mit den **externen Operatoren**. Diese bieten die Verbindung zu den Messungen, die auf dem Computer oder im Internet gelagert werden: `ext_ops(file = "~/data/Population.csv")`. Damit können sie jederzeit aktualisiert werden, was zu einer Kaskade aller abhängigen Variablen führt. Die `read`-Funktion erkennt eine mögliche Subklasse als Quelle (z. B. "Worldbank") und liest die Daten entsprechend ein.
Die N externen Operatoren seien nun durch $O_N := (Op_1, \ldots, Op_N)$ gegeben. Die Menge dieser möglichen Operatoren sei $\mathcal{O}_N := \prod_{i=1}^{N} \mathcal{F}(\mathcal{C}, V_i)$. Für $k > N$ können wir nun den **internen Operator** Op_k in Abhängigkeit von $\mathcal{O}_N$ definieren. Wir haben also eine Herleitung $Op_k^D : \mathcal{O}_k \to \mathcal{F}(\mathcal{C}, V_k)$ gegeben, sodass

$Op_k^D(O_k) = Op_k$ ist. Wohlgemerkt könnte man die Herleitungen als Funktionen auf einer Operatormenge von Compounds definieren, falls dies nützlich ist.
Diese Vorstellung von Operatoren ist zugegebenermaßen noch recht abstrakt, weil ich mir selbst noch nicht ganz sicher bin, welche ihrer Eigenschaften nützlich und welche unnötig kompliziert sind. Deshalb werde ich an dieser Stelle lieber noch kurz auf das Ziel und die angedachten Vorteile dieses Konzepts eingehen:

1. Operatoren sollen die Herleitung einer Regression transparent gestalten. Wird eine Regression vollzogen, so wird ein "Snapshot" der dafür verwendeten Daten gemacht, was lediglich bedeutet, dass sich das System merkt, welche Compounds für die Regression verwendet wurden. Verändern sich diese in den externen Quellen, verschwindet die Regression nicht, aber es wird gewarnt. Zusätzlich kann man manuell eine aktualisierte Regression auswählen und, falls nötig, unkompliziert erkennen, mit welchen Observationen die Regression erstellt wurde und (z. B. in Ergänzung korrellierter Fehler) aktualisieren.
2. Es ist leicht, Operatoren für zusammengesetzte Variablen zu definieren, indem man aus einem gegebenen Operator eine Erweiterung herleitet, die durch einfache Regeln gegeben ist. Betrachten wir diese beispielsweise für die Bevölkerung. Ist diese für Atome von Staaten oder anderen Regionen gegeben, so können wir die Bevölkerung zusammengesetzter Regionen einfach durch die Regel $Op_k(x+y) := Op_k(x) + Op_k(y)$ definieren. Das erleichtert die Arbeit, falls wir unterschiedliche Variablen auf unterschiedlichen Detailebenen gegeben haben. Speziell sind manche der für uns interessanten Variablen für manche Länder für die einzelnen Provinzen verfügbar, während es für andere Länder nur aggregierte Daten gibt. Operatoren und Compounds bieten eine Möglichkeit, diese Unterschiede zu berücksichtigen, aber gleichzeitig alle Daten einbeziehen zu können.
3. Im Operatoren-Framework sollten auch Hypothesentests einfacher funktionieren, indem man sicherstellen kann, dass nur Daten, die zu diesem Zeitpunkt noch nicht eingelesen (oder verwendet) worden waren, zur Überprüfung der Nullhypothese verwendet werden.
4. Indem für jeden Operator ein Content-Äquivalent vorliegt, ist sofort ersichtlich, wie die statistische Modellierung der eigentlichen Planung entspricht und wo sich künstliche Contents wie Principal Components eingliedern. Die Visualisierung der Operators und ihre Herleitung bietet darüber hinaus eine angenehme Übersicht auch über komplexe Systeme, speziell wenn man mehrere Systeme vergleichen will.

Visualisierung

Besonders wichtig für unsere Zwecke ist ein leicht bedienbares Programm, in dem gut erkennbar ist, woher Beurteilungen kommen. In dem vorliegenden Rahmen ist es leicht eine abstrakte Shiny-App dafür zu schreiben und diese dann durch spezielle Funktionen mit Leben zu füllen. Denkbar wäre zum Beispiel eine erste Darstellung der Fragilität auf einer Weltkarte. Klickt man auf eine Region, so wird eine detailliertere Analyse dieser Region angezeigt und wie die unterschiedlichen Operatoren in den letztendlichen Fragilitätswert eingeflossen sind. Klickt man auf einen Operator, so sieht man, wie dieser Operator hergeleitet worden ist bzw. was seine Quelle ist. Darüber hinaus sieht man die Werte dieses Operators auf einer Weltkarte. Speziell ist hier eine Einteilung der Operatoren in verschiedene Klassen denkbar, für die die Visualisierung unterschiedlich aussieht. Wir haben beispielsweise manche Variablen (z. B. Bevölkerung) für Länder, andere (z. B. Terroranschläge) aber für einzelne Koordinaten gegeben. Diese Operatoren in einzelne Klassen aufzuteilen und dementsprechend (gerade in der Visualisierung) unterschiedlich mit ihnen umzugehen, erleichtert einem die Arbeit, das jedes Mal einzeln festlegen zu müssen.

Schlussbemerkung

Zusammenfassend habe ich vor allem zwei Ziele: einerseits möchte ich eine geeignete Sprache finden, um Indikatorsysteme zu beschreiben. Diese soll es erleichtern, die die Systeme zu evaluieren, zu verändern, zu vergleichen und zu visualisieren. Andererseits möchte ich eine möglichst effiziente Implementierung dieser Sprache in R finden, um meine Arbeit zur Fragilität von Staaten (und vergleichbaren Aufgaben) zu erleichtern.

Literaturverzeichnis

Colaresi, Michael, and Zuhaib Mahmood. 2017. "Do the robot: Lessons from machine learning to improve conflict forecasting." *Journal of Peace Research* 54 (2): 193–214. doi:10.1177/0022343316682065.

Hegre, Håvard, Nils W. Metternich, Håvard Mokleiv Nygård, and Julian Wucherpfennig. 2017. "Introduction: Forecasting in peace research." *Journal of Peace Research* 54 (2): 113–24. doi:10.1177/0022343317691330.

Schrodt, Philip A. 2014. "Seven deadly sins of contemporary quantitative political analysis." *Journal of Peace Research* 51 (2): 287–300. doi:10.1177/0022343313499597.