

Visualizing Opinion Space – Interactive Geographic Map Representations for Dynamic Opinion Datasets

Master Thesis of

Sophie von Schmettow

At the Department of Informatics
Institute of Theoretical Computer Science

Reviewers: Prof. Dr. Dorothea Wagner
Prof. Dr. Peter Sanders
Advisors: Prof. Dr. Gregor Betz
M.Sc. Michael Hamann
Dr. Tamara Mchedlidze

Time Period: 1st July 2017 – 31th December 2017

Acknowledgements

First of all, I want to thank my supervisors, Gregor Betz, Michael Hamann, and Tamara Mchedlidze. This work is only possible due to the fruitful discussions in our weekly meetings throughout this project, thank you for your expertise, your patience, and not least for your enthusiasm, which never ceased to motivate me.

I am grateful to the Algorithmics Group of Dorothea Wagner for this opportunity to complete my thesis in a pleasant and creative atmosphere.

I would also like to thank my proofreaders (including my supervisors) for their comments on this work.

Just as much I would like to express my gratitude to my family and friends for their love and encouragement during the last years and for their support of my studies.

Statement of Authorship

I hereby declare that this document has been composed by myself and describes my own work, unless otherwise acknowledged in the text.

Karlsruhe, 29th December 2017

Abstract

Regarding large datasets, humans often have to rely on visualization methods in order to make sense of and extract relevant information from them. Kobourov et al. propose to depict relational data in the form of geography-like maps to create highly intuitive and readable visualizations that show clustering and neighbourhoods explicitly [GHK10]. In this work we explore how this idea can be used for structuring, analysing, and visualizing large bodies of opinion data, as they arise from large public debates.

We develop OPMAP, a tool to create interactive dynamic map representations of the argumentatively structured opinion landscape, which allows users to input and locate their own opinion. First, we specify a formal model of the space of opinions and evaluate various possibilities to measure semantic distances between them. In particular, we extract a graph from the opinion data where edge weights between opinion vertices are given by the *degree of mutual coherence* [DM07]. The latter is a well studied measure in Bayesian epistemology, which renders the notion that coherent propositions support each other formally precise in probabilistic terms. We then investigate which clustering and graph drawing algorithms can be combined with such metrics in practice. The final tool uses infomap community detection [RB08] in conjunction with a force-directed layout algorithm. To test the effectiveness of OPMAP, we use simulated opinion datasets and collect an empirical sample of opinions on eating behaviours. Thereby we illustrate that semantic proximity in the data is captured well in the resulting maps.

Deutsche Zusammenfassung

Um große Datensätze interpretieren und die relevanten Informationen extrahieren zu können, sind menschliche Betrachter oftmals auf visuelle Repräsentationen angewiesen. Kobourov et al. schlagen vor, relationale Datensätze in Form geographischer Karten zu visualisieren und somit hochgradig intuitive und lesbare Visualisierungen zu erstellen [GHK10]. Diese zeigen in den Daten bestehende Cluster- und Nachbarschaftsinformationen explizit. In der vorliegenden Arbeit untersuchen wir, wie diese Idee zur Strukturierung, Analyse und Visualisierung umfangreicher Meinungsdatensätze, wie sie in großen öffentlichen Debatten entstehen, verwendet werden kann.

Hierbei entwickeln wir OPMAP, ein Programm zur Erstellung interaktiver dynamischer Kartenrepräsentationen von argumentativ strukturierten Meinungslandschaften, welches den Nutzern die Eingabe und Lokalisierung ihrer eigenen Meinung erlaubt. Zuerst wird ein formales Modell des Meinungsraums spezifiziert, und verschiedene Möglichkeiten, semantische Distanzen innerhalb dieses Raums zu messen, werden evaluiert. Insbesondere wird dabei ein Graph aus dem zugrundeliegenden Meinungsdatensatz extrahiert, bei welchem die Kanten zwischen den Meinungsknoten mit dem *Grad der wechselseitigen Kohärenz* [DM07] gewichtet werden. Dieser ist ein gut untersuchtes Maß in der Bayesianischen Erkenntnistheorie, welches das intuitive Verständnis, dass kohärente Aussagen sich gegenseitig unterstützen, mittels wahrscheinlichkeitstheoretischer Methoden formal präzisiert. Wir untersuchen weiterhin, welche Cluster- und Layout-Algorithmen mit einem solchen Maß in der Praxis kombiniert werden können. Der Infomap Cluster-Algorithmus [RB08] in Kombination mit einem kräftebasierten Layout-Algorithmus erweist sich für die Verwendung in OPMAP als geeignet. Um die Effektivität der Methode zu testen, nutzen wir simulierte Meinungsdatensätze und erheben eine empirische Stichprobe von Meinungen zum Thema Essverhalten. Hiermit wird gezeigt, dass die resultierenden Karten semantische Nähe in den Daten darstellen.

Contents

1. Introduction	1
1.1. Contribution	2
1.2. Concept & Preliminaries	3
2. Related Work	7
2.1. Maps as a Means for Information Visualization	7
2.1.1. Spatial Statistical Data	7
2.1.2. Map Metaphors	8
2.2. Tools for Argument & Opinion Analysis	10
3. Modelling the Opinion Space	13
3.1. Basic Argumentation Theory	13
3.2. Probabilistic Epistemology behind OPMAP	16
3.2.1. Benchmarks	17
3.2.2. Measures based on the Degree of Justification	18
3.3. Opinion Data	21
3.3.1. Simulated Data	21
3.3.2. Empirical Data	22
4. Finding Structure	25
4.1. Clustering Experiments	25
4.1.1. Louvain Method	25
4.1.2. Infomap	27
4.1.3. Miscellaneous Tested Algorithms	29
4.1.4. Evaluating the Algorithms	31
4.2. Vertex Filtering	34
4.3. Edge Filtering	36
4.4. Semantic Analysis	36
5. Graph & Map Drawing	39
5.1. Force Simulation	40
5.2. Voronoi Decomposition	46
6. Implementation Details	53
6.1. Online Survey	53
6.2. Generating Graph Files	55
6.3. Processing Graph Files	56
6.4. Interactive Visualization	59
7. Conclusion and Outlook	67
7.1. Summary	67
7.2. Future Research	68

List of Figures	71
List of Tables	71
Bibliography	72
Appendix	77
A. Appendix Section 3	77
A.1. Veggie-Debate: Sentence Pool	77
A.2. Survey Design	84
A.3. Survey Results	85
B. Appendix Section 4	87
B.1. Country Descriptions	87
B.2. Clustering Results	88

1. Introduction

As a student I often find myself marvelling at the humongous amounts of knowledge captured in written word. According to an estimate by Google 129,864,880 books had been published by 2010.¹ The work of Shakespeare alone comprises 1,223 figures, and a total of 884,421 words of dialogue.² An ambitious reader may know around 200 books rather well, but that is already difficult. In modern times the amount of textual data grows faster than ever. As of December 2017 Facebook has 2.07 billion users with 5 new profiles being created *every second*, 510,000 comments and 293,000 status updates posted, and 136,000 photos uploaded every minute.³ The Wikipedia online encyclopedia features 5,539,237 English articles and 43,840,783 wiki pages in total⁴. In face of these numbers it is not surprising that throughout my university career I regularly got the nagging feeling that the more I learned, the more I realized how little I knew. Accordingly, algorithmic methods to make sense of huge quantities of information—in particular by creating clearly structured visualizations—are a research topic of growing interest. Cutting-edge approaches include graphs, taxonomies, and maps.

In traditional cartography, the underlying geography has to be represented in a simplified manner while still showing important structures and significant relationships. Likewise, when mapping information the objective is to create a visually accessible representation while preserving as much content and its underlying structural relationships as possible.

This thesis is concerned with opinion data, as they are being generated in large public debates. For example in the run-up to parliamentary elections a multitude of big and small political issues are heatedly discussed on news media, the internet, family breakfast tables etc. When trying to form a thorough, critical opinion voters may have to struggle through a thicket of endless circular internet commenting threads, riddled with misleading partial information and “ad hominems”. The work presented here aims to add a novel approach to the tools that are already available for facilitating in-depth exploration of every aspect of a discussion, some of which are introduced in Section 2.2.

As a use case we employ the ongoing debate about nutrition behaviours, which will henceforth be referred to as the *Veggie-Debate*. People choose a certain diet for an ample variety of reasons—financial, culinary, ethical and health considerations amongst many others. As a result, the amount of opinions on eating habits is vast. The variety of opinions

¹<http://booksearch.blogspot.de/2010/08/books-of-world-stand-up-and-be-counted.html>

²<https://www.opensourceshakespeare.org/stats/>

³<https://zephoria.com/top-15-valuable-facebook-statistics/>

⁴https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

that are advocated in such a complex debate, as well as relationships between them, is hard to grasp if they are presented in a purely textual form (e.g. as a dialogue), even if individual propositions are given in a concise and objective manner.

In order to make sense of these data, we propose to visualize them in form of a 2D geography-like map. Herein, we do not aim to generate accurate but rather very intuitive and readable representations. Most people are quite familiar with geographic maps and even delight in carefully examining them. And just as a geographic map is used to navigate to points of interest, an opinion map may aid empirical discourse analysts, communication scientists, sociologists, political scientists, and everyone who wants to be more thoughtful about an issue to navigate through the landscape of information. Such an opinion map, in case of the Veggie-Debate, could look as shown in Figure 1.1.



Figure 1.1.: A map of the “World of Diets”.

Note that the map shown in Figure 1.1 is purely hypothetical. It was drawn by hand as an initial concept and is not based on real data. However, it illustrates important properties of the visualization: Similarity between opinions is captured by “geographical” distances in the map: People who share the same opinions on how a good diet looks like (as well as the same reasons for their opinion) are part of the same “country”. For example, opinions that advocate an omnivorous diet mostly for the pleasure experience of eating are “citizens” of *Hedonistica*, or opinions that advocate a vegan diet because they hold that animals have a right to life are found in *Animal Ethica*. Furthermore, the size of countries and labels as well as the length of country borders roughly reflect the structure of the underlying *opinion space*. The more opinions exist that represent the same general eating behaviour, the bigger the area of the corresponding country. The font size of the labels is also adjusted according to some notion of importance, such as the number of “inhabitants” of a country. The length of the borders between neighbouring countries corresponds to some notion of similarity between the represented eating behaviours.

1.1. Contribution

In the scope of this thesis we design OPMAP, a tool for structuring, analysing, and visualizing the space of opinions spanned by a complex public debate as a pseudo-geographic map. The contribution of this work is twofold. First, we develop a web-based application that

practically implements OPMAP for the example of the Veggie-Debate. It generates a dynamic map, where users can locate themselves by taking a short survey, which is also used to collect the data to create the initial map. We had the opportunity to set up the application at the exhibition “Open Codes” at the Zentrum für Kultur und Medien (ZKM)⁵ in Karlsruhe. In the installation, the map is projected on a tilted table. With an appropriate note concerning privacy issues, users are offered to input their opinion on an iPad by taking the survey. After completion of the survey, their opinion is integrated into the map and a description of their “home country” appears.

The second contribution is an experimentation on how to create a working application by combining theoretical results from different disciplines, which captures the structural relationships of a dataset well, rather than providing an *exact* representation of it. We tested how various epistemological distance metrics work with several clustering algorithms in conjunction with a force-directed layout algorithm to create an animated map. To make the installation possible, we had to make some design decisions at the cost of the theoretical exploration for the sake of practical feasibility within the prescribed time frame.

Additionally, a multitude of interesting and promising theoretical and practical research directions were cleared up in the course of development (see Section 7.2).

The remainder of this chapter outlines the steps necessary to create a map representation from a relational dataset. Along the way, some important concepts, notation and terminology are introduced. The following chapter provides an overview of related work in the fields of argumentation theory and information visualization. Chapter 3 describes how we model the opinion space, particularly how we obtain opinion samples, and how we measure similarity between opinions. The subsequent two chapters detail the mapping algorithm and concomitant challenges. Chapter 4 describes how the “country” structure was determined, and Chapter 5 introduces the applied methods from the fields of graph drawing and computational geometry. Chapter 6 provides implementational details of the data processing and the application. Finally, Chapter 7 evaluates our work, provides a conclusion and states some research problems which remain to be tackled.

The following typographical conventions will be used throughout the thesis: *Italic* is used within definition environments and indicates new terms (or simply emphasis). SMALL CAPITALS are used to refer to software libraries and tools; programming languages are typeset in regular font. **Typewriter font** indicates code in the running text, example statements from the debate, file names, file extensions, and URLs.

1.2. Concept & Preliminaries

The input to our mapping algorithm is a *graph* or *network*. A graph is a structure that models a set of objects and pairwise *relationships* between them. The objects are called *vertices* of the graph and the related pairs are indicated by *edges*.

Definition 1.1. *Graph/Network*

A graph is a tuple $G = (V, E)$ with a set V of $n = |V|$ vertices and a set $E \subset V \times V$ of $m = |E|$ edges that represent relationships between the vertices they connect. If these relationships are symmetric, the graph is **undirected** and edges are denoted as unordered set $e = \{u, v\}$.

Graphs are ubiquitous data structures in computer science but we are also surrounded by graphs in our daily lives, that are interesting to explore. In fact, two familiar examples have already been mentioned. First, Wikipedia, where the vertices correspond to encyclopedia

⁵<https://open-codes.zkm.de/de>

articles and the edges to hyperlinks between them, second, social networks such as Facebook. The vertices in this case are the Facebook users, and any two users who are friends on Facebook are connected by an edge. If the relation is not binary, as it is the case here (“A is friends on Facebook with B” is either true or false), weights may also be assigned to the edges. A *weighted graph* is a triple $G = (V, E, \omega)$ where ω is a function that assigns weights to the edges

$$\omega : E \rightarrow \mathbb{R}$$

In this case, an illustrative example is the set of books for sale on Amazon. You can connect any two books by an edge weighted with the number of “customers who bought A that also bought B”.

In this thesis, we study the so-called *opinion-graph*, where the vertices represent *opinions* or *positions* (we use these terms interchangeably) and the weighted edges connecting two opinions correspond to a notion of *similarity* between them. In this context, we define a position to be a *truth-value assignment* to the relevant sentences featured in the underlying debate. This formalization will be elaborated in Section 3.1. A formal account of possible distance measures in the space of opinions will be given in Section 3.2.

One of the most common ways to represent a graph structure visually is to draw it in the 2D plane by representing the vertices as points and the edges as curves connecting the corresponding vertices, as it is the case in Figure 1.2. A *drawing* or *layout* of a graph is the assignment of 2D-positions to vertices and edges to curves:

Definition 1.2. Drawing/Layout

Let $G = (V, E)$ be a graph. A *drawing* or *layout* Γ is a function that maps each vertex $v \in V$ to a distinct point $\Gamma(v) \in \mathbb{R}^2$ on the 2D plane and each edge $(u, v) \in E$ to a simple open curve c_{uv} with endpoints $c_{uv}(0) = \Gamma(u)$ and $c_{uv}(1) = \Gamma(v)$.

Creating such drawings by hand becomes infeasible as networks become larger. There is a plethora of network visualization algorithms that produce drawings fulfilling different criteria. Figure 1.2a generically shows two different example representations of the same underlying graph, a small specimen with 20 vertices and 30 edges. Among others, there is a family of layout algorithms of great computational beauty, so-called *Force-Directed Algorithms* (for example [FR91]), also called *Spring Embedders*⁶. They are inspired by the mechanical analogy of a mass-spring system: Vertices in the graph can be imagined as spheres which are connected by springs. In accordance with the similarity measure the spheres are either attracted to or repelled from each other, until the system converges to a state of minimal energy. Or respectively, the drawing converges to a state in which connected vertices are drawn closer to each other and disconnected vertices are farther apart. In the context of drawing algorithms, drawing *conventions* and drawing *aesthetics* are distinguished. The former are properties that must not be violated, and the latter are desirable properties that are optimized but not strictly enforced. Details on the layouting of the graph will be given in Section 5.1.

While network visualization algorithms can provide some intuition, exact analytical methods are required to gain a deeper understanding of a graph’s structure. These methods, which allow us to assess whether the graph contains some denser *communities*, are called *graph clustering* or *community detection*. They aim to assign the vertices in the graph to different groups such that vertices in the same group are more similar to each other than to those in other groups. Or, to use the map metaphor, they determine the “nationality” of every opinion.

⁶*Embedding* a graph, by contrast to drawing a graph, means that vertices are associated with their corresponding dots and edges are associated with their corresponding lines, without assigning actual coordinates or curves.

Definition 1.3. Clustering/Community Detection

Let $G = (V, E)$ be a graph. A clustering is a partition

$$\mathcal{C} = \{C_1, \dots, C_k\}$$

of V such that $\bigcup_{i=1}^k C_i = V$ and $\forall C_i \in \mathcal{C} : C_i \neq \emptyset$. The elements C_i are called clusters or communities.

Intuitively, in a good clustering the vertices of the same cluster are tightly connected to each other, while the connections between the vertices belonging to different clusters are much looser. This is generically illustrated in Figure 1.2b. It can be seen that the clusters (indicated by the dashed circles) are connected more densely than vertices between the clusters. Maximizing intra-cluster density and inter-cluster sparsity is one of the main clustering paradigms. There are a lot of different formalizations of this paradigm. The ones used here are given in Section 4.1.

When a cluster assignment is provided for a network it is natural to have a visualization depict it. There are many ways to do this, the most obvious of which is to use colours. Here we use so-called *Voronoi diagrams* (also known as *Dirichlet decomposition*, see for example [DBCVKO08]) to create the actual map. These diagrams divide the 2D plane into regions based on the distance to points on that plane, in our case the vertices in the drawing. For each vertex there is a corresponding region consisting of all points closer to that vertex than to any other (c.f. Figure 1.2c). Finally, the regions corresponding to nodes of the same cluster are given the same colour and we obtain our countries by merging cells of the same colour, as shown in Figure 1.3. It can be seen that the regions are always convex polygons.

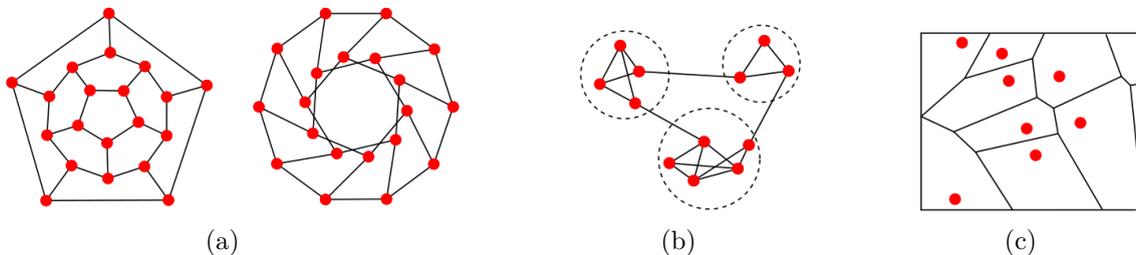


Figure 1.2.: Steps in the mapping algorithm. (a) Graph drawing. (b) Clustering. (c) Voronoi decomposition.



Figure 1.3.: Obtaining “landmass”. (a) Voronoi cells belonging to vertices in the same cluster are assigned the same colour. (b) Cells of the same colour are merged to form countries.

2. Related Work

OPMAP seeks to integrate a variety of results from argumentation theory, network analysis, and visualization algorithms in order to depict the argumentatively structured opinion landscape of a debate. This chapter investigates how (literal and metaphorical) maps are used to visualize information (Section 2.1) and introduces some existing tools for argument and opinion analysis (Section 2.2).

2.1. Maps as a Means for Information Visualization

2.1.1. Spatial Statistical Data

There is a multitude of techniques which use maps of geographic locations to display additional spatial information. Familiar techniques for *thematic mapping* are shown in Figure 2.1. On *choropleth maps*, predefined regions, such as states, are coloured or shaded according to some aggregated statistical dimension, such as population density, water use etc. (Figure 2.1a). This technique assumes that the measured phenomenon is distributed more or less evenly within the regions [AA99]. On *proportional symbol maps*, symbols, most commonly discs, are placed on various locations of interest on the map and are scaled according to a statistic (Figure 2.1b) [CHvKS10]. Continuous and non-continuous *area cartograms* (also called *value-by-area maps*) distort geographic boundaries in a map to depict the area of a region as proportional to some external size rather than the actual area. In case of the latter, regions are rescaled around their centroid, preserving local shapes but breaking the map down into pieces. In case of the former, the map stays connected but all shapes are distorted (Figure 2.1c) [AKV15]. *Dorling cartograms* maintain neither object topology, shape nor centroid—instead, regions are replaced by a uniform shape, usually circles or rectangles, of proportional size (Figure 2.1d) [Dor11]. Not only statistical data that are linked to actual geographic locations can be visualized using maps, but any set of relational data.

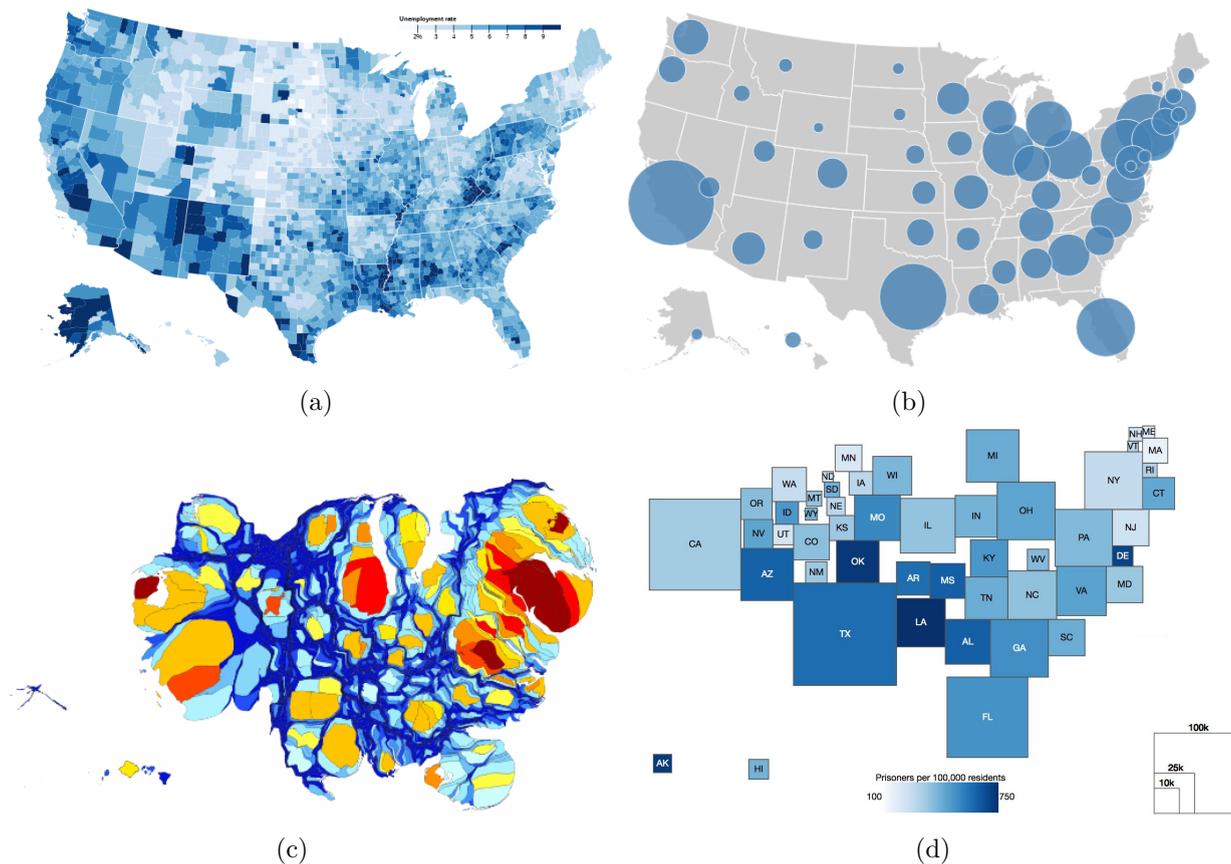


Figure 2.1.: Various types of thematic maps. **(a)** Choropleth map indicating unemployment rates.¹ **(b)** Proportional symbol map showing population densities.² **(c)** Area cartogram with counties sized according to their share in the total U.S. gross domestic product.³ **(d)** Dorling-like cartogram showing for each state the number of prisoners per 100,000 citizens; darker colors indicate a higher percentage.⁴

2.1.2. Map Metaphors

Maps of imagined places already have a long tradition, and picturesque hand-drawn maps are a special gem in many fantasy books. In 1938, the German illustrator Alphons Woelfle depicted the German publishing and book trade in a beautiful allegory, a hand-drawn lithography in baroque manner with antique typeface [Zim09]. The *Karte des Bücherlandes* is obviously not strictly based on underlying data, yet visually engaging.

Since the beginning of the millennium, with higher computing capabilities, sophisticated algorithms, and computational visualization methods allowing analysis of large-scale document datasets, *knowledge or science maps* have become popular in bibliometrics and scientometrics. An in-depth review on a variety of knowledge mapping techniques is provided in [BCB03]. The authors of [BKB05] visualized patterns of scientific influence within and across disciplines based on citation data of around a million articles published in natural and social science journals in the year 2000 (Figure 2.3). Seven “hub disciplines” were identified (bold labels). In this map, the locations of the sciences can be taken more or less at face value, for instance, Electrical Engineering is situated between Mathematics and Physics, or Neurology can be found between Medicine and Psychology. The more

¹<https://bl.ocks.org/mbostock/4060606>

²<https://bl.ocks.org/mbostock/4342045>

³<http://metrocosm.com/map-us-economy/>

⁴<http://www.cyrusbrien.com/datavisualizations/>

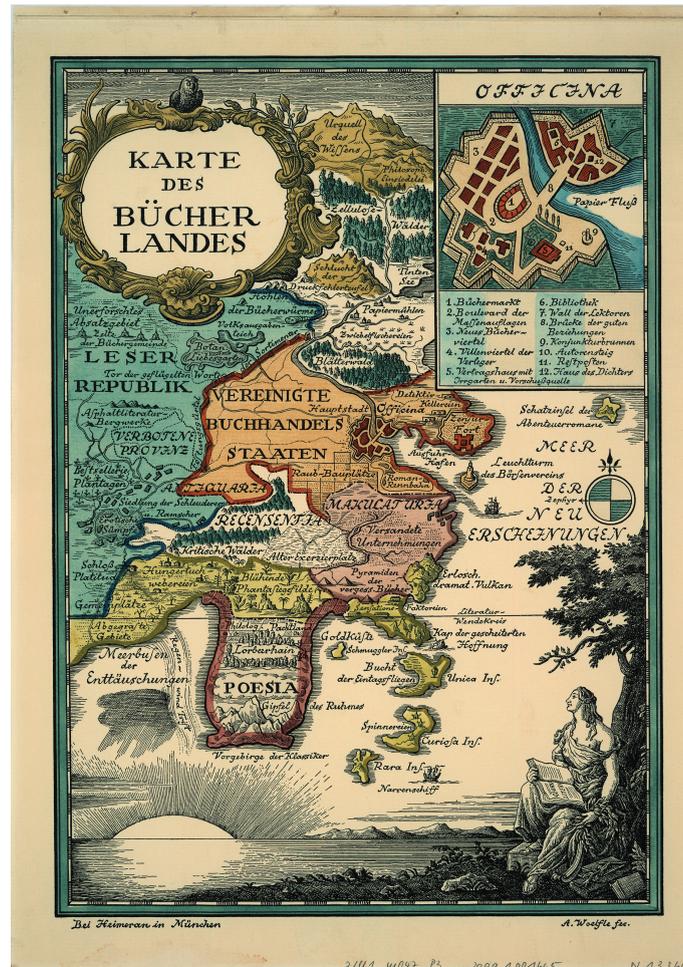


Figure 2.2.: “Karte des Bücherlandes” by Alphonse Woelfle, 1938.

insular fields tend to lie on the outside of the map, whereas the more interdisciplinary fields gravitate towards the center. Boyack states the potentially great benefits, such maps can have: “Our interest in mapping science stems from a desire to understand the inputs, associations, flows, and outputs of the Science and Technology [...] enterprise in a detailed manner that will help us guide that enterprise [...] in more fruitful directions” ([BKB05], p.352).

A trade-off between structural accuracy on the one hand, and clarity and visual appeal on the other hand is inevitable, as these criteria directly contradict each other in many cases.

The work presented here was largely inspired by the GMAP Algorithm of Stephen Kobourov and colleagues at the University of Arizona [GHK10], who place the highest emphasis on generating visually appealing and intuitive representations of possibly huge datasets, which “borrow map-related cognitive concepts: [...] items within a country are similar to each other; areas separated by a mountain range are difficult to connect; islands might have atypical qualities, etc.” ([GHKV09], p.1). GMAP is a means to visualize all kinds of relational datasets by extracting graphs from them. 2D drawings of graphs are obtained by embedding and layouting algorithms, which display underlying proximity information by tendentiously putting similar items closer together. Following a structural analysis of the data by means of clustering algorithms, it stands to reason to define clusters explicitly in the visualization as “national borders” and colouring the regions. On their website⁵, the authors provide the possibility to use their algorithm for one’s own graphs, represented in

⁵<http://gmap.cs.arizona.edu/>

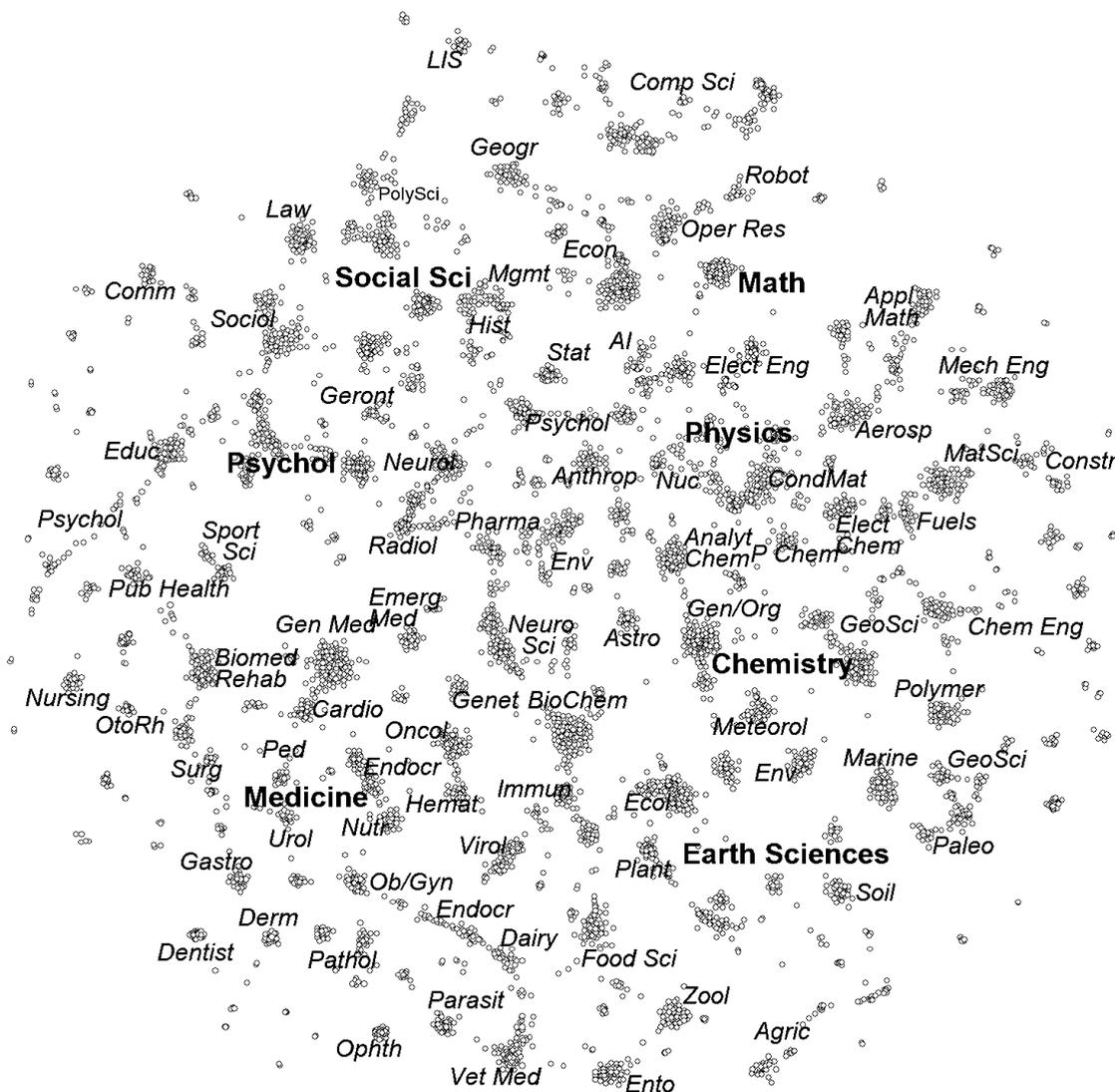


Figure 2.3.: A science map of 7,121 journals published in 2000 [BKB05].

the DOT graph description language.⁶ You can combine two spring model layout algorithms with four different clustering algorithms or directly integrate your own positions and cluster memberships in the DOT-file. The authors also offer an option to ensure contiguity and disjointness of the countries (by selecting the `contiguous` clustering option as opposed to `fragmented`) [EHKP15]. Figure 2.4 shows our collected opinion sample (described in Subsection 3.3.2) visualized with GMAP, using *Scalable Force Directed Placement* [Hu05], a multilevel spring-electrical algorithm suitable for large networks, combined with our own clustering (detailed in Chapter 4).

2.2. Tools for Argument & Opinion Analysis

The majority of visualization tools concerning debates aim to aid the process of forming an opinion by depicting the structure of arguments advanced. An example of a web-based interactive visualization is the online platform Kialo.⁷ It provides trees and sunburst diagrams that clearly visualize the topology of a debate, allowing in-depth exploration

⁶https://graphviz.gitlab.io/_pages/doc/info/lang.html

⁷<https://www.kialo.com>

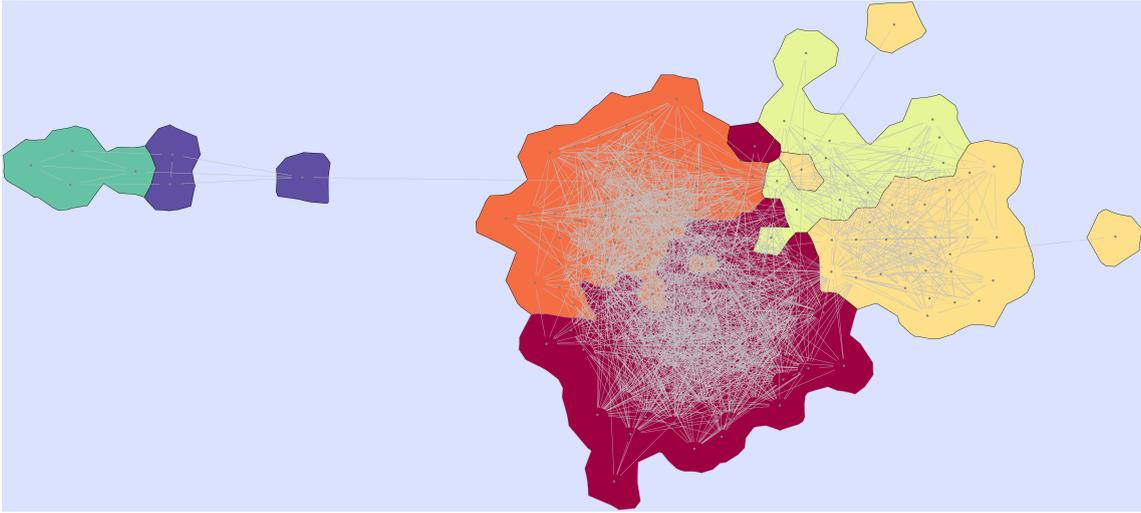


Figure 2.4.: Our opinion dataset visualized with the GMAP algorithm, using the layout option `sfdp`, and the clustering option `fragmented` in combination with our own clustering.

of and deliberation on diverse points of view on a complex issue. Examples are given in Figure 2.5. Pro-arguments are coloured green and con-arguments are coloured red or orange. By hovering or clicking on the nodes of the tree or respectively the rays of the sunburst diagram, the corresponding arguments can be shown. In case of the tree diagram, by clicking on a node you can expand the tree to see pros and cons to the argument you have just clicked on, while the sunburst diagram shows the whole argument structure in one view.

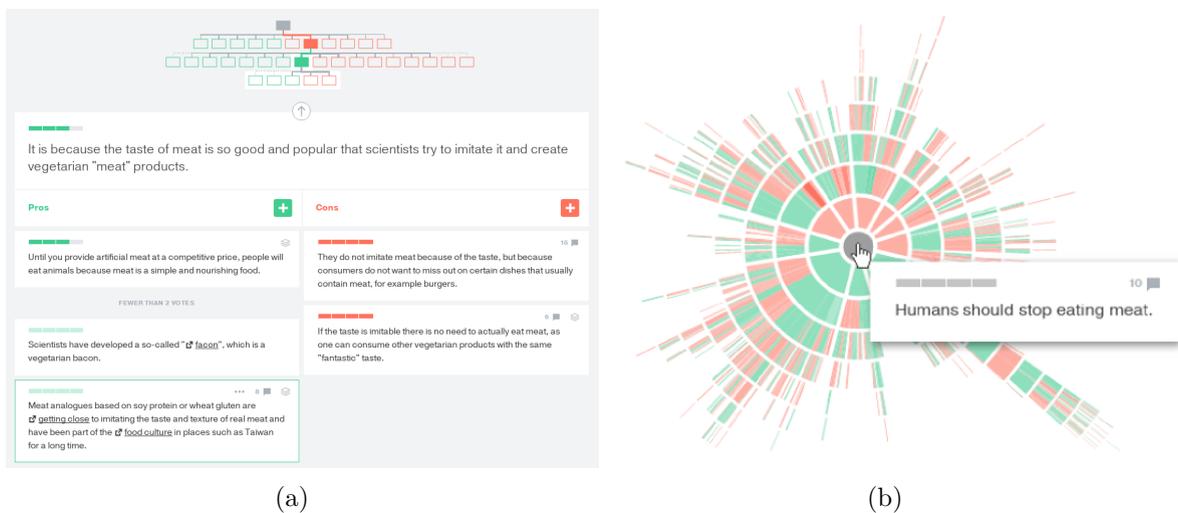


Figure 2.5.: Visualizations on the debate about the proposition **Humans should stop eating meat.**⁸ (a) Tree. (b) Sunburst diagram.

A popular tool for decision making in Germany is WAHL-O-MAT developed by the *Bundeszentrale für politische Bildung* [fpB]. It is supposed to help voters during parliamentary elections to determine which party conforms to the greatest extent with their political position. Users are presented with 38 statements for which they can choose between the answers **agree**, **disagree**, and **neutral**, or they can skip the statement. The political

⁸<https://www.kialo.com/eating-meat-is-wrong-1229/>

parties are presented with the very same survey, such that it can be calculated to which degree a user's answers conform with the answers of the different parties. The users are additionally offered the possibility to assign a double weight to statements that are particularly important to them. Results are presented in the form of a bar chart which indicates the percentage to which the parties' positions conform with the user's position. They are also offered a detailed comparison, showing their answers to individual statements alongside the parties' answers as well as their explanation of their choices.

Familiar methods for mapping public opinion data, as they are collected in grand amounts for example during political elections, are the ubiquitous pie and bar charts. However, the map visualizations mentioned in Subsection 2.1.1 are also adequate for spatial opinion data.

3. Modelling the Opinion Space

As explained in Section 1.2, the input to the OPMAP algorithm is a graph, where the set of vertices corresponds to opinions held within a debate. An opinion sample can be elicited on the basis of a *logical reconstruction* of the debate, i.e. a formalization of the advanced arguments. The logical reconstruction of our use case, the Veggie-Debate, was kindly provided by Gregor Betz. The following section describes the Veggie-Debate and employs it as an example to explain and illustrate the argumentation-theoretic foundations of OPMAP as stated in [Bet17]. Afterwards, quantitative distance measures which we have considered are described and the final section of this chapter presents our opinion datasets.

3.1. Basic Argumentation Theory

In the following, let S be a finite set of natural language sentences, which is closed under negation, that is $\forall s \in S : s = \neg\neg s$. More precisely, S can be represented by $S^+ \cup S^-$, a set of unnegated sentences S^+ conjoined with their negations S^- . Let $|S| = 2N$.

An *argument* is a series of *premisses* supporting a *conclusion*. Formally, an argument is a tuple $a = (P, c)$ with

- a set of premisses $P \subseteq S$
- a conclusion $c \in S$

In our interpretation, the Veggie-Debate comprises eight core claims:

1. [Meat-OK]: There exist meat and animal products which one is allowed to eat.
2. [Eat-what-you-want]: One may eat meat and other animal products of any kind.
3. [Organic-meat]: You should eat meat and animal products only from sustainable, species-appropriate manufacturers.
4. [No-mass-farming]: One must not eat meat produced in modern mass farming facilities.
5. [Strict-veggie]: One must not eat meat at all.
6. [Strict-vegan]: One must not eat animal products at all.
7. [Less-meat]: One should reduce the consumption of meat.
8. [Less-animal]: One should reduce the consumption of animal products.

Conclusions can be either core claims or other premisses. Thus, claims, premisses and conclusions are *sentences* while arguments are *sets of sentences*.

Logical *support and attack relations* exist between arguments, i.e. an argument may *contradict* or *imply* another argument:

Let $a = (P_a, c_a)$ and $b = (P_b, c_b)$ be two arguments.

- a supports b iff $c_a \in P_b$
- a attacks b iff $\neg c_a \in P_b$

A debate can now be formally defined as a *dialectical structure* on S . It is given by a triple $\tau = (T, A, U)$ of arguments T and the support and attack relations A, U among them. The argumentative analysis identifies pro- and con-arguments in favour of or against the core claims. In order to facilitate grasping logical relations between a multitude of arguments, we employ *argument maps*, visual representations that neatly display how arguments are related. Such maps may be automatically generated from *Argdown* documents.¹ Argdown is a syntax that is used to encode logical relations in an argument analysis.² The argument mapping technique based on Argdown was developed within the ARGUNET project³ by DebateLab at KIT's Institute of Philosophy.⁴ Figure 3.1 shows the logical relations between the core claims⁵ stated above. Green arrows indicate support relations, red arrows indicate attack relations. For example, [Eat-what-you-want] contradicts [No-mass-farming], which means one cannot agree to both claims without maintaining a logically inconsistent opinion. On the other hand, [Strict-vegan] implies [Strict-veggie], which means that if one accepts the former, one automatically accepts the latter as well.

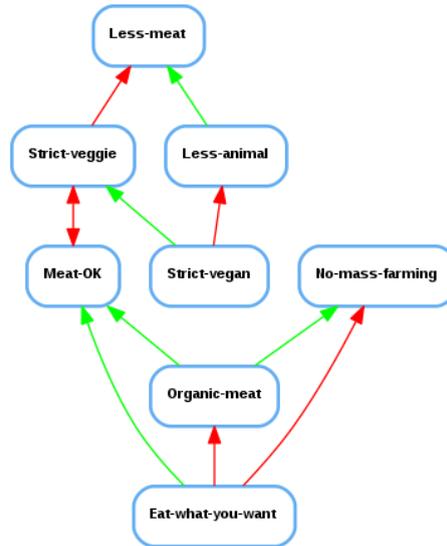


Figure 3.1.: Argument map of the core claims advanced in the Veggie-Debate.

The advanced arguments can be grouped into the following categories:

- Culinary considerations
- Health considerations
- Financial considerations

¹<http://christianvoigt.github.io/argdown>

²The core claims stated above as well as the examples to come are formatted according to the Argdown conventions.

³<http://www.argunet.org/>

⁴<http://debatelab.philosophie.kit.edu/>

⁵With slight abuse of terminology, an individual sentence may be actually referring to a more detailed argument, of which it is the conclusion.

- Naturalness and normality considerations
- Climate change considerations
- Arguments concerning nature conservation
- Animal rights arguments
- World nutrition considerations
- Arguments concerning personal autonomy

In order to calculate a quantitative measure of distance on the space of opinions, a formalization of opinions, which allows such a calculation, is required. This is given for the following formalization:

Definition 3.1. Position/Opinion

A position \mathcal{A} , which is maintained in a debate comprised by sentences S , is a truth-value assignment to sentences $S_D \subseteq S$

$$\mathcal{A} : S_D \rightarrow \{true, false\}$$

A position \mathcal{A} with domain S_D is called

- *complete* $\Leftrightarrow S_D = S$
- *partial* $\Leftrightarrow S_D \subset S$
- *atomic* $\Leftrightarrow |S_D| = 1$
- *tautological* $\Leftrightarrow S_D = \{\}$

We take into account only logically *consistent* positions. Let $\tau = (T, A, U)$ be a dialectical structure on S . A complete position \mathcal{A} ($S_D = S$) on τ is consistent if

- $\forall p, \neg p \in S : \mathcal{A}(p) \neq \mathcal{A}(\neg p)$ (minimal consistency)
- $\forall a \in T : (\forall p \in P_a : \mathcal{A}(p) = true) \Rightarrow (\mathcal{A}(c_a) = true)$

A partial position \mathcal{B} on τ is (minimally) consistent if there exists a complete and (minimally) consistent position which extends \mathcal{B} .

That is, a consistent position cannot accept and reject a premise at the same time, and if all premises of an argument are accepted then the conclusion must be accepted as well.

In order to represent an opinion, we assume that the pool of sentences S is numbered. The full sentence pool of the Veggie-Debate is given in Appendix Section A. Example statements, in case of the Veggie-Debate are:

- s_1 <Reduce harmful fats>: Animal fats are unhealthy. Therefore one should reduce the consumption of animal products as much as possible.
- s_2 <Meat dishes are cultural heritage>: Completely abandoning meat and/or animal products would put an end to a centuries-old, cultural tradition - the art of cooking."
- s_3 <Organic only for rich people>: Many people cannot afford organic products, especially animal products from sustainable agriculture. Organic foods are only for rich people.

A person can give their opinion by either agreeing to, rejecting or being indifferent towards each of the statements. For instance, someone could agree to statement 1, reject statement 2 and be indifferent towards statement 3. This opinion can be formalized as a set: $\{s_1 \rightarrow true, s_2 \rightarrow false, s_3 \rightarrow neutral\}$ or $\{s_1, \neg s_2\}$ for short. If $|S^+ \cup S^-| = 2N$, all possible positions on S can be organized as a *directed acyclic graph* (DAG) with 2^{2N} nodes as shown in Figure 3.2. The edges of a directed graph $D = (V, E)$ are ordered pairs

$e = (u, v)$. A graph is acyclic if one cannot start at any vertex and follow a consistently-directed sequence of edges that eventually loops back to the starting vertex.

Directed edges are typically visualized as arrows. The only source in the position graph is the tautological position and the only sink is a position which maintains all sentences in the set as well as their negations, obviously an inconsistent position. If there is a directed *path* from a position p_i to a position p_j in the graph, p_j implies p_i .

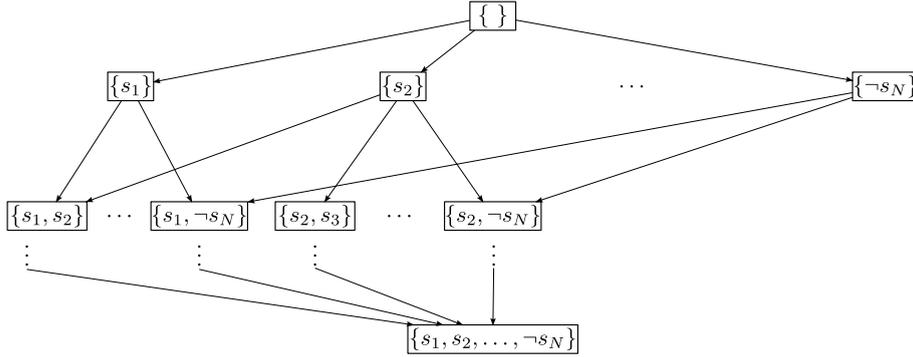


Figure 3.2.: All possible positions organized in a DAG.

The following properties hold as consequences of the definitions given above:

1. The tautological position is consistent.
2. Every restriction of a consistent position is itself consistent.
3. Every extension of an inconsistent position is inconsistent. (Every vertex in the DAG that has a path from an inconsistent position is itself an inconsistent position.)

3.2. Probabilistic Epistemology behind OpMap

For different kinds of data, different *distance metrics* may be suitable. In an everyday understanding, when we think of distance, we usually think of the length of the shortest straight line through space between two points (the *Euclidean* distance). OPMap projects *conceptual* distances between opinion data to Euclidean distances on a 2D-map. A location in this map represents a particular opinion. Distances between two locations are proportional to the extent to which the corresponding opinions *cohere* with each other. The closer two opinion points in the map, the better they fit together.

As stated in Section 1.2, we model the space of existing opinion vectors as a weighted graph. The vertices in this graph are the opinion vectors. The weight function, which assigns a numerical value to an edge that connects two opinion nodes, should reflect the semantic similarity or *coherence* between these opinions. Coherence is a controversial and often vague epistemological concept. Shogenji, in his pioneering work on formal analyses of coherence [Sho99] from almost two decades ago, states the general idea, that coherent beliefs (or opinions) “hang together”. Coherence is not synonymous with consistency (the absence of contradictions) but rather refers to relations of mutual support or non-deductive agreement. Shogenji states two further properties: First, that coherence comes in degree—beliefs that are more likely to be true together have a higher degree of coherence. Second, that coherence is symmetrical—that is, the order of beliefs is irrelevant to the assessment of their coherence. Since then, various distance measures to quantify this intuitive notion have been proposed. These coherence measures consider all implications of an original opinion, i.e. every position node is *deductively closed* before coherence is calculated. This means that a position is the union of the set of sentences on which an explicit stance was taken, and the set of all statements that can be deduced from the statements in the former set.

Definition 3.2. Deductive Closure

Let $\tau = (T, A, U)$ be a dialectical structure on S and let \mathcal{A} be a position. Γ^τ shall denote the set of all consistent and complete positions on τ . If \mathcal{A} is a position on τ , let $\Gamma_{\mathcal{A}}^\tau$ be the set of consistent and complete positions that extend \mathcal{A} :

$$\Gamma_{\mathcal{A}}^\tau := \{\mathcal{X} \in \Gamma^\tau : \mathcal{X} \supset \mathcal{A}\}$$

The deductive closure $\overline{\mathcal{A}}$ of \mathcal{A} is given by

$$\overline{\mathcal{A}} := \bigcap_{\mathcal{X} \in \Gamma_{\mathcal{A}}^\tau} \mathcal{X}$$

Deductive reasoning is based on our use of language. For example, if you agree to the statements *If it rains, the ground gets wet.* and *The ground is not wet.* you thereby agree that *It is not raining.*

The following subsections describe the measures we tested to make the opinion space more or less a metric space. As a simple benchmark we use three variants of the *edit distance* and one other measure, which are cheap to calculate. Furthermore, we experimented with two more sophisticated and more meaningful distance measures used in Bayesian epistemology [Tal16]. All the measures are applicable to arbitrary, even inconsistent, positions.

3.2.1. Benchmarks

An *edit distance* quantifies the similarity of two strings (or words) by counting the minimum number of insertions, deletions and substitutions required to transform one string into the other [Nav01]. We use the *Hamming distance*, which allows only substitutions and is therefore applicable only to words of fixed length. It is given by the number of positions at which the corresponding symbols are different [Nav01].

Definition 3.3. Hamming distance

Let Σ be a finite alphabet and $\mathcal{X} = (\sigma_1^{\mathcal{X}}, \dots, \sigma_N^{\mathcal{X}}), \mathcal{Y} = (\sigma_1^{\mathcal{Y}}, \dots, \sigma_N^{\mathcal{Y}}) \in \Sigma^N$ two words on that alphabet. The normalized Hamming distance is given by

$$nEditSim(\mathcal{X}, \mathcal{Y}) := \frac{1}{N} \cdot |\{j \in \{1, \dots, N\} | \sigma_j^{\mathcal{X}} \neq \sigma_j^{\mathcal{Y}}\}|$$

Say we have a set of five sentences and two positions

$$\begin{aligned} \mathcal{A} &= \{s_1, _, s_3, \neg s_4, \neg s_5\} \\ \mathcal{B} &= \{_, \neg s_2, s_3, \neg s_4, s_5\} \end{aligned}$$

In this case the simple Hamming distance is $nEditSim(\mathcal{A}, \mathcal{B}) = \frac{3}{5}$, as stances towards three sentences have to be edited in order to make the two positions equal.

Note that the Hamming distance satisfies all axioms of a *metric space*. A metric space is a tuple (M, d) where M is a set and d is a metric on M , i.e. a function

$$d : M \times M \rightarrow \mathbb{R}, \text{ such that}$$

$d(x, y) \geq 0$	non-negativity
$d(x, y) = 0 \Leftrightarrow x = y$	identity of indiscernibles
$d(x, y) = d(y, x)$	symmetry
$d(x, z) \leq d(x, y) + d(y, z)$	triangle inequality

From an argumentation theoretic point of view, it makes sense to introduce penalty weights, such that explicit disagreement on a statement is weighted more than suspension of judgement vs. explicit stance on a statement. Therefore, we use a normalized and weighted variant of the Hamming distance.

Definition 3.4. *The **weighted normalized Hamming distance** is given by*

$$gHamSim := \frac{1}{|S|} \sum_{s_j \in S} d(s_j^A, s_j^B) \in [0.0, 1.0] \text{ where}$$

$$d(s_j^A, s_j^B) = \begin{cases} 2 & \text{if } (s_j^A = s_j \wedge s_j^B = \neg s_j) \vee (s_j^A = \neg s_j \wedge s_j^B = s_j) \text{ (conflict penalty)} \\ 1 & \text{if } (s_j^A = s_j \wedge s_j^B = _) \vee (s_j^A = \neg s_j \wedge s_j^B = _) \text{ (contraction penalty)} \\ 1 & \text{if } (s_j^A = _ \wedge s_j^B = s_j) \vee (s_j^A = _ \wedge s_j^B = \neg s_j) \text{ (expansion penalty)} \\ 0 & \text{if } s_j^A = s_j^B \end{cases}$$

Consider the example from above:

$$\begin{aligned} \mathcal{A} &= \{s_1, _, s_3, \neg s_4, \neg s_5\} \\ \mathcal{B} &= \{_, \neg s_2, s_3, \neg s_4, s_5\} \\ gHamSim(\mathcal{A}, \mathcal{B}) &= \frac{1 + 1 + 0 + 0 + 2}{5} = \frac{4}{5} \end{aligned}$$

Besides the classical Hamming distance $nEditSim$ and the global Hamming distance $gHamSim$ we use a third variant, the core Hamming distance $cHamSim$ which is defined analogous to $gHamSim$ but considers only the core claims featured in a position.

Definition 3.5. $S_C \subset S$ shall denote the set of core claims.

$$cHamSim := \frac{1}{|S_C|} \sum_{s_j \in S_C} d(s_j^A, s_j^B) \in [0.0, 1.0]$$

We use one other cheap measure, the normalized closeness.⁶

Definition 3.6. Normalized Closeness

Let n_{Cont} be the number of contractions, n_{Exp} the number of expansions and n_{Conf} the number of conflicts between two positions \mathcal{A} and \mathcal{B} and let p_{Cont} , p_{Exp} , and p_{Conf} be the penalty weights associated with these cases. The normalized closeness is then given by

$$nClos := 1 - \left[\frac{p_{Cont} \cdot n_{Cont} + p_{Exp} \cdot n_{Exp} + p_{Conf} \cdot n_{Conf}}{\max(p_{Cont}, p_{Exp}, p_{Conf}) \cdot |S|} \right]^2$$

3.2.2. Measures based on the Degree of Justification

This subsection introduces two probabilistic coherence measures, the *degree of mutual coherence* and the *Shogenji coherence measure*, which are based on *degrees of justification* [Bet12]. The degree of justification of a position \mathcal{A} held within a debate is the ratio of the number of complete and consistent positions that extend \mathcal{A} to the total number of possible complete and consistent positions. Let $\tau = (T, A, U)$ be a dialectical structure on S and \mathcal{A} a position on τ . Let Γ^τ denote the set of all complete and consistent positions on τ . The degree of justification of \mathcal{A} in τ is given by

⁶The terms *distance measure* and *closeness* or *similarity measure* are used somewhat interchangeably, as they are directly related: Similarity corresponds to 1 - distance.

$$DOJ(\mathcal{A}) := \sigma_{\mathcal{A}}/\sigma, \text{ where}$$

- $\sigma := |\Gamma^\tau|$
- $\sigma_{\mathcal{A}} := |\{\mathcal{X} \in \Gamma^\tau : \mathcal{X} \supset \mathcal{A}\}|$

Intuitively, this means that positions with little attacks on them are well justified. The *degree of partial entailment* is a conditional degree of justification. For two positions \mathcal{A}, \mathcal{B} the degree of partial entailment of \mathcal{A} by \mathcal{B} is given by

$$DOJ(\mathcal{A}|\mathcal{B}) := \sigma_{\mathcal{A},\mathcal{B}}/\sigma_{\mathcal{B}}$$

where $\sigma_{\mathcal{B}} = |\Gamma_{\mathcal{B}}|$, i.e. the size of the set of all consistent and complete positions that extend \mathcal{B} and $\sigma_{\mathcal{A},\mathcal{B}} = |\Gamma_{\mathcal{A}} \cup \Gamma_{\mathcal{B}}|$, i.e. the size of the intersection of the sets of all consistent and complete positions that extend \mathcal{A} or \mathcal{B} respectively.

Now we can define the degree of mutual coherence, a measurement of how well two opinions \mathcal{A}, \mathcal{B} support each other [DM07].

Definition 3.7. Degree of Mutual Coherence

The degree of mutual coherence is given by

$$MutCoh_\tau(\mathcal{A}, \mathcal{B}) := \frac{1}{2} \cdot (S(\mathcal{A}, \mathcal{B}) + S(\mathcal{B}, \mathcal{A})) \in [-1, 1]$$

Here, the overall support of a position \mathcal{A} by another position \mathcal{B} is given by

$$S(\mathcal{A}, \mathcal{B}) := \frac{1}{2^{|\mathcal{A}|}} \sum_{\mathcal{X} \subseteq \mathcal{A}} Conf_\tau(\mathcal{X}, \mathcal{B})$$

In turn, $Conf_\tau$ is a suitable Bayesian confirmation measure that indicates to which extent \mathcal{X} is confirmed by \mathcal{Y} . We use the *Kemeny-Oppenheim measure*.

Definition 3.8. Kemeny-Oppenheim Measure

For two positions \mathcal{X}, \mathcal{Y} the Kemeny-Oppenheim measure of confirmation is given by

$$Conf_\tau^{KO}(\mathcal{X}, \mathcal{Y}) := \begin{cases} \frac{DOJ(\mathcal{Y}|\mathcal{X}) - DOJ(\mathcal{Y}|\neg\mathcal{X})}{DOJ(\mathcal{Y}|\mathcal{X}) + DOJ(\mathcal{Y}|\neg\mathcal{X})} & \text{if } \mathcal{Y} \not\models \mathcal{X} \wedge \mathcal{Y} \not\models \neg\mathcal{X} \\ 1 & \text{if } \mathcal{Y} \models \mathcal{X} \\ -1 & \text{if } \mathcal{Y} \models \neg\mathcal{X} \end{cases}$$

MutCoh has various desirable features as a measure of coherence: It takes the maximum value 1 for equivalent positions, 0 for fully independent positions, and negative values for inconsistent positions. However, one major drawback of *MutCoh* is its computational complexity. Therefore we investigated another, computationally cheaper measure, the one originally proposed by Shogenji [Sho99].

Definition 3.9. Shogenji Measure

$$ShoCoh_\tau(\mathcal{A}, \mathcal{B}) := \frac{DOJ(\mathcal{A} \wedge \mathcal{B})}{DOJ(\mathcal{A}) \times DOJ(\mathcal{B})}$$

where the conjoined position is $\mathcal{A} \wedge \mathcal{B} = \{s_1^{\mathcal{A}}, \dots, s_n^{\mathcal{A}}\} \cup \{s_1^{\mathcal{B}}, \dots, s_m^{\mathcal{B}}\}$.

$ShoCoh_\tau$ compares the joint probability of the statements of \mathcal{A} and \mathcal{B} (the numerator) with the value that this joint probability would take if these statements were statistically independent of one another (the denominator). All but the Shogenji measure yield complete graphs.

In order to clarify some of the concepts introduced above and especially mutual coherence, consider a simple example: Let S be a pool of three sentences, which, for simplicity, do not contradict or imply each other, such as

$$S = \{s_1 = \text{Going to the cinema is fun.}, \\ s_2 = \text{Popcorn is delicious.}, \\ s_3 = \text{Inception is a great movie.}\}$$

This means, that every complete position on that pool is also a consistent position. Consider two positions on this pool, $\mathcal{A} = \{s_1, \neg s_2\}$ and $\mathcal{B} = \{s_2, s_3\}$. The deductive closure

$$\overline{\mathcal{A}} = \{s_1, \neg s_2, s_3\} \cap \{s_1, \neg s_2, \neg s_3\} = \{s_1, \neg s_2\} = \mathcal{A},$$

and for the other position also holds $\overline{\mathcal{B}} = \mathcal{B}$. When calculating $S(\mathcal{A}, \mathcal{B})$ we need to consider all subsets of \mathcal{A} , which are $\{\}, \{s_1\}, \{\neg s_2\}, \{s_1, \neg s_2\}$. Using the Kemeny-Oppenheim measure we obtain

$$S(\mathcal{A}, \mathcal{B}) = \frac{1 + 0 + (-1) + (-1)}{2^2} = -0.25$$

The summands are the respective Kemeny-Oppenheim degrees of confirmation:

- $Conf_\tau^{KO}(\{\}, \{s_2, s_3\}) = 1$, as $\{s_2, s_3\} \models \{\}$, that is, \mathcal{B} logically entails the tautological position. (It is entailed by any other position as well.)
- To calculate $Conf_\tau^{KO}(\{s_1\}, \{s_2, s_3\})$ we need to calculate two degrees of partial entailment:

$$DOJ(\{s_2, s_3\}|\{s_1\}) = \sigma_{\{s_2, s_3\}, \{s_1\}} / \sigma_{\{s_2, s_3\}} \\ = 0.5 \text{ where}$$

$$\sigma_{\{s_2, s_3\}, \{s_1\}} = |\{\{s_1, s_2, s_3\}, \{s_1, \neg s_2, s_3\}\}| \\ \cap |\{\{s_1, s_2, s_3\}, \{s_1, \neg s_2, s_3\}, \{s_1, s_2, \neg s_3\}, \{s_1, \neg s_2, \neg s_3\}\}| \\ = |\{\{s_1, s_2, s_3\}\}| = 1 \text{ and}$$

$$\sigma_{\{s_2, s_3\}} = |\{\{s_1, s_2, s_3\}, \{s_1, \neg s_2, s_3\}\}| = 2 \text{ and analogously}$$

$$DOJ(\{s_2, s_3\}|\{\neg s_1\}) = \sigma_{\{s_2, s_3\}, \{\neg s_1\}} / \sigma_{\{s_2, s_3\}} \\ = 0.5$$

$$\text{Therefore } Conf_\tau^{KO}(\{s_1\}, \{s_2, s_3\}) = \frac{0.5 - 0.5}{0.5 + 0.5} = 0$$

- $Conf_\tau^{KO}(\{\neg s_2\}, \{s_2, s_3\}) = -1$, as $\{s_2, s_3\} \models \{\neg \neg s_2\} = \{s_2\}$, that is, \mathcal{B} entails the negation of $\{\neg s_2\}$.
- $Conf_\tau^{KO}(\{s_1, \neg s_2\}, \{s_2, s_3\}) = -1$, as the two positions directly contradict each other in one sentence and are therefore inconsistent.

Analogously we obtain

$$S(\mathcal{B}, \mathcal{A}) = \frac{1 + (-1) + 0 + (-1)}{2^2} = -0.25$$

So according to Definition 3.7

$$MutCoh_\tau(\mathcal{A}, \mathcal{B}) = \frac{1}{2} \cdot (-0.25 - 0.25) = -0.25$$

This value makes sense because \mathcal{A} and \mathcal{B} are not consistent, as they explicitly disagree on one statement and each of the positions suspends judgement on one of the two remaining statements.

Name	Sample size	Characteristics
2017-04-27T15-31-02	265	Random Sampling: $\{\{500, \{0.8, 0.1, 0.1\}\}, \{300, \{0.6, 0.2, 0.2\}\}, \{200, \{0.4, 0.3, 0.3\}\}\}$ Core Sentences: $\{\}$
2017-05-02T19-56-11	97	Random Sampling: $\{\{10000, \{0.4, 0.3, 0.3\}\}\}$
2017-05-02T22-56-28	137	Random Sampling: $\{\{300, \{0.9, 0.05, 0.05\}\}\}$ Core Sentences: $\{1, 2\}$
2017-05-03T08-58-04	242	Random Sampling: $\{\{300, \{0.9, 0.05, 0.05\}\}\}$
2017-05-03T15-13-56	255	Random Sampling: $\{\{500, \{0.8, 0.1, 0.1\}\}, \{300, \{0.6, 0.2, 0.2\}\}, \{200, \{0.4, 0.3, 0.3\}\}\}$
2017-05-03T21-51-03	113	Random Sampling: $\{\{10000, \{0.4, 0.3, 0.3\}\}\}$
2017-05-03T22-44-20	133	Random Sampling: $\{\{300, \{0.9, 0.05, 0.05\}\}\}$ Core Sentences: $\{1, 2\}$
2017-05-04T10-35-40	241	Random Sampling: $\{\{300, \{0.9, 0.05, 0.05\}\}\}$
2017-05-04T20-01-57	273	Random Sampling: $\{\{500, \{0.8, 0.1, 0.1\}\}, \{300, \{0.6, 0.2, 0.2\}\}, \{200, \{0.4, 0.3, 0.3\}\}\}$ Core Sentences: $\{\}$
2017-05-13T05-38-29	200	Ensemble with 5 clusters à 40 positions for different core claims.

Table 3.1.: Description of the simulated datasets.

3.3. Opinion Data

In order to quickly start testing various clustering methods opinion samples can be generated in a simulation while collecting real opinions using an online survey.

3.3.1. Simulated Data

We tested with ten artificial opinion datasets, nine of which were obtained by random sampling, partly with additional constraints. Table 3.1 describes these datasets. For example 2017-05-03T15-13-56 was generated as follows:

- 500 positions were generated, such that

80%	of positions suspend judgement on	80%	of sentences
10%	”	60%	”
10%	”	40%	”
- 300 positions were generated, such that

60%	of positions suspend judgement on	80%	of sentences
20%	”	60%	”
20%	”	40%	”
- 200 positions were generated, such that

40%	of positions suspend judgement on	80%	of sentences
30%	”	60%	”
30%	”	40%	”

The second parameter specifies core claims on which each position has to take explicit stance. From the resulting 1000 positions, only the consistent ones, in this case 255, are considered.

From each of these datasets, we derive six opinion-graphs that use as edge weights the distance metrics described in Section 3.2. The last sample is supposed to mimic the anticipated structure of the real survey data. Figure 3.3 visualizes this sample for original and deductively closed opinions. Opinions are plotted on the ordinate, statements are

plotted on the abscissa. A green box indicates agreement and a red box disagreement with the statement, white indicates suspension of judgement.

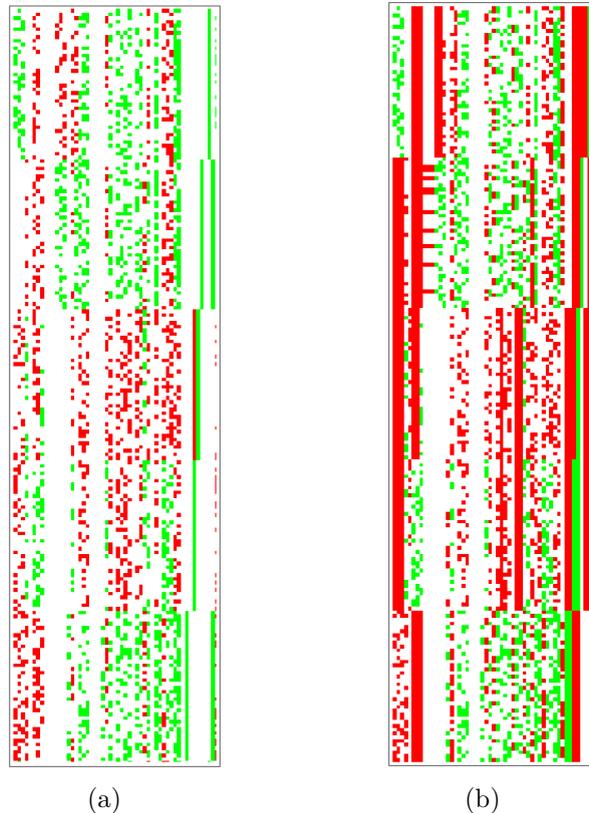


Figure 3.3.: Structure of the stylized five-cluster opinion sample 2017-05-13T05-38-29. (a) Original opinions. (b) Deductively closed opinions.⁷

3.3.2. Empirical Data

In order to quickly gather a relatively large set of opinions, we conducted an online survey. In our survey design, we first elicit the participant's core opinion by asking: Not considering what your actual eating habits are, how do you think should a good diet look like? and then offering a selection of five options, which correspond roughly to the core claims:

1. Strictly vegan: You should not eat animal products.
2. Strictly vegetarian: You should not eat meat, but other animal products are ok.
3. Restricted meat consumption: You should restrict meat consumption, e.g. by only buying organic products or by eating meat only occasionally.
4. Omnivore: Everybody can eat whatever they like.
5. None of the options

The question intentionally asks the participant to not consider his or her *actual* eating behaviour, as we want to enquire their opinion on how a good diet should *ideally* be like. On the next page, the participant is asked: What are the reasons for your choice? Please indicate for the following arguments if you agree, disagree or are indifferent

⁷This figure was generated by Gregor Betz in Mathematica.

towards/have no information about them. Afterwards, arguments that are *logically consistent* with the selected option are presented and the participant can indicate for each if he or she agrees, rejects, or is indifferent or respectively has no information about the statement. If the first option is selected, arguments for a strictly vegan diet as well as some arguments for a vegetarian diet are presented. If the second option is selected, arguments for a strictly vegetarian and some arguments for a vegan diet are presented. The **restricted** option then enquires how one should restrict meat consumption, presenting the core claims [Organic-meat], [Less-meat], and [Less-animal]. Afterwards, suitable arguments in line with these core claims as well as [Meat-OK] are presented. The second to last option then lists arguments for [Eat-what-you-want] and [Meat-OK]. The last option is a catch-all case where the full set of statements is presented. Here, the participant is asked **Does none of the options reflect your opinion? Or is it that you simply do not have an opinion on that question? In the following, you can indicate your stance towards individual statements nonetheless.**

Table A.2 in Appendix Section A shows the complete sets of sentences presented in each survey option. In case of some of the options, it is possible to give logically inconsistent opinions (for example if a participant agrees that meat consumption should be reduced as much as possible but disagrees that the consumption of animal products should be restricted). Inconsistent options are not considered on the map.

The online survey yielded a sample of 210 opinions, two of which were discarded as they were inconsistent. Appendix Section A includes the overall acceptance/rejection rates to each statement.

As outlined in Chapter 1, the visualization method consists of three steps.

The first step is the clustering of the opinion-graph to determine the “nationality” of every opinion. The second step is to draw the graph in the 2D plane, so that the opinions of the same nationalities lie close to each other. The last step is to draw the boundaries of the countries. In the following chapters we describe the algorithms behind these three steps.

4. Finding Structure

Choosing a clustering algorithm is a fundamental design decision in the mapping algorithm. It is crucial to match the clustering algorithm to the embedding algorithm, as vertices that belong to the same clusters need to be assigned to geometrically close positions. Some clustering algorithms derive the clustering from the network itself, while geometric clustering algorithms work on the basis of the embedded point set. The *k-means* algorithm [Llo82] is an example of the latter kind. As such, it may depict structures that are not actually in the data, but completely artificial. However, when creating a map, k-means may be used in conjunction with an embedding that was derived for example from multidimensional scaling [KW78] or any other embedding algorithm that yields good separation between clusters and puts similar vertices in the same geometric region. The clustering may also be used to softly influence the layout e.g. by reducing inter-cluster weights. In order to obtain a meaningful clustering, we also found that vertex and edge filtering methods are required. This chapter presents how we experimentally determined which clustering and filtering methods are applicable for opinion spaces and describes the methods we employ in the final application.

4.1. Clustering Experiments

We used the IGRAPH-library¹ [CN06] to explore several clustering algorithms that may produce suitable embeddings: `community_multilevel`, `community_infomap`, `community_spinglass`, `community_walktrap`, and `community_label_propagation`. The following subsections give an intuition of how each of these algorithms works and how we evaluated their compatibility with the coherence measures. We settled with `community_infomap` for the application, which is therefore explained in more detail.

4.1.1. Louvain Method

IGRAPH's method `community_multilevel` implements the *Louvain* method of community detection, which is applicable even for very large networks [BGLL08]. It is based on *modularity*, a density-based objective function for clustering, introduced by Newman and Girvan [NG04]. Some definitions are required before modularity can be defined (see also [HSWZ17]). The *neighbourhood* $N(v)$ of a node $v \in V$ is defined as

$$N(v) := \{u \in V \mid \{u, v\} \in E\}$$

¹<http://igraph.org/>

The *weighted degree* of a node $v \in V$ is then given by

$$\deg(v) := \sum_{u \in N(v)} \omega(\{u, v\})$$

The *volume* of a set of nodes C is the sum of their weighted degrees:

$$\text{vol}(C) := \sum_{v \in C} \deg(v)$$

The *internal volume* of a set of nodes C is the sum of weights of all intra-cluster edges:

$$\text{int}(C) := \sum_{e \in C} \omega(e)$$

The *cut* between two sets of nodes C and D is the sum of all inter-cluster edges which connect C and D :

$$\text{cut}(C, D) := \sum_{u \in C, v \in D} \omega(\{u, v\})$$

where $\text{cut}(C) := \text{cut}(C, V \setminus C)$ and $\text{cut}(\mathcal{C}) := \sum_{C \in \mathcal{C}} \text{cut}(C)$.

The last three concepts are illustrated in Figure 4.1.

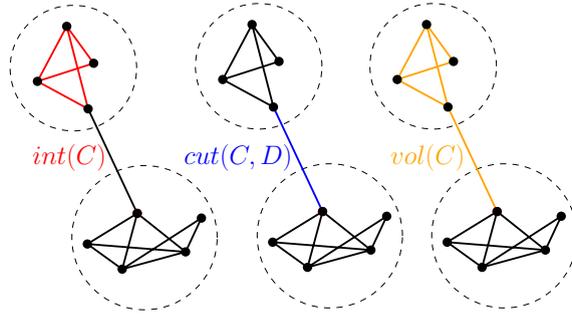


Figure 4.1.: The internal volume $\text{int}(C)$ of a cluster C is given by the sum of weights of intra-cluster edges (red). The cut $\text{cut}(C, D)$ between two clusters C and D is given by the sum of weights of inter-cluster edges connecting the respective clusters (blue). The volume $\text{vol}(C)$ of a cluster C is the sum of weights of all intra-cluster edges and edges going out of that cluster (orange).

Modularity is defined in terms of *coverage*, the fraction of intra-cluster edges divided by the total number of edges, or respectively the sum of the weights. Coverage is given by

$$\text{cov}(\mathcal{C}) := \frac{1}{\text{vol}(V)} \sum_{C \in \mathcal{C}} \text{int}(C) \in [0, 1]$$

Using the whole graph as the only cluster yields the optimum value of 1, which means that coverage is not a suitable optimization criterion for clustering. The modularity of a partition is its coverage subtracted by the expected value of the coverage for the same clustering of the same set of vertices with the same degrees but otherwise random connections between them. Modularity can now be defined as:

Definition 4.1. Modularity

$$Q(\mathcal{C}) := \text{cov}(\mathcal{C}) - \frac{1}{\text{vol}(V)^2} \sum_{C \in \mathcal{C}} \text{vol}(C)^2 \in [-1, 1]$$

The Louvain algorithm is an agglomerative bottom-up algorithm. At first, every vertex forms its own community. In each following iteration, each vertex is re-assigned to a neighbouring community in a greedy manner: For each vertex v_i , all of its neighbours v_j are considered and the respective local contribution to the overall modularity score if v_i is placed in the community of v_j is evaluated. Once a partition is determined where no single move of a vertex to another community would increase the modularity score, each cluster is contracted into a single vertex (while retaining the summed up weight of the adjacent edges) and the process is repeated on this new level. The algorithm terminates once the modularity score cannot be increased any more by contracting communities into vertices.

Louvain runs in almost linear time on sparse networks [BGLL08]. In very large networks it is possible that smaller clusters cannot be detected with this optimization criterion [FB07]. The algorithm produces nice clusters, but will always separate, even if there is really no structure. Therefore it is necessary to further investigate if the clusters are semantically meaningful.

4.1.2. Infomap

The *infomap* algorithm, devised by Rosvall and Bergstrom, optimizes a criterion called the *map equation* [RB08]. It is a quality measure for clustering that—just as modularity—formalizes the above mentioned internally-dense-externally-sparse paradigm. Conceptually, it is however quite different from modularity. The basic idea is to make use of the fact that there is a duality between finding patterns in the structure of a network and finding a description of minimal length of the movement of *flow* that is induced by the links of the network. This is illustrated in Figure 4.2.

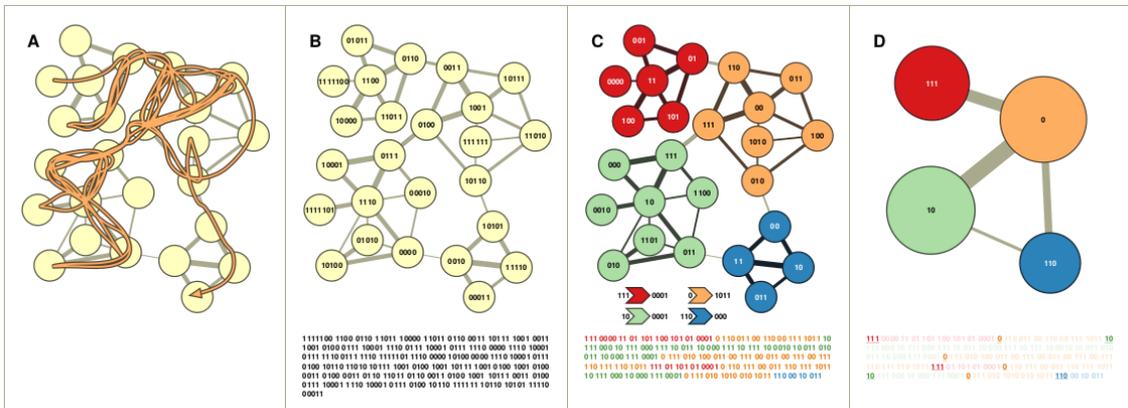


Figure 4.2.: Community detection by code compression. Source: [RB08]

Figure 4.2 (A) shows a 71-step random walk on a graph with $n = 25$ nodes. (B) gives one possible description of that walk by using a *prefix-free Huffman code*. This means, that a unique bit string is assigned to every node, and that none of these names is a prefix of any other name. To reduce code length, the length of a codeword representing a particular node is derived from its ergodic node visit rate in an infinite random walk. The ergodic node visit frequencies correspond to the eigenvector of the leading eigenvalue of the network’s transition matrix. (C) gives another possibility, a hierarchical description in which unique codewords are assigned only to major clusters while the same codeword may be assigned to nodes in different clusters or *modules*. To decipher the code, two types of codebooks are used. An *index codebook* featuring the module codewords, and a *module codebook* that contains for each module the codewords for nodes comprised by

that module as well as an exit code. The resulting code in this example requires only 243 bits. (D) shows a coarse-grained description that reports only the module names and omits the locations within the modules. However, the actual codes are not necessary for identifying an optimal partition of the network, but merely the compression rate. The map equation specifies for a given network partition a theoretical lower limit of the description length of a random walker's trajectory along the network. The random walker is supposed to model real flow. In order to find an appropriate coarse-grained description of a network, the map equation is minimized over all possible partitions.

At the heart of the map equation lies Shannon's source coding theorem [Sha01], which again, is central to information theory. Let X be a random variable with n states and associated probabilities of occurrence p_1, p_2, \dots, p_n . When using binary codewords to describe these n states, the average length of the codewords can be no less than the *entropy* of X ; otherwise, information will be lost, according to the theorem.

Information, in the context of information theory, thus is a measure of *eliminated uncertainty*. It is not to be confused with *meaning* in the everyday sense, as also a meaningless bit string may contain a high information content. The information content of a symbol with probability p is given by $I(p) = \log_2(\frac{1}{p}) = -\log_2 p$ (c.f. Figure 4.3a). Now Shannon's entropy H is a measure of the average information content per symbol, produced by a stochastic source of data. Suppose we have a set of n possible events with associated probabilities of occurrence p_1, p_2, \dots, p_n . H quantifies the amount of uncertainty about which event will occur. It is given by the expected information content:

$$H = -\sum_{i=1}^n p_i \log_2(p_i)$$

Consider the example given in Figure 4.3b: In case of two possibilities with probabilities p and $1 - p$, the entropy equals $H = -(p \log p + q \log q)$. If both probabilities are equal, the entropy is highest. If one event has lower probability, the uncertainty about its occurrence is lower.

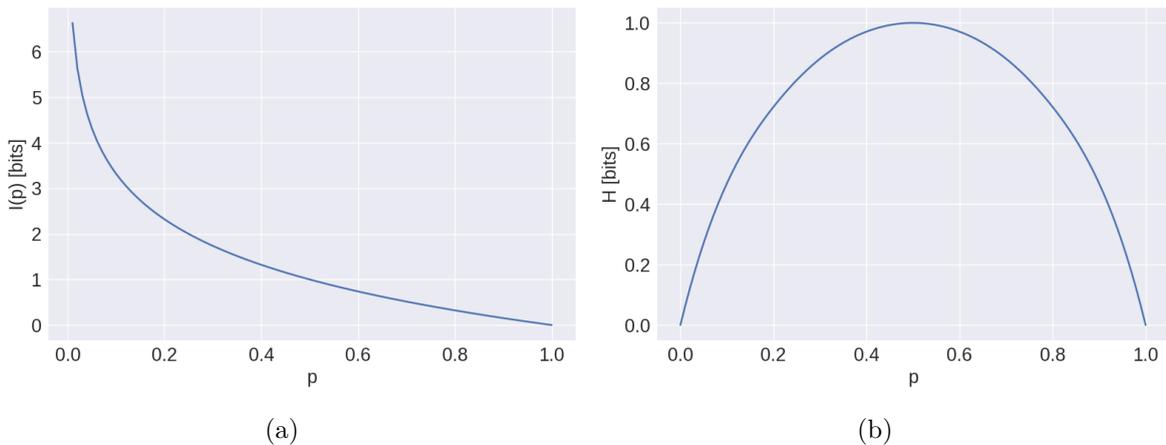


Figure 4.3.: **(a)** Information content of a symbol that occurs with probability p . The lower that probability, the higher is its information content. **(b)** The entropy for the case of two possibilities with probabilities p and $1 - p$, $H = -(p \log p + q \log q)$, as a function of p .

The map equation is given by

$$L(\mathcal{C}) = q_{\mathcal{C}} H(\mathcal{Q}) + \sum_{i=1}^m p_{i \circ} H(\mathcal{P}^i)$$

Let \mathcal{C} be a partition of a graph's n nodes into m clusters. Then $L(\mathcal{C})$ specifies a lower bound of the length of the code that describes on step of the random walker on the graph. Herein,

- q_{\curvearrowright} denotes the probability that the walker switches clusters on any given step and
- $p_{i\circlearrowleft}$ denotes the usage rate of module codebook i .

$H(\mathcal{Q})$ and $H(\mathcal{P}^i)$ are the frequency-weighted average lengths of codewords in the index codebook or module codebook i respectively.

An equivalent definition of the map equation, in terms of the concepts defined in Subsection 4.1.1 is given in [HSWZ17], where $p\log p(x) := x \log x$ for simplification:

$$L(\mathcal{C}) := p\log p\left(\frac{\text{cut}(\mathcal{C})}{\text{vol}(V)}\right) - 2 \sum_{C \in \mathcal{C}} p\log p\left(\frac{\text{cut}(C)}{\text{vol}(V)}\right) \\ + \sum_{C \in \mathcal{C}} p\log p\left(\frac{\text{cut}(C) + \text{vol}(C)}{\text{vol}(V)}\right) - \sum_{v \in V} p\log p\left(\frac{\text{deg}(v)}{\text{vol}(V)}\right)$$

The map equation can be heuristically optimized [RAB09]. As the last term remains constant for all possible partitions, it is sufficient to keep track of the changes in $\text{cut}(\mathcal{C})$ and $\text{vol}(C)$. The heuristic applied in [RAB09] is an extended version of the algorithm presented in [BGLL08], described in the last subsection. A move that was optimal in an early stage of the algorithm does not need to be optimal at a later stage, as nodes that have been assigned to the same community are restricted to move jointly when the network is rebuilt. The final state of the main algorithm is reached when the hierarchical rebuilding of the network does not result in any further reduction of the map equation. After it is reached, two extensions are introduced, which increase the accuracy of the algorithm. They are repeated in sequence after the final state of the main algorithm, until the clustering cannot be improved any further. The extensions involve breaking the communities in two different ways: First, to allow *submodule movements*, the main algorithm is re-applied to the clusters determined in the final step, treating each cluster as a network of its own. This generates sub-clusters, that are then moved back to their clusters of the previous step but each sub-cluster is now freely movable between these clusters. This extension is recursively applied until no more splits into sub-clusters are possible. Second, to allow *single-node movements*, each node is re-assigned to form its own cluster but moved back to its respective cluster in the previous step. The algorithm can then be re-applied with the individual nodes being freely movable. This extended version of the Louvain algorithm reduces the probability that the final clustering corresponds to a local optimum.

4.1.3. Miscellaneous Tested Algorithms

We investigated three other promising algorithms implemented in IGRAPH, the general ideas of which are now given for the sake of completeness.

Spinglass. The *spinglass* community detection method was proposed by Reichardt and Bornholdt [RB06] and was extended in [TB09] to support signed graphs. For negatively valued links, the internally-dense-externally sparse paradigm is reversed: There should be only few negative edges within clusters and many negative edges between clusters. The algorithm is based on a physical model which models the interaction of atomic spins on a crystalline lattice, a so-called *Potts model*. The original authors propose a physical interpretation of community detection as finding the spin configuration which minimizes the energy of an infinite range spinglass. A spinglass is a disordered magnet, which means that the magnetic spins of its atoms are not aligned in a regular lattice, as it

is the case with ferromagnets. The energy of a dynamical particle system as function of its phase space, that is, the space of all possible states of the system, is given by the *Hamiltonian* function. In [TB09], the concept of modularity is adapted to handle negative links by using this physical interpretation: Positive and negative links are first treated separately, defining a Hamiltonian for each part that represents the “energy” of a clustering. Cluster-internal positive links are rewarded and their absence is penalized. Analogously, internal negative links are penalized and their absence is rewarded. The authors then show that minimizing a combination of the two Hamiltonians corresponds to maximizing modularity. The minimum, or *ground-state*, is then found with *simulated annealing*, a heuristic for approximating the global optimum of a given function.

Walktrap. The community detection algorithm of Latapy and Pons is based on the tendency of short random walks to stay within a community [PL05]. From the information given by short random walks (of 4 steps per default) in the graph, a distance to measure similarity between vertices or groups of vertices is derived. Starting from a partition where every vertex forms its own singleton community, adjacent communities (having at least one edge between them) are merged in a bottom-up manner, based on the distance measure. The stages of the algorithm can be encoded as a tree, a so-called *dendrogram*. The method then uses modularity for evaluating the quality of a partition and thereby selecting where to cut the dendrogram (c.f. Figure 4.4).

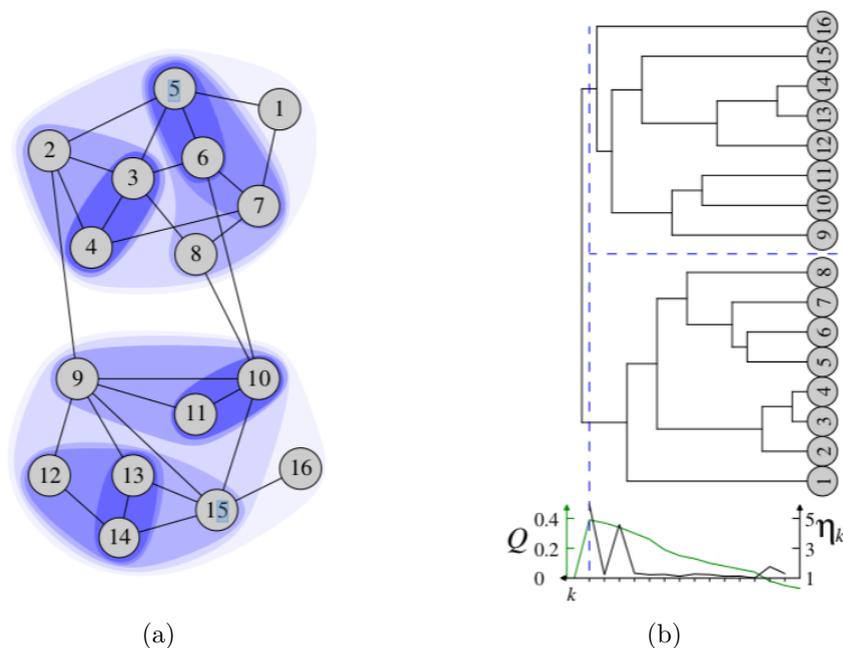


Figure 4.4.: **(a)** Community structure found by the algorithm using 3-step random walks. **(b)** The iterations encoded as a dendrogram. Modularity Q is maximized for two communities. Image source: [PL05]

Label Propagation. The near linear time label propagation method is a randomized algorithm introduced by Raghavan et al. [RAK07]. By contrast to the other methods, this method does not optimize any particular measure of community strength, but is guided by the network structure alone.

Initially, a unique label is assigned to every node. At every following iteration, nodes are arranged in a random order, and each node in that order adopts the label of the maximum of its neighbours, with ties broken uniformly at random. This means, that if there is more than one label with the maximum sum of edge weights, the algorithm picks one of them at random, with each of them having the same probability to be picked.

In this way, labels propagate through the network and densely connected subgraphs reach a consensus on their labels. The algorithm continues until the stopping criterion is fulfilled, which demands for every node to have at least as many neighbours within its community as within any of the other communities. There can be various community structures which fulfil that criterion, e.g. the trivial one with only one cluster.

4.1.4. Evaluating the Algorithms

In order to get an idea which clustering algorithm is able to identify semantically meaningful structure in the data we produced for each algorithm and each dataset histogram visualizations as shown in Figure 4.5. The horizontal axis displays each of the debate’s statements and the vertical axis shows the clusters (including the number of vertices featured in a cluster) as found by a specific algorithm. Note that statements 1 to 8 are core claims. In case of Figure 4.5 the clusters are separated rather well, especially with respect to the core claims, which indicates a semantically meaningful clustering. This clustering was produced in exactly this way for the mutual coherence graph and the Shogenji coherence graph by infomap, multilevel, and walktrap. In case of the former, label propagation produces exactly that result as well, whereas a slightly different result is produced in case of the latter. C_0 comprises 80 positions, all of which agree to [Meat-OK] (statement 1) but many of which also agree to [Less-meat] and [Less-animal] (statements 7 and 8). Other prominent statements in this cluster are

- 10. An acceptable diet must also meet culinary standards.
- 16. Everything that is natural is good and legitimate, the unnatural is bad and illegitimate.

On the other hand, abandoning meat and animal products altogether (statements 5 and 6, [strict-veggie] and [strict-vegan]) is rejected by 100%. Also, statements 46 - 48 are rejected by 100%. Positions in C_1 all agree to [strict-veggie] (statement 5). Positions in C_2 all agree to [strict-vegan] (statement 6) and consequently also to [strict-veggie]. C_3 is made up of omnivores, 100% of positions here agree to [Meat-OK] and [Eat-what-you-want].

Tables 4.1, 4.2, and 4.3 show exemplarily the results of the five clustering algorithms in combination with the six different distance measures for three of the simulated datasets. The first dataset was generated completely randomly, in the second one explicit stance via some core claims was enforced, and the last one has a high degree of artificial structure (please refer to Subsection 3.3.1 for details).

In case of the opinion-graphs weighted by mutual coherence, we had to perform some initial weight modification for all but the spinglass algorithm, which is the only algorithm that handles negative edge weights. This modification was to add a global constant $c = 0.5$ to every edge weight and removing all edges that remained negative afterwards.

- 2017-04-27T15-31-02 is a dataset comprising 265 relatively large positions: on average, explicit stance is taken on 10.43 statements. Otherwise, the positions were generated randomly, with no explicit stance on the core claims being enforced.
 - Per default, we tested on the complete weighted graph (featuring 34980 edges).
 - In case of mutual coherence, 12614 edges remain after the modification stated above. In case of the spinglass algorithm in combination with mutual coherence, again the complete graph was used.
 - The Shogenji measure does not yield complete graphs. For this dataset, the Shogenji graph features 7187 edges.

The number of clusters produced by each combination is given in Table 4.1.

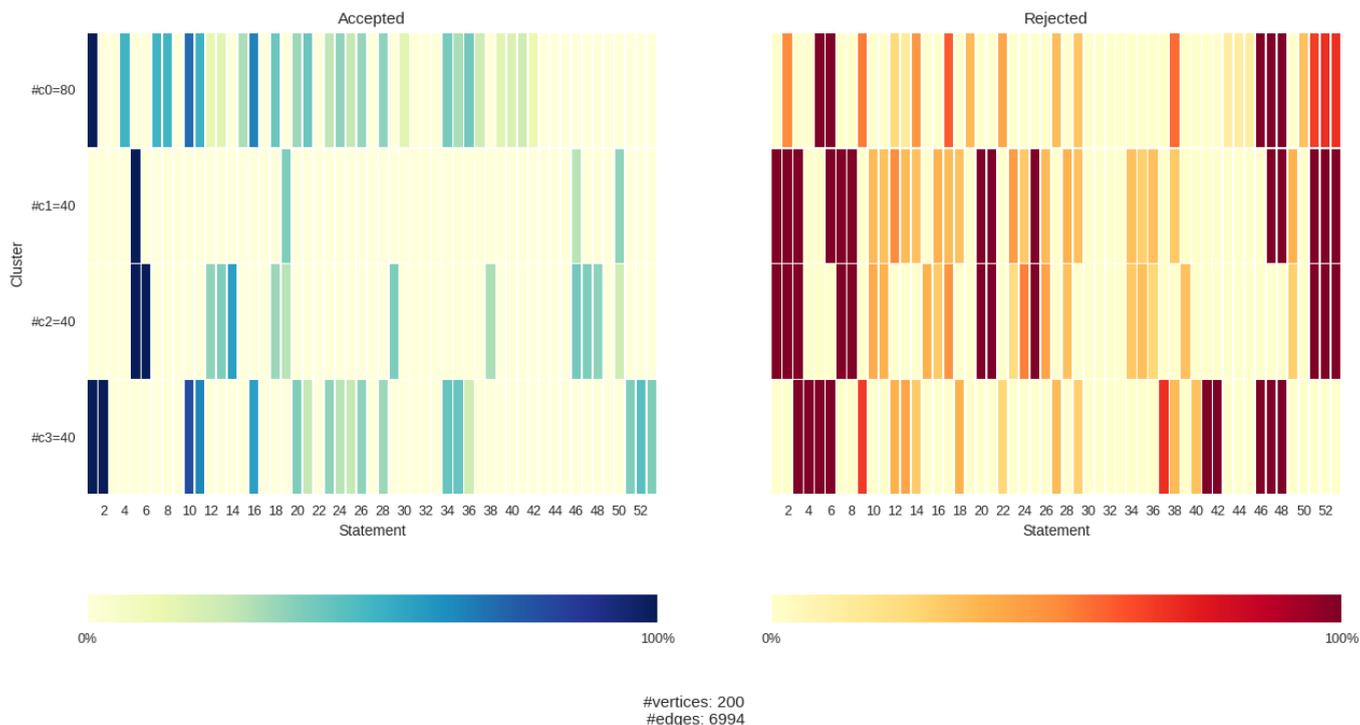


Figure 4.5.: Histograms of per cluster acceptance (cold shades) and rejection (warm shades) rates to each of the debate’s statements in dataset 2017-05-13T05-38-29. This clustering is produced by infomap, multilevel, label propagation, and walktrap for the mutual coherence graph filtered with $c = 0.5$. In case of the Shogenji coherence graph, this same clustering is yielded by all algorithms but label propagation.

- 2017-05-02T22-56-28 comprises 137 relatively small positions, with explicit stance taken on 6.39 statements on average, and explicit stance on core claims 1 and 2 ([Meat-OK], [Eat-what-you-want]) being enforced.
 - Here, the complete weighted graph comprises 9316 edges.
 - The filtered mutual coherence graph has 3095 edges.
 - The Shogenji graph features 2428 edges and is decomposed into three disconnected components, meaning that the spinglass algorithm, which does not work for unconnected graphs, cannot be applied.

The number of clusters produced by each combination is given in Table 4.2.

- 2017-05-13T05-38-29 is made up of 200 positions with average length of 11.63 statements.
 - Here, the complete weighted graph comprises 19900 edges.
 - The filtered mutual coherence graph has 6994 edges and two disconnected components.
 - The Shogenji graph features 5775 edges and three disconnected components.

The number of clusters produced by each combination is given in Table 4.3.

Our decision of combining mutual coherence with infomap in the final application relies on the following results: The cheap measures, $nEditSim$, $gHamSim$, $cHamSim$, and $nClos$ do not yield sufficiently structured graphs for representing the opinion space, as even for the pre-structured dataset, trivial clusterings (that is, all nodes are assigned to one

	nEditSim	gHamSim	cHamSim	nClos	mutCoh	shoCoh
Infomap	1	1	1	1	1	13
Multilevel	3	3	2	2	3	8
Label Propagation	1	1	1	1	1	6
Spinglass	3	3	3	2	4	10
Walktrap	4	3	2	2	2	10

Table 4.1.: Number of clusters for different combinations of algorithms and coherence measures in dataset 2017-04-27T15-31-02, which comprises relatively large positions (mean length: 10.43) and does not enforce explicit stance on the core claims.

	nEditSim	gHamSim	cHamSim	nClos	mutCoh	shoCoh
Infomap	1	1	1	1	1	4
Multilevel	3	2	3	2	3	4
Label Propagation	2	1	1	1	2	3
Spinglass	3	2	3	2	3	–
Walktrap	3	2	3	2	2	3

Table 4.2.: Number of clusters for different combinations of algorithms and coherence measures in dataset 2017-05-02T22-56-28, which comprises relatively small positions (mean length: 6.39), with explicit stance on core claims 1 and 2 ([Meat-OK], [Eat-what-you-want]) enforced.

cluster) are produced in many cases. Mutual coherence and Shogenji coherence are therefore the more promising candidates. The Shogenji graphs show a tendency to have many clusters.

In case of the pre-structured dataset, the clusterings show the highest degree of consensus in case of the mutual coherence and Shogenji coherence graphs. For quantifying the similarity between two clusterings, we use the *Jaccard index*, a general measure for comparing two sets. The Jaccard index J of two sets (clusters in this case) A, B is defined as the size of the intersection divided by the size of the union of the sets:

$$J(A, B) := \frac{|A \cap B|}{|A \cup B|}$$

We short-listed infomap and multilevel because they are established algorithms that are frequently in use [FH16], and the other algorithms seemed to offer no advantage. To the contrary—not being able to use disconnected graphs, as it is the case for spinglass, is quite a disadvantage in our case as there may be survey inputs that destroy connectivity. Figure 4.6 shows Jaccard matrices plotted as heatmaps, which compare the clusterings found by infomap and multilevel for each of the three datasets mentioned above, each for mutual coherence (left) and Shogenji coherence (right). As can be seen in Figure 4.6a, infomap produces a trivial clustering for the random dataset weighted by mutual coherence. In case of the other two datasets, each combination yields perfect (or in the case shown in Figure 4.6d almost perfect) consensus. Interestingly, the algorithms do not identify the five predetermined clusters in the structured dataset but identify only four clusters. Multilevel will always separate, even if there is really no structure in the graph. However, OPMAP requires a clustering algorithm that correctly detects that random data is random. This excludes multilevel and *shoCoh*. Therefore, we finally settled on infomap in concert with the degree of mutual coherence.

	nEditSim	gHamSim	cHamSim	nClos	mutCoh	shoCoh
Infomap	1	1	1	1	4	4
Multilevel	2	2	2	2	4	4
Label Propagation	2	1	1	1	4	6
Spinglass	4	2	4	2	3	–
Walktrap	2	2	2	2	4	4

Table 4.3.: Number of clusters for different combinations of algorithms and coherence measures in dataset 2017-05-13T05-38-29, an artificially pre-structured sample comprising larger positions (mean length: 11.63).

Not only for reasons of performance, but also to obtain a better clustering, it is desirable to filter out many edges of the complete graph. We additionally introduce a vertex filtering. These two methods are described in the following sections.

4.2. Vertex Filtering

Nodes that are well connected to nodes of almost any cluster are called *hub nodes*. In our case, they correspond to very unspecific positions that have many connections as they are consistent with many other nodes. We had to find a way to deal with hubs, especially because they make layouts difficult. An initial idea was to filter them and put them in the middle of the layout (as “capital”) at first and in the end let the force-directed algorithm move only the hubs. Later in the design process we decided to use a hybrid clustering algorithm that derives the clustering not only from the structure of the graph, but takes the structure of the underlying opinion data into account as well, in a manner that makes sense from an argumentation theoretic point of view.

In particular, we construct another graph from our data which organizes the opinions according to *logical entailment*. This graph is *not* the same as the opinion-graph. This new graph, in contrast to the opinion-graph, is a directed graph. Two opinion vertices o_i and o_j are connected by an edge $o_i \rightarrow o_j$, if $o_j \Rightarrow o_i$. As stated in Section 3.2 we calculate the deductive closure for all gathered opinions. Therefore, the logical entailment relation “ \Rightarrow ” corresponds to the subset relation “ \subseteq ”. That is, if $o_i \subseteq o_j$, then $o_j \Rightarrow o_i$: If the statements of opinion o_i are a subset of the statements of opinion o_j , then the latter is deductively entailed by the former. Such an entailment-graph is an instance of a DAG. An illustrative example is given in Figure 4.7. The entailment-graph contains four types of nodes: *Sources*, with only outgoing edges, *sinks* with only incoming edges, *internal nodes* with incoming and outgoing edges and *isolated nodes* with no adjacent edges. The sink nodes tend to be the most specific positions, and the source nodes the most general ones. The opinions that are actually drawn on the map are only the isolated nodes and the sink nodes of the entailment-DAG.

With the online survey, we collected 210 real-world opinions to draw the initial map, two of which were not considered as they were logically inconsistent. 159 of them are sinks, two are isolated. The other two node types are considered in the map in the form of vertex weights. By default, every opinion node has a weight of 1.0. If it is a non-sink node, its weight is distributed evenly among its *children*. Also, if the same opinion is stated multiple times, a weight of 1.0 is added to the corresponding vertex each time (or respectively distributed among the children, if it is a non-sink opinion).

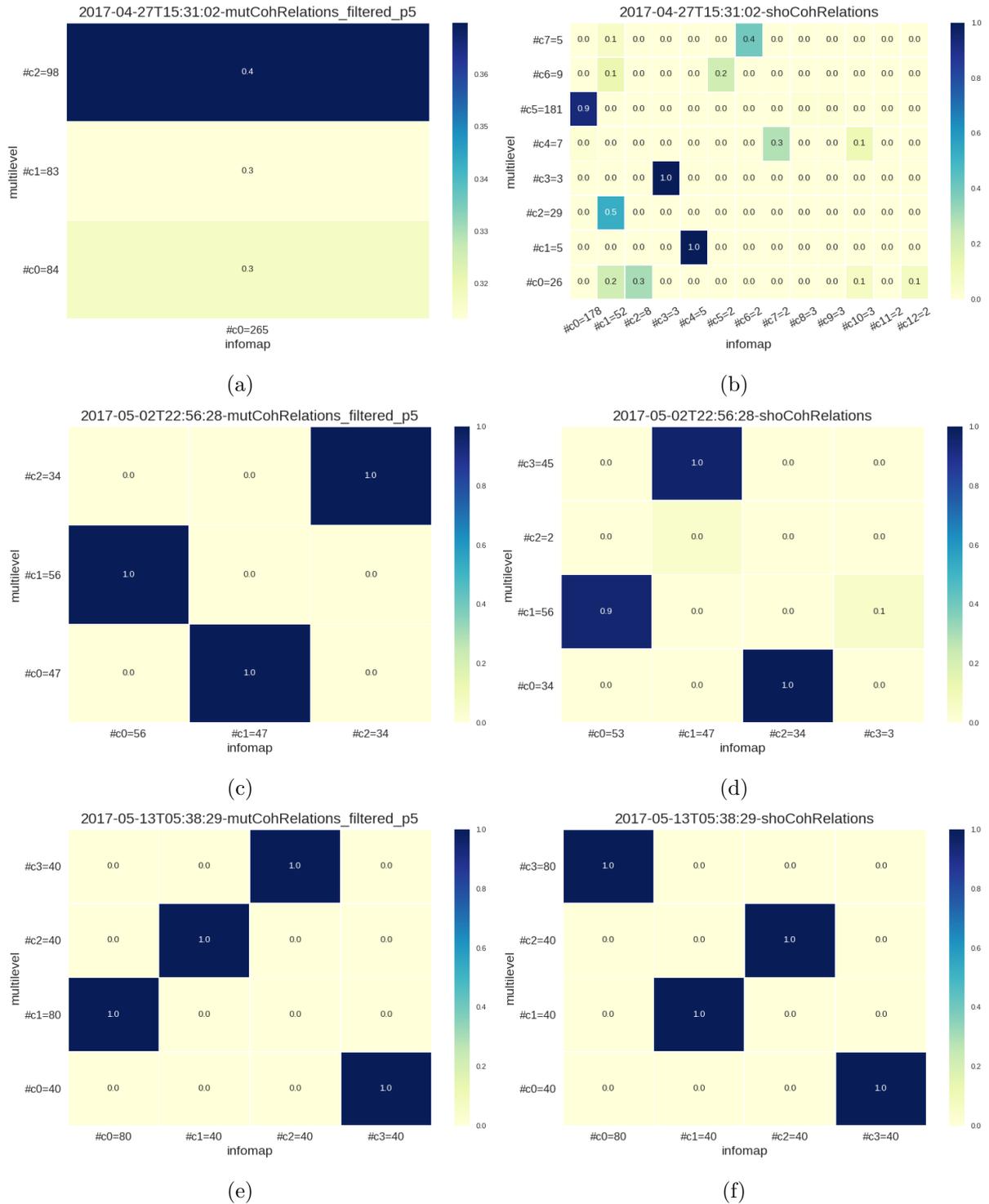


Figure 4.6.: Jaccard similarities of multilevel and infomap clusterings in three datasets of varying degree of structure. In case of the random dataset, no high consensus was reached, neither for mutual coherence (a), nor for Shogenji coherence (b). The slightly structured dataset was clustered by the algorithms with (almost) perfect consensus (c,d) and the structured dataset with perfect consensus (e,f).

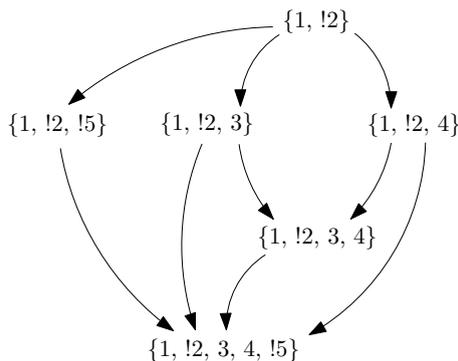


Figure 4.7.: Illustrative entailment-graph.

4.3. Edge Filtering

As established in the last section, the opinion-graph on which the initial map is based, does not include all of the opinions. Instead, we use the subgraph induced by the vertices that are sinks or isolated vertices in the entailment-graph, which are $n = 161$ opinions of the 210 collected opinions. As mutual coherence values are calculated for all pairs of opinions, this subgraph still comprises $\frac{n \cdot (n-1)}{2} = 12880$ edges. But not only for reasons of performance, sparsification is necessary. Recall that the edge values given by the degree of mutual coherence are real numbers in the range $[-1, 1]$. As established above, most clustering and layouting algorithms accept only positive edge weights. Furthermore, the internally-dense-externally-sparse paradigm can hardly be optimized in a complete graph since it is supposed to find a partition where the intra-cluster density is higher than the inter-cluster density.

Therefore we first apply *global edge filtering* by adding a constant of $c = 0.808$ to every edge weight and discarding any edge with a weight that remains negative afterwards. A constant > 1.0 turns every edge weight positive, so the closer to 0.0 the constant, the more edges are filtered out. The constant of 0.808 is used because with this value, just enough edges are retained for the graph to not decompose into connected components.

Furthermore, we apply *local edge filtering* as proposed in [SPR11]. For every vertex v the $\text{deg}(v)^e$ edges with the highest edge weights are marked. Here, $\text{deg}(v)$ is the *unweighted vertex degree*, that is, the number of edges adjacent to v . e is a constant $\in [0.0, 1.0]$. This ensures, that for each node, at least one edge is marked. If $e = 1.0$, all edges are marked. Any edge that has been marked by at least one vertex is retained while unmarked edges are discarded.

4.4. Semantic Analysis

We use the free graph visualization software Gephi² [BHJ09] for sanity checks. Figure 4.8 shows a visualization rendered in Gephi using the inbuilt layouting algorithm Force Atlas 2 [JVHB14]. It depicts the vertex- and edge-filtered opinion-graph. The colours are assigned according to an infomap clustering. Vertex radii are proportional to vertex weights. The labels indicate which survey option was most frequently selected in every cluster. Vegans and vegetarians each form their own clusters. The majority of participants selected “restricted meat consumption”. These, as well as the “omnivore”-opinions are subdivided into two clusters by the algorithm.

We performed a semantic analysis of the clusters by hand, revealing an interesting pattern in the opinion landscape. It is made up of three main divisions:

²<https://gephi.org/>

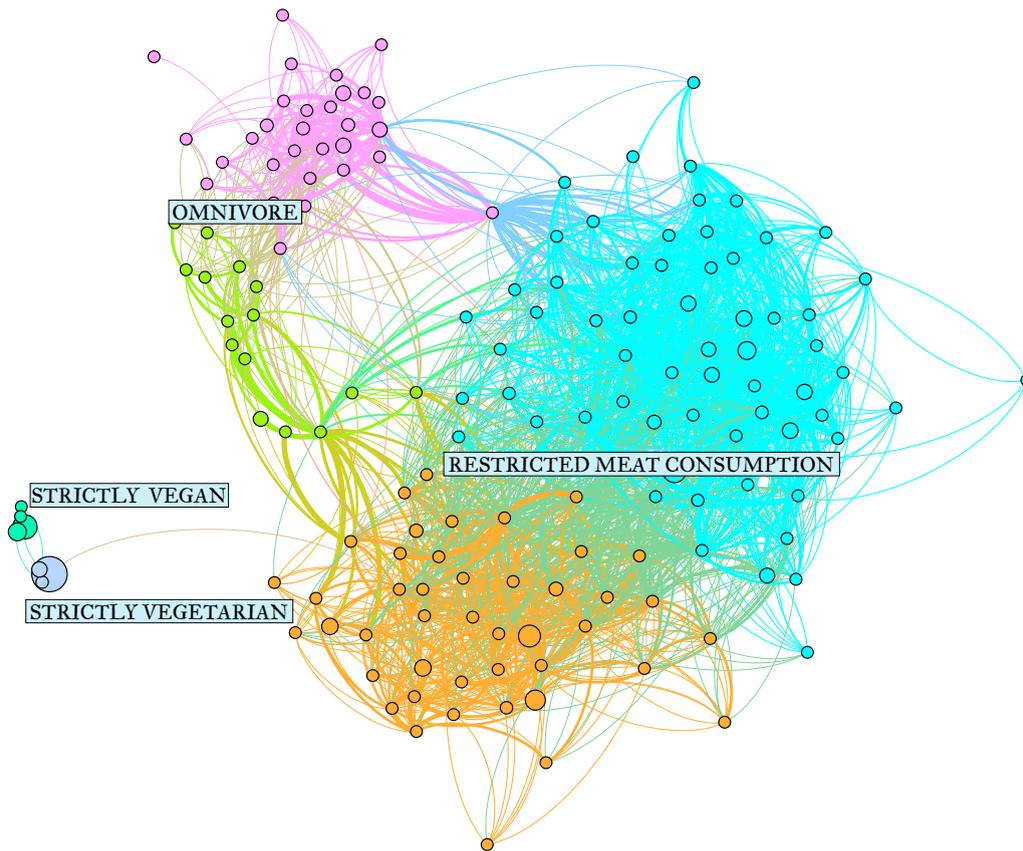


Figure 4.8.: Graph drawing of sink opinions created in Gephi with Force Atlas 2. Vertices are coloured according to the infomap clustering and vertex radii correspond to vertex weights. Labels indicate the initial survey option that was mostly chosen in each cluster.

- *No-meat*, subdivided into vegan and vegetarian.
- *Less-meat*, subdivided into people who do not really have a reason to eat meat but do it anyway, perhaps just out of habit, and those who do not abandon meat entirely because of the pleasure experience.
- *Eat-whatever-you-want*, subdivided into people who eat meat for health reasons, and again people who eat meat because of the pleasure experience.

This can also be seen in Figure 4.8. In order to perform this qualitative analysis, we investigated which statements were most frequently rejected and which statements were most frequently accepted per cluster. Figure 4.9 summarizes these results as histogram. Appendix Section B includes bar charts which indicate the complete acceptance and rejection rates as well as tables listing the top five most frequently accepted and rejected statements. For the exhibition, we formulated somewhat feuilletonist qualitative country names and descriptions based on the most frequently accepted and rejected statements, which can also be found in Appendix Section B.

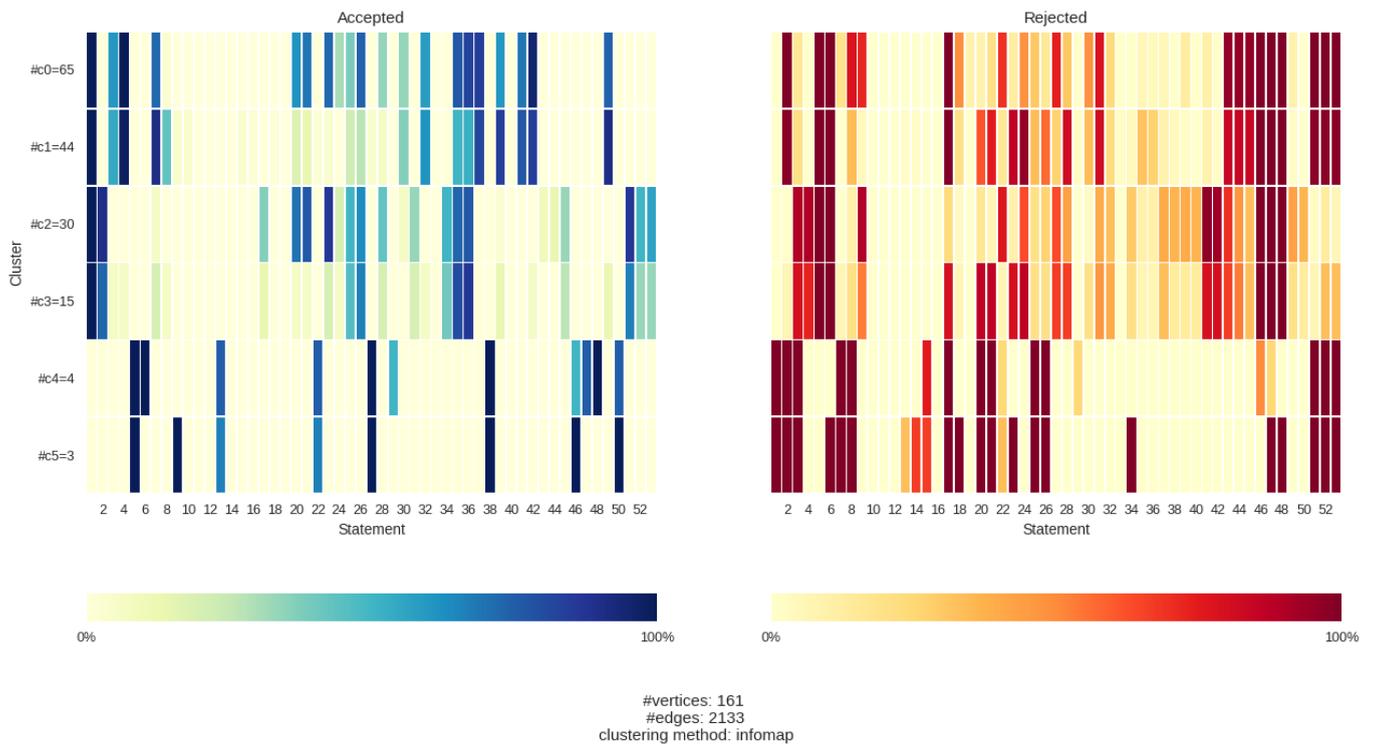


Figure 4.9.: Histogram of per cluster acceptance (cold shades) and rejection (warm shades) rates to each of the debate’s statements in the survey data. Here, the 159 sink opinions and 2 isolated opinions in their deductively closed form are considered.

5. Graph & Map Drawing

When an abstract graph is represented as an adjacency list or a matrix, it clearly contains just the same information as a node-link visualization. Yet, displaying hundreds of opinions and their similarities in form of an adjacency list would be not very helpful for a human user.

In a clear drawing, the user can immediately find the information he or she is looking for, while a poor visualization can be confusing and requires a lot of time to extract the relevant information from. However, there is no unique measure of quality and it depends on the individual application which aspects of the graph structure are emphasized. Therefore, if the overall community structure is the relevant piece of information, a visualization that makes the clusters visible but is possibly less exact is preferable. In the case of OPMAP, both, a rough overview of the structure, as well as the detailed relationships of the graph's elements are offered to the viewer within the same visualization (c.f. Figure 5.1).

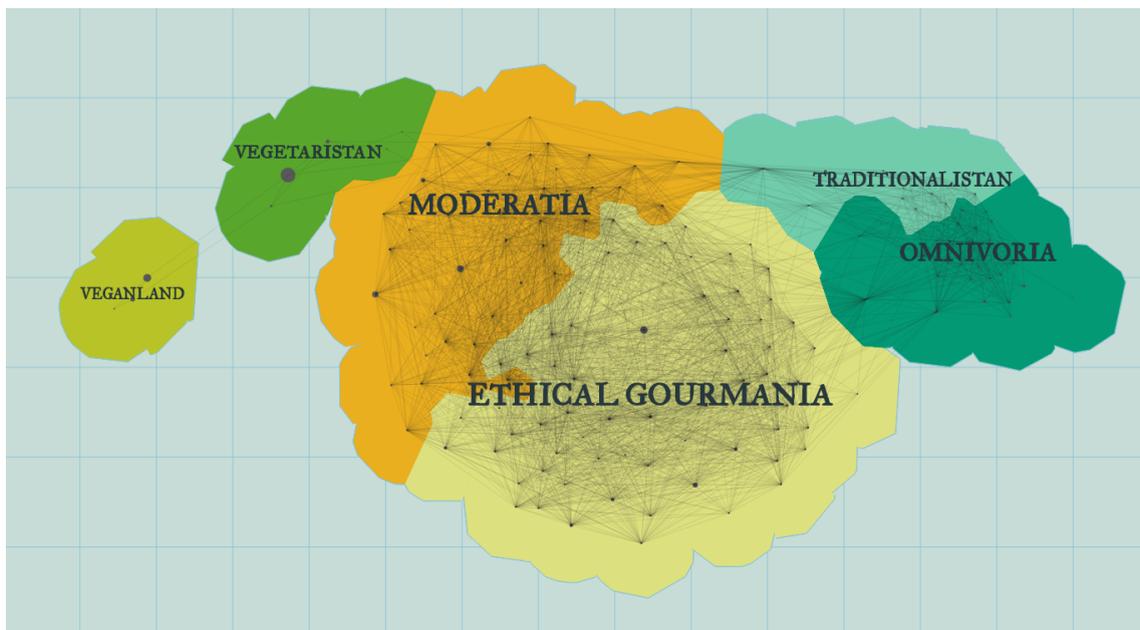


Figure 5.1.: A map of the “World of Diets”.

Below we list a drawing convention and several aesthetic criteria we want to optimize:

- Each node lies in the country specified by the given clustering. Similar nodes—with similarity specified by the weights on the edges connecting them—are positioned close to each other, whereas dissimilar nodes lie relatively far away from each other.
- The sizes of countries and labels correspond to the number of represented inhabitants or opinions.
- Adjacent countries represent semantically closer eating behaviours than non-adjacent ones.
- The lengths of the country borders are roughly proportional to the extent of the semantic closeness of neighbouring countries.
- The countries should not be too fragmented, although some enclaves are in line with the map metaphor and may in some cases represent the opinion space more accurately.

Figure 5.1 shows that our implementation optimizes these criteria fairly well. In this map, a total of 217 opinions are represented, where 86 form *Ethical Gourmania*, 59 form *Moderatia*, 34 form *Omnivoria*, 15 form *Traditionalistan*, 14 form *Vegetaristan*, and 9 form *Veganland*. Country memberships are indicated by colours. Country adjacencies and border lengths are semantically meaningful: The two biggest countries, *Ethical Gourmania* and *Moderatia* both contain opinions which advocate restricted meat consumption and accordingly share a long border. *Omnivoria* and *Traditionalistan* are made up of opinions in favour of an omnivorous diet. Therefore, they also share a long border. The opinions in *Omnivoria* and *Ethical Gourmania* have in common that the pleasure experience of meat consumption is held in high regard, thus *Omnivoria* also shares a relatively long border with *Ethical Gourmania*. *Traditionalistan* and *Ethical Gourmania*, in turn, share the common belief that killing and eating animals is something very natural. The shared border of *Vegetaristan* and *Moderatia* is accounted for by the shared profound awareness of the diverse objections to meat consumption. Veganism, in this map, is an insular practice. Indeed, the vegan way of living comes with rather extensive dietary restrictions.

Two algorithms from the fields of graph drawing and computational geometry are central to generating opinion maps. This chapter outlines these algorithms and provides some background.

5.1. Force Simulation

Force-directed graph drawing methods date back to the sixties and enjoy great popularity due to their intuitiveness and generality. They produce fairly clear and readable results, optimizing a variety of drawing properties that partially contradict each other:

- Adjacent nodes are close whereas non-adjacent nodes are further apart.
- Nodes and edge lengths are distributed evenly.
- Densely connected parts form communities.
- The number of crossings is minimized.

Being physics simulations, these algorithms (in their native form) do not take into account domain-specific knowledge—neither semantic information about the underlying dataset (if there is one) nor graph theoretical properties, such as planarity—but calculate the layout using merely the structural properties of the network.

Force-directed algorithms are a good choice for dynamic graphs, as the user can watch how the graph untangles into an aesthetically pleasing configuration with roughly uniform edge length and clear community structure. Obviously, not both of these criteria can be optimized to an ideal state at the same time, but some good compromise has to be reached. An excellent overview, from classical to cutting-edge force layout algorithms is provided in [Kob12].

As data visualization library, we use D3.js¹, a popular JavaScript library, whose module D3-FORCE² provides extensive physical tools to display clustered data with complex relations in a simple and natural way. We now describe the layout algorithm as provided by D3. It implements a numerical method of integrating Newton's equation of motion to calculate the movements of a system of particles through time, *Verlet integration*, originally proposed by Loup Verlet [Ver67], or more precisely *velocity Verlet integration* [SABW82].

In the context of physics simulations, the term *tick* denotes the passage of some amount of time. The algorithm effectively performs an *n-body* simulation, meaning it calculates the positions of a dynamical system of n particles under the influence of physical forces. *Velocity* is given by the change of position over time. In the simulation we consider the *average* velocity over discrete time intervals:

$$\mathbf{v} = \frac{\Delta \mathbf{p}}{\Delta t} = \Delta \mathbf{p},$$

as Δt is always 1 tick. Here, $\mathbf{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$ is the xy -velocity and $\mathbf{p} = \begin{pmatrix} p_x \\ p_y \end{pmatrix}$ is the xy -position. The basic velocity Verlet algorithm calculates the next position $\mathbf{p}_{t+1}(p)$ of each particle p using the current and previous positions [Pet15]:

Algorithm 5.1: Basic Velocity Verlet

```

1 Let  $\mathbf{p}_{t-1}(p)$  be the position of particle  $p$  at tick  $t - 1$ .
2 function UPDATEPOSITIONS()
3   for particles  $p \in 1 \dots n$  do
4      $\mathbf{v}_t(p) \leftarrow \mathbf{p}_t(p) - \mathbf{p}_{t-1}(p)$ 
5      $\mathbf{p}_{t+1}(p) \leftarrow \mathbf{p}_t(p) + \mathbf{v}_t(p)$ 

```

Physical forces can be mimicked as needed in the specific context, for example to simulate a bouncy ball [Pet15]:

Algorithm 5.2: Simulate Bouncy Ball with Velocity Verlet

```

1 function UPDATEPOSITIONS()
2    $gravity \leftarrow 0.5$ 
3    $friction \leftarrow 1 - \varepsilon$ , where  $\varepsilon > 0$ 
4   for particles  $p \in 1 \dots n$  do
5      $\mathbf{v}_t(p) \leftarrow (\mathbf{p}_t(p) - \mathbf{p}_{t-1}(p)) \cdot friction$ 
6      $\mathbf{p}_{t+1}(p) \leftarrow \mathbf{p}_t(p) + \mathbf{v}_t(p)$ 
7      $p_{y,t+1}(p) \leftarrow p_{y,t} + gravity$ 

```

¹<https://d3js.org/>

²<https://github.com/d3/d3-force>

In D3-FORCE, Verlet integration is used to implement a force-directed graph layout. Recall the main rationale of spring algorithms: “To embed a graph we replace the vertices by steel rings and replace each edge with a spring to form a mechanical system. The vertices are placed in some initial layout and let go so that the spring forces on the rings move the system to a minimal energy state” [Ead84].

In this context, the forces are functions which modify each node’s velocity and as a result, their positions. Some forces may also directly influence the position. They either mimic physical forces, such as electrical charge or gravity, or they resolve geometric constraints, such as centering the layout in the center of the viewport. In the simulation the forces accept an input parameter α which is responsible for cooling the simulation down, such that it eventually converges at a good layout. Cooling will be detailed later in this section.

A variety of forces serving different visualization goals has been introduced in the literature. D3 provides the following customizable forces (see the documentation of D3-FORCE [Bos17b]):

- f_{center} is a uniform translation of all the nodes’ positions such that their mean position is at some specified xy -position. It serves to center the drawing around a particular point, in our case the center of the viewport. Unlike $f_{gravity}$, it does not distort their relative positions.
- f_{link} is the force that acts on two nodes adjacent through an edge (link). Links create forces which push nodes together or apart. For each link e , two parameters are computed: **distance** and **strength**.

Link **distance** is the target distance between pairs of connected nodes. If not specified otherwise, it defaults to 30. We set it to

$$distance(e) = \begin{cases} \frac{1}{w(e)} \cdot 0.2 & \text{if } e \text{ is an intracluster edge} \\ \frac{1}{w(e)} \cdot 3 & \text{if } e \text{ is an intercluster edge} \end{cases}$$

where $w(e)$ is the edge weight. We take the inverse of the edge weight because the higher the weight of an edge between two nodes, the closer they should be together in the drawing, as a higher weight indicates greater similarity. To separate the clusters or “countries” further, we use a factor of 0.2 for edges that connect two vertices which have the same cluster membership or “nationality” and a factor of 3 for edges that connect vertices with different memberships. These numbers have been determined experimentally.

Link **strength** is a parameter that defines for each link, how rigidly the target link distance is enforced. We leave it at the default value, which is

$$strength(e) = \frac{1}{\min(deg(u), deg(v))}$$

where u and v are the vertices connected by the edge, so just the inverse of the smaller vertex degree. Algorithm 5.3 states how f_{link} as a function of α influences velocities. Again, the vectors $\mathbf{p}(v) = \begin{pmatrix} p_x(v) \\ p_y(v) \end{pmatrix}$ and $\mathbf{v}(v) = \begin{pmatrix} v_x(v) \\ v_y(v) \end{pmatrix}$ denote the xy -position or respectively the velocity of vertex v .

- f_{charge} mimics electrostatic effects which makes for the natural, organic feeling of the graph as the nodes interact with each other. Thereby, it prevents nodes from coming too close to each other. A negative value results in repulsion, while a positive value results in node attraction. For each node, a parameter **strength** is computed. It defaults to -30.

Algorithm 5.3: Link force

```

1 function JIGGLE()
2    $x \leftarrow$  draw randomly from  $\mathcal{U}\{-0.5, 0.5\}$ 
3   return  $x \cdot 10^{-6}$ 
4 function FORCELINK( $\alpha$ )
5   forall the links  $e = (u, v) \in E$  do
6      $\begin{pmatrix} x \\ y \end{pmatrix} \leftarrow \mathbf{p}(v) + \mathbf{v}(v) - \mathbf{p}(u) - \mathbf{v}(u)$ 
7     if  $x = 0$  then
8        $x \leftarrow$  JIGGLE()
9     if  $y = 0$  then
10       $y \leftarrow$  JIGGLE()
11       $l \leftarrow \sqrt{x^2 + y^2}$ 
12       $l \leftarrow \frac{l - \text{distance}(e)}{l \cdot \alpha \cdot \text{strength}(e)}$ 
13       $\text{bias} \leftarrow \frac{\text{deg}(u)}{\text{deg}(u) + \text{deg}(v)}$ 
14       $\mathbf{v}(v) \leftarrow \mathbf{v}(v) - \begin{pmatrix} x \\ y \end{pmatrix} \cdot \text{bias}$ 
15       $\mathbf{v}(u) \leftarrow \mathbf{v}(u) + x \cdot (1 - \text{bias})$ 

```

We set it to

$$\text{strength}(v) = -10 \cdot w(v)^2 \cdot \text{density}_C$$

where $w(v)$ is the vertex weight and density_C is given by the number of edges within the node's cluster divided by the number of vertices within the cluster. Recall that vertices with higher weight represent a greater number of opinions. Therefore they should take up a larger area on the map. This can be reached by giving them a greater negative charge, such that surrounding nodes are pushed apart. Also, very dense clusters with many nodes should not be pulled together too tightly, as countries with more nodes should occupy a greater area. To that end, the charge is weighted by the cluster density. f_{charge} as a function of α is computed as stated in Algorithm 5.4.

Algorithm 5.4: Charge force

```

1 function FORCECHARGE( $\alpha$ )
2   forall the vertices  $v \in V$  do
3     forall the vertices  $u \in V, u \neq v$  do
4        $\begin{pmatrix} x \\ y \end{pmatrix} \leftarrow \mathbf{p}(u) - \mathbf{p}(v);$ 
5        $l \leftarrow x^2 + y^2;$ 
6        $\mathbf{v}(v) \leftarrow \mathbf{v}(v) + \text{strength}(u) \cdot \frac{\alpha}{l};$ 

```

Unlike the link force, which affects only pairs of linked nodes, this force is global, as every node affects every other node, even if they are part of disconnected subgraphs. This leads to a complexity of $\mathcal{O}(n^2)$ for the direct-sum algorithm. D3 implements the Barnes-Hut approximation reducing the order to $\mathcal{O}(n \log n)$ [BH86]. Herein, the simulation area is recursively subdivided into rectangular regions by means of a quadtree, such that only particles (or nodes) in nearby cells have to be considered individually, whereas more distant ones can be considered a single particle centered at their center of mass. Thereby, only a fraction of the pair interactions has to be computed.

- $f_{collision}$, intuitively speaking, assigns a radius to the above mentioned “steel rings” to prevent the nodes from overlapping. For every node, nodes in its vicinity that will overlap it in the next tick (using predicted positions $\mathbf{p}(v) + \mathbf{v}(v)$) are determined and its velocity is set to push it out of the radius of each overlapping node. For every node, the parameters `radius` and `strength` can be specified. As experiments did not yield promising results, we left them at the default values, constants 1.0 and 0.7.
- $f_{gravityX}$ and $f_{gravityY}$ can be seen as horizontal and vertical “gravitational” forces. For each of the forces, one can specify a coordinate of the center of gravity, as well as a `strength`-parameter. We leave the center of gravity at the default position `{x:0, y:0}`, as the drawing is geometrically translated to the viewport center with the center force. `strength` defaults to 0.1 for both forces. If the two forces are given the same strength, the map becomes roughly circular. In order to obtain a more elongated map, we leave the horizontal force at 0.1 and assign a higher strength of 0.2 to the vertical force.

Section 6.4 describes how we use D3-FORCE to configure force layouts and gives relevant code snippets from our implementation.

A velocity Verlet numerical integrator for simulating physical forces on particles as implemented in D3 is given in Algorithm 5.5 [Bos17b].

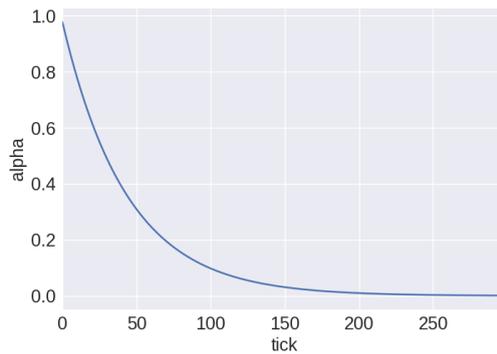
As long as the termination criterion is not fulfilled, the tick-function is executed on every tick. Here, the α -parameters $\in [0, 1]$ are responsible for cooling the simulation down:

- Once α falls below α_{min} , the simulation’s internal timer is halted.
- Line 14 increments α by $(\alpha_{target} - \alpha) \times \alpha_{decay}$. Therefore, when the simulation is started, it ticks $\lceil \log(\alpha_{min}) / \log(1 - \alpha_{decay}) \rceil = 300$ times (c.f. Figure 5.2).
- The α_{decay} rate is used to modify how quickly α approaches the desired α_{target} , that is, how quickly the simulation cools. Thus, if either one sets $\alpha_{target} > \alpha_{min}$ or $\alpha_{decay} = 0$, the simulation keeps running forever. On the other hand, a higher α_{decay} leads to a faster stabilization of the simulation, but concomitant is a higher risk of getting stuck in a local minimum. At a lower decay rate the simulation may converge on a better layout.
- In each tick, after application of all the bound forces (Lines 15ff.), every node’s xy -velocity is decremented by $velocity \times velocityDecay$ and every node’s xy -position is incremented by the current $velocity$ (Line 19). The $velocityDecay$ -parameter can be seen as analogue to atmospheric friction, slowing each node’s velocity down by a factor of $1 - velocityDecay$. Again, a lower $velocityDecay$ rate may yield a better layout at a higher risk of numerical instabilities (unacceptably high deviations from the exact solution) and oscillation

Algorithm 5.5: Velocity Verlet Numerical Integrator

Input: $G = (V, E)$ extracted from a relational dataset
Output: Dynamic straight-line drawing of G

- 1 **Initialize positions:** place vertices of G in phyllotaxis arrangement
- 2 **Initialize velocities:**
- 3 **foreach** *vertex* $v \in V$ **do**
- 4 $\mathbf{v}(v) \leftarrow 0$;
- 5 **Initialize simulation parameters:**
- 6 $\alpha \leftarrow 1$;
- 7 $\alpha_{min} \leftarrow 0.001$;
- 8 $\alpha_{decay} \leftarrow 1 - \text{power}(\alpha_{min}, \frac{1}{300})$;
- 9 $\alpha_{target} \leftarrow 0$;
- 10 $velocityDecay \leftarrow 0.6$;
- 11 $forces \leftarrow \{f_{center}, f_{collision}, f_{link}, f_{charge}, f_{gravityX}, f_{gravityY}\}$;
- 12 **function** TICK()
13 **if** $\alpha > \alpha_{min}$ **then**
14 $\alpha \leftarrow \alpha + (\alpha_{target} - \alpha) \times \alpha_{decay}$;
15 **foreach** *force* $f \in forces$ **do**
16 **foreach** *vertex* $v \in V$ **do**
17 $\mathbf{v}(v) \leftarrow f(\alpha, \mathbf{v}(v))$;
18 **foreach** *vertex* $v \in V$ **do**
19 $\mathbf{p}(v) \leftarrow \mathbf{p}(v) + \mathbf{v}(v) \times velocityDecay$;
20 draw a straight-line segment for each edge;
21 draw a circle for each vertex;
- 22 Execute TICK() every tick until simulation has cooled down.

Figure 5.2.: Cooling of the simulation with the α -parameter.

As stated in Algorithm 5.5, node positions are initialized in a phyllotaxis arrangement. This ensures a deterministic, uniform distribution around the origin. To create greater stability for the exhibition, we store an array of initial positions which yield a good layout and use it for initializing the node positions.

The list of desired drawing properties given in the beginning of this chapter demands for more similar opinions to be drawn closer together. In order to get a rough estimate of the extent to which this criterion is optimized we plot a hexagonal binning (Figure 5.3). It aggregates data points into bins and correlates the edge weights with the edge lengths

in one instance of a drawing produced by the OPMAP algorithm. The Pearson correlation coefficient is -0.2 , with a p -value of $1e-27$, meaning that there is a negative correlation between edge lengths and weights. Accordingly, edges with lower weights tend to have longer edge lengths. While the results look satisfactory, such plots have to be taken with a grain of salt, as the projection of such high-dimensional data into 2D does not necessarily give a representative picture.

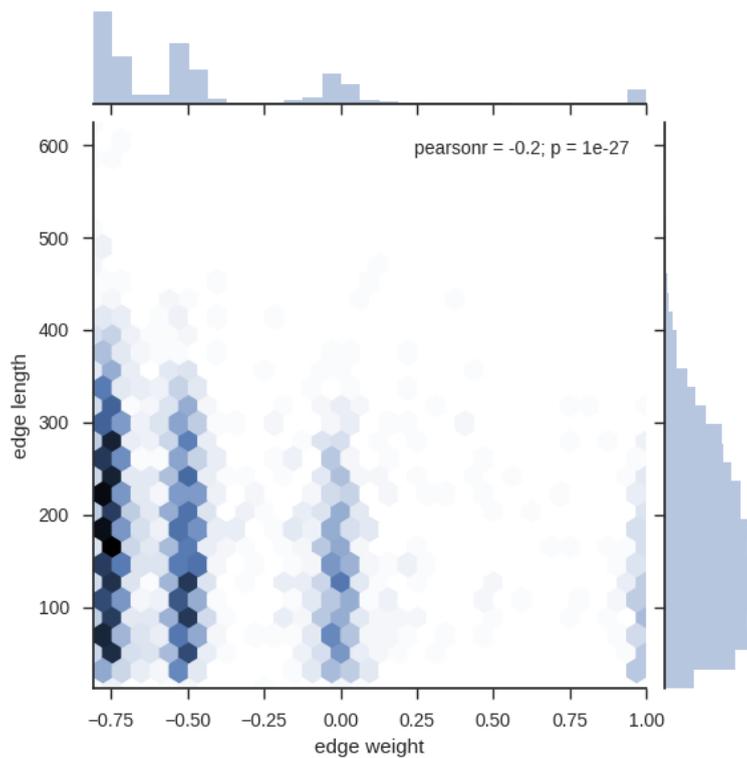


Figure 5.3.: Correlation between edge lengths and (original) edge weights.

As can be clearly seen on the marginal distribution plotted on top, the mutual coherence values are not distributed evenly, but concentrate around particular values. These distributions are caused by our opinion sample, which is well-structured by design of the survey and hence may cause these agglomerations: Pairs of opinions from one and the same cluster dovetail a lot and therefore have high edge weights around 1.0, the maximum mutual coherence value. The values around 0.0 are caused by pairs of opinions from the same cluster that disagree on one or two central reasons, but agree on the core claims. Pairs of opinion which disagree on the core claims have values < -0.4 .

5.2. Voronoi Decomposition

The last step is the one that actually yields the map. Here we use so-called *Voronoi diagrams* (also called *Dirichlet tessellation*). These diagrams subdivide the 2D plane into regions based on the distance to points (in this context often referred to as *sites*) on that plane. Like fractals or the golden ratio, Voronoi diagrams belong to the mathematical phenomena that show up frequently in the natural world: Vein patterns in leaves, cell membranes, whose locations can be estimated using the Voronoi diagram of the nuclei, the wings of a dragon fly, or even drying mud. They are not only visually appealing but have practical applications in many fields. For example, in economy: If a new branch of a supermarket chain is supposed to be opened, a Voronoi diagram can help to predict if

it will be profitable at a certain location ([DBCVKO08], chapter 7). In our case, the set of sites is made up of the vertices in the drawing. In order to calculate Voronoi diagrams, D3 implements Fortune's $\mathcal{O}(n \log n)$ sweep line algorithm [For87]. Before describing the algorithm, we first take a closer look at the properties of the Voronoi Diagram (see also [DBCVKO08], chapter 7). It is beyond the scope of this thesis to provide proofs for these properties, and they are only mentioned as far as they are important to the functionality of the algorithm.

Let $P := \{p_1, p_2, \dots, p_n\}$ be a set of n distinct points in the plane and let $\text{Vor}(P)$ denote the Voronoi diagram of P . The Voronoi cell corresponding to site p_i is denoted by $\mathcal{V}(p_i)$. It consists of all points closer to p_i than to any other point, in terms of the Euclidean distance, which is given by

$$\text{dist}(p, q) = \text{dist}(q, p) := \sqrt{(p_x(p) - p_x(q))^2 + (p_y(p) - p_y(q))^2}.$$

The *bisector* of two points p_i and p_j is defined as the perpendicular bisector of the line segment $\overline{p_i p_j}$. It splits the plane in two open half-planes, where $h(p_i, p_j)$ denotes the half-plane containing p_i and $h(p_j, p_i)$ denotes the half-plane containing p_j . A point q lies in $h(p_i, p_j)$ if $\text{dist}(q, p_i) < \text{dist}(q, p_j)$. Accordingly,

$$\mathcal{V}(p_i) = \bigcap_{1 \leq j \leq n, j \neq i} h(p_i, p_j),$$

that is, the cell that corresponds to the site p_i is the intersection of the $n - 1$ half-planes containing it. The cell is a convex polygonal region bounded by at most $n - 1$ edges (parts of the bisectors of two sites) and at most $n - 1$ vertices (intersection points of the bisectors), see Figure 5.4. Thus, the complete Voronoi diagram is a planar subdivision with straight edges.

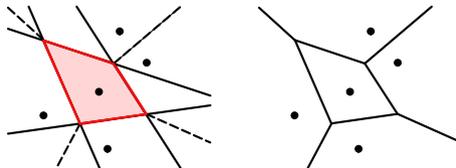


Figure 5.4.: Obtaining a Voronoi cell.

If all the sites are collinear then $\text{Vor}(P)$ is made up of $n - 1$ parallel full lines. Otherwise, the edges of $\text{Vor}(P)$ are either line segments or half-lines, and the union of the edges and vertices form a connected set. In practice, no cell can be an unbound region because the set of sites is surrounded by a bounding box, meaning that all the edges are line segments. When adding an extra vertex v_∞ to the set of vertices of $\text{Vor}(P)$, it becomes a proper planar graph, meaning that Euler's characteristic defined for connected plane graphs holds. It states that

$$|V| + |E| + |F| = 2$$

where $|V|$, $|E|$ and $|F|$ are the numbers of vertices, edges and faces (including the outer face). In the case of the augmented Voronoi diagram:

$$(|V| + 1) - |E| + n = 2$$

where n is the number of sites. Together with the fact that each vertex has a degree ≥ 3 , this implies that the average number of vertices of a Voronoi polygon is less than six,

that is, the complexity of $Vor(P)$ is linear. As there is a quadratic number of bisectors, this means that not all bisectors and intersections between them define edges and vertices of $Vor(P)$. Which bisectors and intersections define the features of the Voronoi diagram can be characterized in terms of the following auxiliary definition: $C_P(q)$ denotes the *largest empty circle of q with respect to P* , that is, the largest circle with center q that does not contain any $p \in P$ (c.f. Figure 5.5a). For each vertex q of the diagram, it has to hold that $C_P(q)$ contains at least three sites on its boundary. A bisector of $\overline{p_i p_j}$ defines an edge of $Vor(P)$ only if it contains a point q such that the boundary of $C_P(q)$ contains only the sites p_i and p_j (c.f. Figure 5.5b).

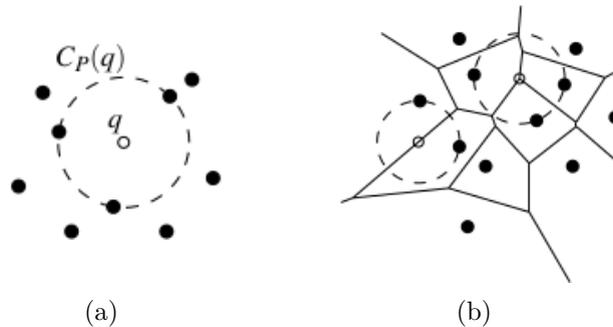


Figure 5.5.: (a) $C_P(q)$, the largest empty circle of q with respect to P . (b) Characterization of the bisectors and intersections that define the features of $Vor(P)$. Image source: [DBCVKO08]

Sweep line algorithms are a common technique in computational geometry. The general idea is to sweep a straight line across the Euclidean plane, while maintaining information about the intersection of the structure to be computed with the sweep line. At certain points, the *event points*, the sweep line stops and the information changes. The sweep paradigm is applied in Fortune's algorithm.

As the sweep line ℓ moves downwards, it intersects the topmost vertices of a cell $\mathcal{V}(p_i)$ before encountering p_i . For that reason, the algorithm additionally maintains a *beach line*, which delimits the part of the diagram above ℓ that cannot be changed any more. Let ℓ^+ be the half-plane above the sweep line. For all points $q \in \ell^+$, that are at least as close to some point $p_i \in \ell^+$ as they are to ℓ , their nearest site is certain (their distance to any site below the sweep line must be greater than their distance to p_i). The points for which this condition holds are bounded by a sequence of parabolic arcs. Each $p_i \in \ell^+$ defines a complete parabola

$$\beta_i := y = \frac{1}{2(p_y(p_j) - l_y)}(x^2 - 2p_x(p_j)x + p_y(p_j)^2 - l_y^2)$$

where l_y denotes the y -coordinate of the sweep line. Now the beach line is given by the piecewise curve that passes for each x -coordinate through the lowest point of all β_i . This is illustrated in Figure 5.6. At any time, the input points above the beach line have been considered in the Voronoi diagram, while the points below it have not been incorporated yet. As the sweep line progresses, the *breakpoints* of the arc segments trace out the edges of the Voronoi diagram. Two kinds of events may change the topological structure of the beach line: Either when a new arc segment appears or when a present arc segment shrinks to a point and then disappears. The first kind of event happens if, and only if, the beach line reaches a new site. This is shown in Figure 5.7.

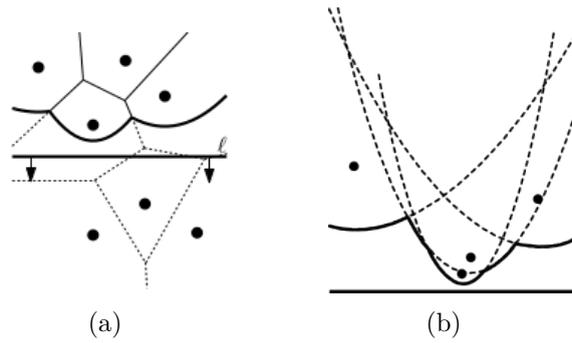


Figure 5.6.: The beach line. Image source: [DBCVKO08]

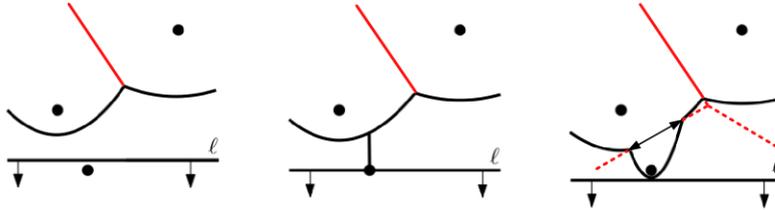


Figure 5.7.: Site event (middle panel). At the moment of tangency of the beach line with a site, the parabola defined by that site is degenerate, with zero width. As the beach line progresses, the parabola gets wider, its breakpoints moving in opposite directions and tracing out an edge of the diagram. Image source: [DBCVKO08]

Initially, the growing edge traced out by the breakpoints of the parabola defined by the newly encountered site is not connected to the rest of the Voronoi diagram. Eventually, it runs into another edge and a new vertex is integrated into the Voronoi diagram. This happens when the second type of event occurs. It is illustrated in Figure 5.8.

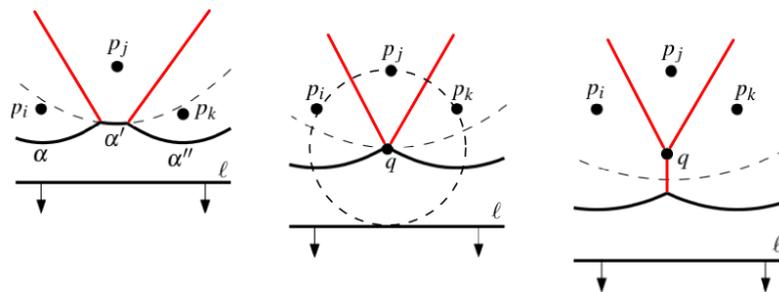


Figure 5.8.: Circle event (middle panel). Arc α' shrinks to a point. Arcs α and α'' are its neighbours before its disappearance. At the moment when α' disappears, all three parabolas pass through a point q , which is equidistant from the three sites that define the arc segments and ℓ , hence the name of this event: there is a circle with q at its center and the three sites on its boundary. Image source: [DBCVKO08]

The arc segments α and α'' cannot be part of the same parabola for the same reason, a new arc segment can appear on the beach line only at a site event: Using the formula for parabolas stated above, and the fact that two sites $p_i, p_j \in \ell^+$ have y -coordinates larger than ℓ_y it can be shown that it is not possible that the two parabolas β_i and β_j have exactly one point of intersection. However, if α and α'' were part of the same parabola, at one moment in time they would have only one intersection point with α' ,

which cannot be. Therefore, it is certain that the arc segments are defined by three distinct sites, p_i , p_j , and p_k in the figure. An arc segment disappears from the beach line only when a circle event occurs. As data structures, the algorithm maintains

- a doubly-connected edge list \mathcal{D} in which the diagram being constructed is stored,
- a balanced binary search tree \mathcal{T} , describing the topological structure of the beach line, and
- a priority queue \mathcal{Q} listing the already known upcoming events that could change the structure of the beach line.

The leaves of \mathcal{T} store the sites that define the arc segments of the x -monotone beach line in order (c.f. Figure 5.9). That is, the leftmost leaf represents the leftmost arc etc. The internal nodes store the breakpoints as ordered tuples. Furthermore, the leaves store pointers to the circle events in \mathcal{Q} in which the represented arc α will disappear, if such an event has been detected yet (otherwise, the pointer is *nil*). The internal nodes store pointers to the edges in \mathcal{D} . The events in \mathcal{Q} are prioritized by their y -coordinate. All the site events can be stored in the beginning, by storing the sites themselves. The circle events are stored by storing the lowest point of a circle, together with a pointer to the leaf in \mathcal{T} representing the arc that will disappear during the event. Every vertex of the diagram is detected by a circle event. The circle events, in turn, are detected as follows: A circle event is always defined by a triple of consecutive arcs. At every event, the resulting new triples are checked for converging breakpoints, and events are inserted to \mathcal{Q} as required. Not every new triple can cause a circle event, for example in a site event, where the new arc is in the middle of a new triple, the arcs left and right belong to the same parabola, meaning that no circle event can be caused. For disappearing triples, the corresponding event in the queue is deleted, if there is one, as it was apparently a false alarm.

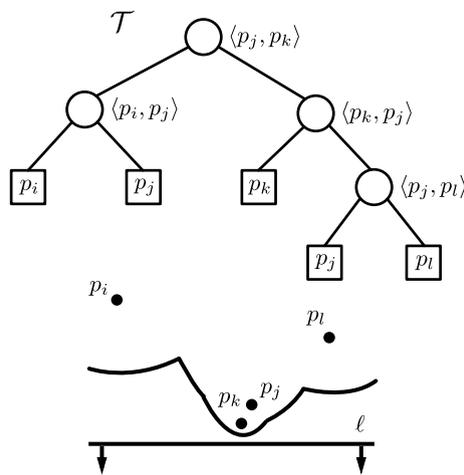


Figure 5.9.: Binary search tree \mathcal{T} used to represent the beach line.

Detailed pseudocode is given in [DBCVKO08], which is shown slightly altered in Algorithm 5.6.

Algorithm 5.6: Fortune's Algorithm

Input: A set $P := \{p_1, \dots, p_n\}$ of point sites in the plane
Output: The Voronoi diagram $Vor(P)$ given inside a bounding box (represented as a doubly-connected edge list \mathcal{D})

- 1 Let \mathcal{T} be the beach line.
- 2 Initialize the event queue \mathcal{Q} with all site events, initialize an empty status structure \mathcal{T} and an empty doubly-connected edge list \mathcal{D} .
- 3 **while** \mathcal{Q} is not empty **do**
- 4 Remove the event with largest y-coordinate from \mathcal{Q}
- 5 **if** the event is a site event, occurring at site p_i **then**
- 6 HANDLESITEEVENT(p_i)
- 7 **else**
- 8 HANDLECIRCLEEVENT(γ), where γ is the leaf of \mathcal{T} representing the arc that will disappear
- 9 The internal nodes still present in \mathcal{T} correspond to the half-infinite edges of the Voronoi diagram. Compute a bounding box that contains all vertices of the Voronoi diagram in its interior, and attach the half-infinite edges to the bounding box by updating the doubly-connected edge list appropriately.
- 10 Traverse the half-edges of \mathcal{D} to add the cell records and the pointers to and from them.
- 11 **function** HANDLESITEEVENT(p_i)
- 12 **if** \mathcal{T} is empty **then**
- 13 insert p_i into \mathcal{T}
- 14 **else**
- 15 Search leaves of \mathcal{T} for arc α vertically above p_i and delete the corresponding circle event in \mathcal{Q} if there is one.
- 16 Replace the leaf with a subtree having three leaves, the middle of which stores the new site p_i , the left and right store the originally stored site p_j . The internal nodes, representing the new breakpoints, store the tuples $\langle p_j, p_i \rangle$ and $\langle p_i, p_j \rangle$.
- 17 Rebalance \mathcal{T} as required.
- 18 Create record for half-edge separating $\mathcal{V}(p_i)$ and $\mathcal{V}(p_j)$ in \mathcal{D} . Check for the triples of consecutive arcs where the new arc is either the left or the right arc if the breakpoints converge and if so, insert circle events into \mathcal{Q} and respective pointers in \mathcal{T} .
- 19 **function** HANDLECIRCLEEVENT(γ)
- 20 **if** \mathcal{T} is empty **then**
- 21 Delete the leaf γ representing the disappearing arc α from \mathcal{T} and update the affected tuples at the internal nodes.
- 22 Rebalance as required.
- 23 Delete all circle events involving α from \mathcal{Q} using the pointers from the predecessor and successor of γ in \mathcal{T} .
- 24 Add the center of the circle as a vertex record to \mathcal{D} .
- 25 Create two half-edge records corresponding to the new breakpoint of the beach line and set the pointers between them appropriately.
- 26 Check for converging breakpoints in the new triples of consecutive arcs, where the middle arc is a former right or former left neighbour of α , add corresponding circle events into \mathcal{Q} and respective pointers into \mathcal{T} .

Once the Voronoi diagram is computed, the cells can be coloured according to the country membership of the site that defines them. Finally, countries are obtained by merging cells of the same colour. As only six countries are distinguished, we do not choose map colours algorithmically. Colour assignment in maps is a research problem of its own. We picked the colours by hand. In geographic maps, readability is enhanced if dissimilar colours are assigned to neighbouring countries. However, we decided to use colour to aid the communication of similarities in the countries, by using similar shades: Countries which support restricted meat consumption are coloured in yellow and orange. Countries which support complete abstinence of meat are coloured in shades of green, and countries which do not advocate any dietary restrictions are coloured in shades of teal (c.f. Figure 5.1). Vegan and vegetarian diets are easily associated with green shades, whereas carnivorous diets are likely to be rather associated with fleshy colours. Nonetheless we wanted to stick to a colour theme that people might remember from their school atlases, like a pastel colour scheme or the vintage safari theme we finally settled on.

Fortune describes in [For87] a modified version of the sweep line algorithm that can construct *additively weighted* Voronoi diagrams, where weights are assigned to the sites. The distance from a site to a point is then given by the Euclidean distance plus its additive weight. As the opinion-graph is vertex-weighted and vertices with higher weights should have more “landmass”, we briefly experimented with weighted Voronoi diagrams. However, as compared to the classical Voronoi diagrams, they have some undesired properties. For example, a site may be outside its zone of influence or have no zone of influence at all. These situations arise when some sites are outweighed by others. This can be seen in Figure 5.10. It shows the vertex-weighted opinion-graph extracted from our empirical dataset and the different versions of the Voronoi diagram drawn on top. In Figure 5.10a the countries are contiguous whereas colouring becomes problematic in case of the weighted Voronoi diagram shown in Figure 5.10b, due to the properties stated above.

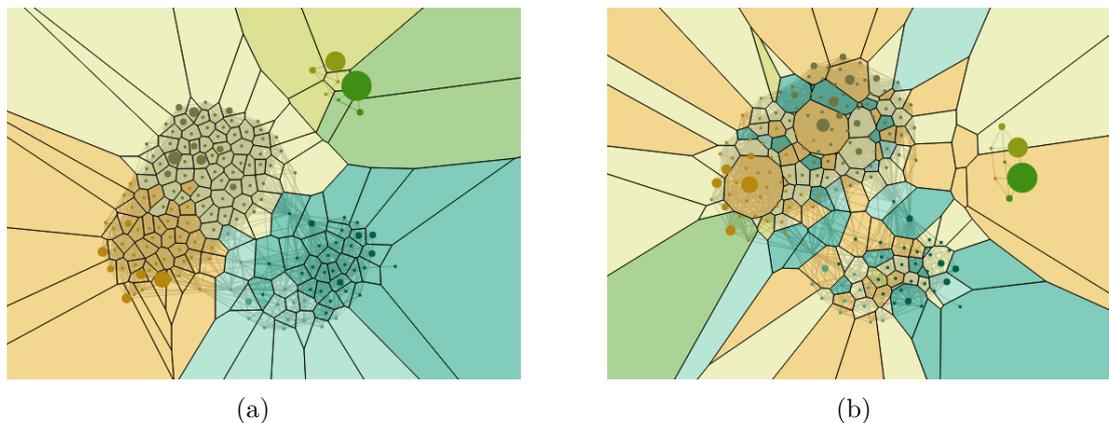


Figure 5.10.: Opinion-graph extracted from the empirical dataset with (a) classic and (b) additively weighted Voronoi diagram. Country membership is indicated by the colours of the opinion sites, weight is indicated by the circle radii.

We therefore decided to optimize cell sizes by means of the charge force, as described in the last subsection.

6. Implementation Details

This chapter states the programming languages and tools that were required—from gathering an opinion sample to the interactive web page. Figure 6.1 provides a rough overview of the architecture of the interactive application: The first front-end component is the online survey (c.f. Section 6.1), which sends a new opinion node to the back-end Mathematica module (c.f. Section 6.2), where the deductive closure and the coherence values are calculated. The server and the database were kindly set up by Christian Voigt. After a post-processing step (c.f. Section 6.3), the new opinion node along with its adjacent edges are send to the second front-end component, a web page programmed with JavaScript, or more specifically, the library D3.JS (c.f. Section 6.4).

6.1. Online Survey

One of the first steps was to gather some initial opinion data. As we needed a big opinion sample as quickly as possible, we published an online survey. We set up a bilingual web page using the scripting language PHP to dynamically generate HTML content. Michael Hamann kindly provided a PHP-script to use as a basis for building the survey. It was hosted at a server of the Institute of Theoretical Informatics and can be found under <http://i11www.itl.kit.edu/~svschmettow/index.php?lang=en>.

Data was saved into an SQLITE database¹, exported as .csv-file and converted to the correct .json-output format with the help of a parser implemented in Python.

In the interactive application, we use the same survey as the one used for collecting the initial dataset (blue box in Figure 6.1). However, we do not use PHP but only the generated HTML along with JavaScript to dynamically display the next page of the survey based on the selected diet and to post-process an entered opinion which is then sent as a JSON-object to the Mathematica-module (green box in Figure 6.1) via an AJAX request.

¹<https://www.sqlite.org/>

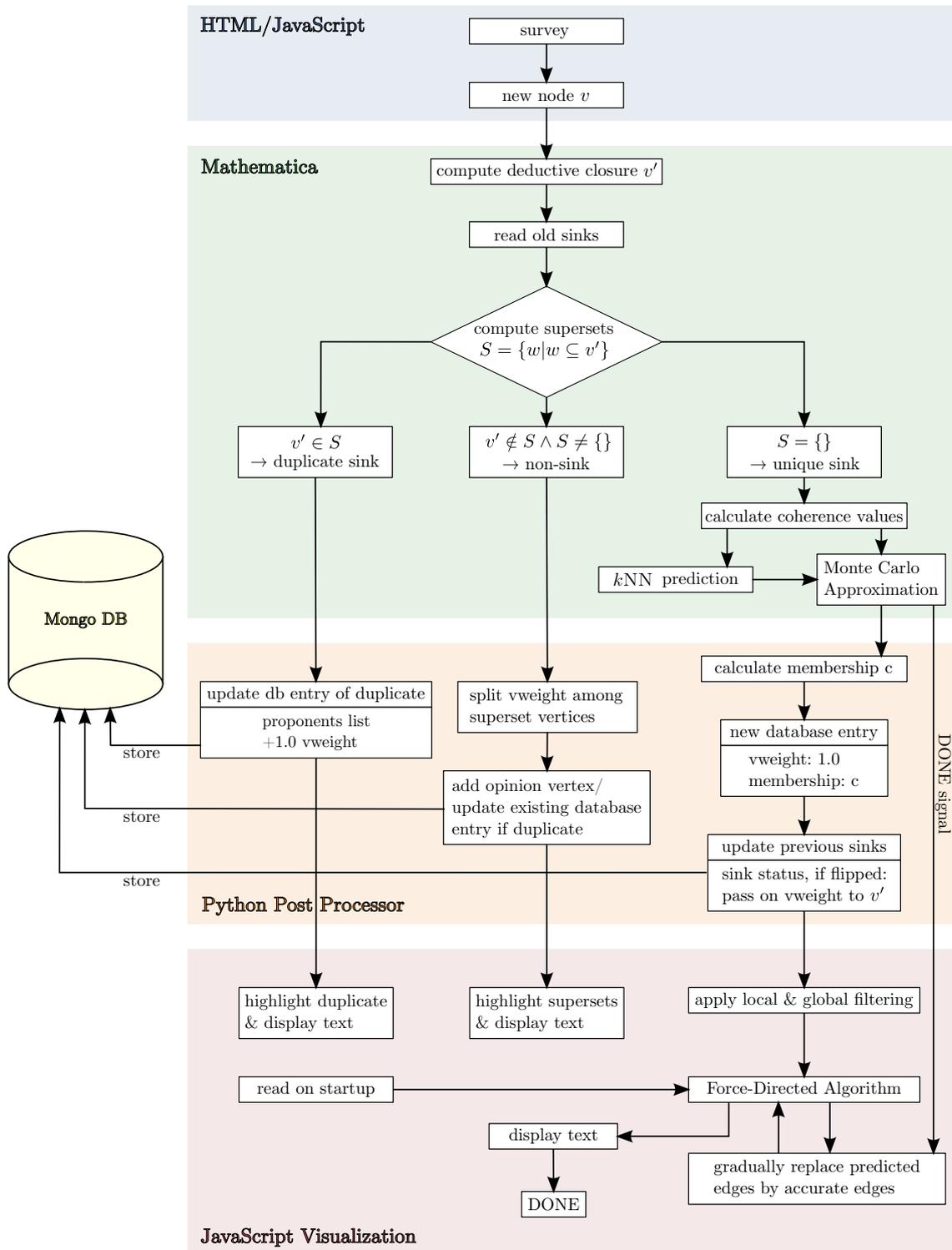


Figure 6.1.: Overview of the OPMAP architecture.

In the exhibition, the survey web page is displayed on an iPad. The browser app is opened in the **guided access** mode featured in iOS, which prevents visitors to navigate away from the survey. If a user navigates to the second page of the survey but stops its completion before sending any data, the web page displays a warning with a 20 second countdown before reloading the page after one minute of no interaction (tapping or scrolling). This can be stopped by touching a button or tapping anywhere on the screen.

6.2. Generating Graph Files

Graph files were generated in Mathematica. The Mathematica module was implemented by Gregor Betz and makes use of his logical reconstruction of the Veggie-Debate, given in Argdown syntax. Argdown documents can be exported to `.json` files, which, in turn, can be interpreted by Mathematica. In order to extract a graph from an opinion sample we generated a complete weighted graph of the opinions collected in the survey. They are processed as JSON-objects in the form:

```
{"opinion": ["1", "2", "14", "24", "!24"], "id": "xy"}
```

As a preprocessing step, the deductive closure is computed for every opinion in the sample. The weights are given by the degree of mutual coherence between pairs of the deductively closed opinions. This measure is calculated on the basis of the attack and support relations encoded in the logical reconstruction of the debate.

If \mathcal{A} and \mathcal{B} are opinions over a sentence pool S with $|S| = 53$, each of these positions has 2^{53} possible subsets, as each subset either contains an element of the position or does not contain it. $MutCoh(\mathcal{A}, \mathcal{B})$ (Definition 3.7) is the averaged sum of the degree of support of \mathcal{A} for \mathcal{B} and vice versa, therefore one would have to calculate the Kemeny-Oppenheim measure (Definition 3.8) 2^{54} times to determine its *exact* value. As this is not feasible, we use a *Markov chain Monte Carlo* (MCMC) sampler with a precision of 500. This basically means that 500 subsets are randomly sampled to approximate the actual mutual coherence.

The green module in Figure 6.1 shows how a new opinion node is processed in the interactive application: First, its deductive closure is computed. Second—before any edges are calculated at all—three cases are distinguished. To that end, a property `"super"` is calculated. It is an array of the ids of all sink nodes representing positions which are supersets of the current node.

- If `"super"` is an empty array the current node is a unique sink (right case in the figure).
- If `"super"` contains only one element, the id of a node that features the same `"opinion"`-property, it is a duplicate sink (left case).
- If the array of supersets does not include the id of a node representing the same opinion and is non-empty, the current node is a non-sink (middle case).

Only if the new opinion is a unique sink opinion, edges are calculated. Despite the applied heuristic, the process of calculating an edge for every other opinion in the opinion-graph would keep a user of the interactive application waiting too long. Therefore a less time-complex method is required to calculate the edge weights of a new opinion. More precisely, a method to predict *which* edges should be calculated using MCMC approximation. As it turned out, mutual coherence can be reliably interpolated as a function of *positive and negative overlap* of two positions. Let $\mathcal{A} = \{s_1, \dots, s_n\}$ and $\mathcal{B} = \{s_1, \dots, s_n\}$ be two positions on the sentence pool S where $s_j \mapsto \{true, false, _ \}$. The positive and negative overlap between \mathcal{A} and \mathcal{B} is given by

$$k_+(\mathcal{A}, \mathcal{B}) = \frac{1}{|S|} |\{s_j | \mathcal{A}(s_j) = \mathcal{B}(s_j)\}|$$

$$k_-(\mathcal{A}, \mathcal{B}) = \frac{1}{|S|} |\{s_j | \mathcal{A}(s_j) = \neg \mathcal{B}(s_j)\}|$$

That is, k_+ is the number of sentences the positions explicitly agree on and k_- is the number of sentences they explicitly disagree on, normalized by the total number

of sentences in the pool. Figure 6.2 shows that the overlap-coordinates and mutual coherence values are highly correlated.

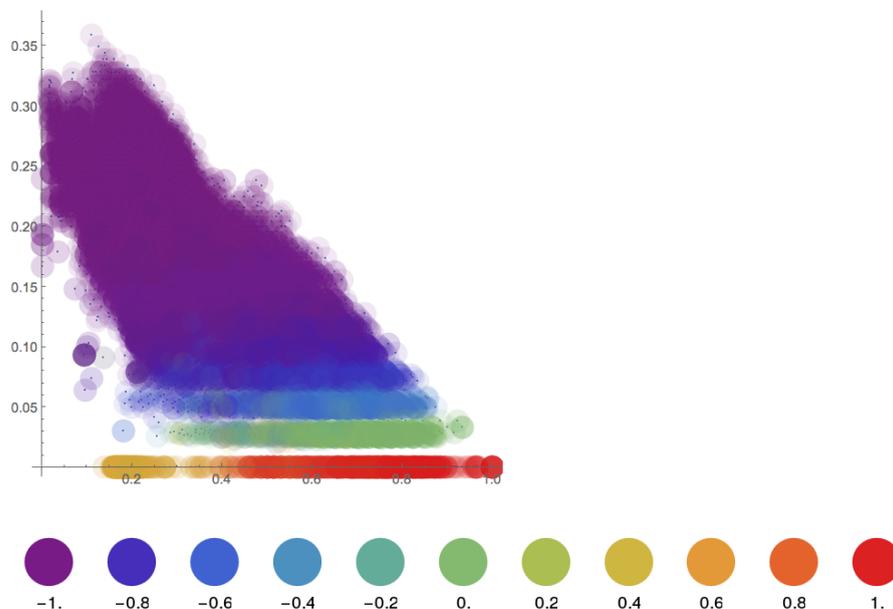


Figure 6.2.: Interpolating mutual coherence as function of the positive and negative overlap, shown on the x - and y -axis, respectively. Mutual coherence is given by the colour of the data points, as indicated by the legend below.²

On that ground, we use Mathematica’s integrated machine learning capability `Predict`³, a highly automated function which can learn a `PredictorFunction` on the basis of a training set consisting of input-output pairs. Using a training set comprising 21528 training examples a *nearest neighbours classifier* was trained, which accepts two-dimensional numerical feature vectors (k_+, k_-) for a pair of positions, that can be obtained at very low computational cost. The resulting *k-Nearest-Neighbours* (k NN) algorithm compares such a vector to the k nearest neighbours in the feature space, which yields a good estimate of the actual mutual coherence. Interestingly, there is a direct relation between the k NN algorithm and Voronoi diagrams: If $k = 1$, the classification of a new vector is performed using the Voronoi diagram of the feature space—the new vector is assigned the class of the site of the respective Voronoi cell it is located in.

As can be seen in Figure 6.1, the more accurate mutual coherence value is calculated dependent on the prediction: Only for those edges where the absolute value of mutual coherence estimated with the k NN algorithm exceeds a threshold of 0.6 the value is recalculated with the more time-complex MCMC sampler. The Mathematica module communicates with the server via standard input and output. When starting to calculate edge values with MCMC approximation, it communicates to the server how many edges will be calculated, such that the visualization module can inform the user about the calculation progress to grant a more satisfying user experience. Furthermore, it sends a DONE signal to the visualization, once calculations are complete.

6.3. Processing Graph Files

Graph files were post-processed using the network analysis package IGRAPH that can be programmed in Python, among others. Complete opinion-graphs can be exported from

²This figure was generated by Gregor Betz in Mathematica.

³<http://reference.wolfram.com/language/ref/Predict.html>

Mathematica in `.dat` or `.json` format. Parsers to convert between `.dat`-files, `.json`-files, `.dot`-files and IGRAPH objects as needed were implemented at first. The following steps are executed, once the export of the complete graph has been converted to an IGRAPH object:

1. DAG filtering is performed on the vertices of the opinion-graph, as explained in Section 4.2.
 - An entailment-DAG is created from the input graph, where nodes that represent duplicate opinions are removed and added as vertex weights instead.
 - Next, a new vertex attribute of the type `string` is added to the opinion-graph object using the corresponding vertices in the DAG, which are identified by the same `ids`: `DAGnodetype ∈ {"internal", "source", "sink", "isolated"}`
 - Using this attribute, all sources and internal nodes including their incident edges are deleted, as well as vertices which represent a duplicate opinion.
 - In order to assign vertex weights to the opinion-graph, sources are recursively removed from the entailment-DAG while splitting their weights evenly among the children. When only isolated nodes remain, their accumulated vertex weights can be assigned to the corresponding nodes in the opinion-graph.
2. Local and global edge filtering is applied to the edges of the opinion-graph, as described in Section 4.3
3. Clustering attributes for the vertices in the filtered graph are calculated. Various vertex attributes are assigned as required, for example `membership_ml`, `membership_im`, `membership_lp`, `membership_sg`, and `membership_wt` to indicate the cluster assignment according to the different algorithms described in Chapter 4.

For additionally investigating if the clusterings are semantically meaningful, we used Gephi, which allows flexible filtering and colouring of vertices and edges according to the attributes. To that end, we added vertex attributes indicating for each statement if it is supported, rejected, or not considered in the respective opinion. We then exported graph objects as `.graphml`-files, that can be interpreted by Gephi.

For the interactive visualization, we use MONGODB, a NoSQL database program using JSON-like documents, together with PYMONGO, the Python interface. In order to prepopulate the database with the initial opinion-graph, it is exported to JSON, or rather BSON (binary JSON) format. For the sake of completeness, the database is supposed to contain all opinions that have ever been stated, including non-sinks. The final JSON-file is structured as follows:

```

{
  "nodes": [
    {"id": "a", "opinion": ["1", "7", ..., "!23", ...], "membership": 1, ...},
    {"id": "b", "opinion": ["1", "3", ..., "!37", ...], "membership": 2, ...},
    ...
    {"id": "z", "opinion": ["2", "5", ..., "!49", ...], "membership": 5, ...}
  ],
  "links": [
    {"source": "a", "target": "b", "value": -0.998},
    {"source": "a", "target": "c", "value": -0.755},
    ...
    {"source": "x", "target": "z", "value": -0.999}
  ]
}

```

Listing 6.1: Opinion Nodes

It has two properties, "nodes" and "links", the values of both of which are arrays of JSON-objects with further property-value pairs as elements. For node objects in the database we store the following properties:

- "id", a unique identifier in the form of a hexadecimal number, generated with `uuid4()`, provided by the Python module `UUID` that enables the creation of immutable Universally Unique Identifier objects, as specified in RFC 4122⁴.
- "opinion" and "closure", the original and deductively closed positions as arrays of strings.
- "sink", a boolean value that indicates if the node corresponds to a sink (or isolated node) in the current entailment-DAG.
- "vweight", a float indicating the vertex weight.
- "super", as provided by the `Mathematica` module.
- "membership" $\in \{-1, 0, \dots, 5\}$, the cluster membership according to the infomap algorithm. If the value of "sink" is `false`, this value is set to -1.
- "proponents", a list of users who hold exactly this opinion. This was introduced in order to not create multiple node entries in the database for the same opinion. The value of this property is a list, featuring a JSON-object that contains as string values the selected diet option, the demographic data of the proponent ("age", "gender", "education", and "residence", as well as a timestamp "added" whose value is itself a JSON-date object.
- "added", another timestamp which indicates when the node entry was first created.
- "load_on_startup", another boolean value which indicates which nodes are supposed to be considered in the map when the application is reset.

"membership", in case of sink nodes, references one of the "countries":

```
{  
  0: "Ethical Gourmania",  
  1: "Moderatia",  
  2: "Omnivorica",  
  3: "Traditionalistan",  
  4: "Veganland",  
  5: "Vegetaristan"  
}
```

The nodes are connected by links, which have three properties, `source`, `target` and `value`. The first two are ids referencing one of the nodes. The latter is the original mutual coherence value between the source and target opinion. The global filtering constant is re-added in the visualization module. The post-processing of a new opinion with Python is shown in the orange module in Figure 6.1: The properties "opinion", "closure", and "super" have already been provided by the `Mathematica`-module. Three cases are distinguished:

- If the new opinion is a unique sink, its cluster assignment is calculated according to a simplified algorithm (given in Algorithm 6.1) and the property "membership" is set accordingly. "vweight" is set to one, "sink" is set to `true`, "proponents" is initialized as an empty array and timestamps and identifier are generated. Furthermore, it has to be checked if the sink status of any of the previous nodes has changed because the new position is a superset of any of the previous sink nodes. If so, the status in the respective entry has to be flipped, "vweight" is set to 0 and "vweight" of the current node is incremented by 1.

⁴<https://tools.ietf.org/html/rfc4122>

- If the current node is a duplicate sink, the "vweight"-property of the existing database entry is incremented by 1 and the "proponents" array is *upserted*⁵.
- If the current node is a non-sink, its vertex weight is divided evenly among the nodes listed in "super". If the non-sink opinion has been stated before, merely the "proponents"-property of the corresponding database entry has to be updated. Else, a new entry with "membership":-1 and "vweight":0 is inserted to the database.

The clustering algorithm for new nodes is not nearly as elaborate as the one used to cluster the initial data. We simply find the edges the node has to each cluster and sum up their weights. These sums are then normalized by the number of nodes in each cluster, in order to prevent that nodes are more likely to be assigned to the bigger clusters. The node is then assigned to the country with the highest corresponding normalized sum of edge weights.

Algorithm 6.1: Clustering Algorithm for New Nodes

Input: node v with set of adjacent edges E
Output: cluster membership c of v

- 1 Let $|\mathcal{C}|$ be the number of clusters in the determined clustering \mathcal{C} and S be an array that is used to store for each $C \in \mathcal{C}$ the sum of weights of edges v has to the respective cluster.
- 2 Initialize $c \leftarrow -1$, $S[i] \leftarrow 0 \forall i = 0, \dots, |\mathcal{C}|$
- 3 **foreach** $e \in E$ **do**
- 4 $S[\text{membership}(e)] \leftarrow S[\text{membership}(e)] + \text{weight}(e)$
- 5 **foreach** $s_i \in S$ **do**
- 6 $s_i \leftarrow \frac{s_i}{|C_i|}$
- 7 $c \leftarrow \arg \max_{s_i \in S} i$;
- 8 **return** c

The updates of the "vweight" and "sink" properties later turned out to be unnecessary: Due to limits in computing power of the computer connected to the beamer in the exhibition we decided to reset the map every night and restart with the initial 208 opinions. Only for these, the "load_on_startup"-flag is set to `true`.

6.4. Interactive Visualization

Our web application is supposed to provide structural insight into a large opinion sample. The raw data is difficult to understand and analyse, therefore we needed to choose state-of-the-art technologies that would allow us to communicate information clearly and efficiently. We use D3.JS as primary data visualization library which is integrated with the Python backend to serve all the data and other services.

The goal was to create an interactive and dynamic force-directed graph similar to an example by D3's creator Mike Bostock himself [Bos17a]. It is based on D3-FORCE and uses the force simulation API⁶.

The simulation takes a list of data objects which are used as nodes, in our case the opinions as given in Listing 6.1. At first, an `svg` element needs to be selected with D3 which functions as canvas for the graph; `width` and `height` are set to fill the window.

⁵This is database terminology for updating an existing entry.

⁶<https://github.com/d3/d3-force#simulation>

```
<body>
  <div id="viz"></div>
</body>
<script type="text/javascript" charset="utf-8">
  var width = innerWidth,
      height = innerHeight;
  var svg = d3.select("#viz")
    .append("svg")
    .attr("width", width)
    .attr("height", height);
</script>
```

Then, a `forceSimulation` instance needs to be created and forces are added.

```
var simulation = d3.forceSimulation();
function initializeForces() {
  // add forces and associate each with a name
  simulation
    .force("link", d3.forceLink())
    .force("charge", d3.forceManyBody())
    .force("collide", d3.forceCollide())
    .force("center", d3.forceCenter())
    .force("forceX", d3.forceX())
    .force("forceY", d3.forceY());
  // apply properties to each of the forces
  setForces();
}
```

As explained in Section 5.1, forces are functions which influence the nodes' positions and velocities. We add six different forces which are adapted to optimize the desired drawing properties, e.g. that country areas correspond roughly to the number of represented opinions and proximities on the map reflect semantic proximities in the data.

```
function setForces() {
  // get each force by name and set the properties
  simulation.force("center")
    .x(width * 0.5)
    .y(height * 0.5);
  simulation.force("charge")
    .strength(adjustCharge)
    .distanceMin(1)
    .distanceMax(2000);
  simulation.force("collide")
    .strength(0.7)
    .radius(adjustRadius)
    .iterations(5);
  simulation.force("forceX")
    .strength(0.1)
    .x(width * 0.5);
  simulation.force("forceY")
    .strength(0.2)
    .y(height * 0.5);
  simulation.force("link")
    .id(function(d) {return d.id;})
    // .strength(adjustLinkStrength)
    .distance(adjustDistance)
    .iterations(1)
    .links(graph.links);
  // restarts the simulation (important if simulation has already slowed down)
  simulation.alpha(1).restart();
}
```

The map is composed of individual cells which correspond to each node. In order to draw the node elements, an `svg` group element is added to the `svg` canvas, using the `nodes`-array as data source:

```

var node = svg.selectAll('.node')
    .data(graph.nodes, function(d) { return d.id;});
node.enter().append('g')
    .attr('title', name)
    .attr('class', 'node')
    .attr("id", function (d) {
        return "n_"+d.id; //the "n_" ensures that the id is valid, i.e. begins with a
        letter
    })

```

As it is done with D3, svg elements are now *bound* to the data:

```

var cell = node.append("polygon")
    .attr('class', 'nodepoly')
    .attr('points', function(d){
        var adjustedRadius = radius*log(10, 9+d.vweight);
        return irregularNGon(d.id, {'x':0, 'y':0}, adjustedRadius);})
    .attr('fill', colorByMembership)
    .attr('stroke', "#8bc4d6");

```

The above listing sets the properties of the svg polygon elements. Their shapes are obtained using `irregularNGon`, a function of the node's identifier. As stated in the last section, the node elements are randomly assigned unique identifiers on the server side. These identifiers are 32-digit hexadecimal numbers given as string values, for example "2ac531f6-e4fc-43fc-8f45-d1c051045686". In order to make the outer boundaries of the map more natural, an apparently random polygon is generated for each node using its id. As polygons are recalculated when a new node is added to the graph, a completely deterministic algorithm, that will always generate the same polygon for the same input, was required. There are infinite ways to come up with such a process. The one applied here is given in Algorithm 6.2 and produces polygons that create a rather natural-looking outer border. They are obtained by splitting the input id into 16 pairs, which are interpreted as smaller hexadecimal numbers. After being converted to integers and normalized to be in a range between 0 and 1, they are used to increment the angle α in irregular steps to calculate points that lie on a circle in irregular distances around the target center point p_{center} . As these steps were on average a little too small for the polygons to look good, the stepsize is modified with the δ -parameter.

Algorithm 6.2: IrregularNGon

Input: Hexadecimal 32-digit node id x_{16} , node position p_{center} , target radius r

Output: An array of points $P = [p_1, \dots, p_n]$, $10 \leq n \leq 20$ that define an irregular polygon and lie on a circle with center p_{center} and radius r

- 1 Initialize $minPoints \leftarrow 10$, $maxPoints \leftarrow 20$, $P \leftarrow []$,
angle $\alpha \leftarrow 0$, step size $\delta \leftarrow 1.3$
 - 2 $A \leftarrow$ array of smaller numbers obtained by splitting x_{16} into smaller strings of
length two and interpreting them as numbers
 - 3 Convert all a in A to integers
 - 4 Normalize s.t. all $a \in [0, 1]$
 - 5 **for** $i = 1, \dots, maxPoints$ **do**
 - 6 $\alpha \leftarrow \alpha + \delta * A[i \bmod |A|] * \frac{2\pi}{minPoints}$
 - 7 **if** $\alpha > 2\pi$ **then**
 - 8 | break;
 - 9 $p_{i,x} \leftarrow p_{center,x} + r \cdot \cos(\alpha)$
 - 10 $p_{i,y} \leftarrow p_{center,y} + r \cdot \sin(\alpha)$
 - 11 $P[i] \leftarrow p_i$
 - 12 **return** P
-

Note that the radius was set as a function of the decadic logarithm of the node's vertex weight. Most nodes represent unique opinions and therefore have a vertex weight 1. In that case, the radius should correspond to the nominal radius. The nominal radius is adapted every time the graph is updated. It is set to a third of the average edge length. As $\log_{10}(10) = 1$ the constant 9 was added to ensure this property. The polygons are coloured according to their country membership.

Building on one other D3-example [Cou16], the inner boundaries are obtained by calculating a Voronoi decomposition based on the node center points.

```
var voronoi = d3.voronoi()
  .x(function(d) { return d.x; })
  .y(function(d) { return d.y; })
  .extent(clipPoly);

function recenterVoronoi(nodes) {
  var shapes = [];
  voronoi.polygons(nodes).forEach(function(d) {
    if ( !d.length ) return;
    var n = [];
    d.forEach(function(c){
      n.push([c[0] - d.data.x, c[1] - d.data.y]);
    });
    n.point = d.data;
    shapes.push(n);
  });
  return shapes;
}
```

`voronoi.polygons(nodes)` gives for each node an array of 2D-coordinates which form the respective Voronoi polygon. These polygons are used as clip-paths for the svg polygons bound to the nodes. The `clip-path` svg element defines a clipping path, which restricts the region to which paint can be applied, that is, any parts of the drawing that lie outside of this region bounded by the currently active clipping path are not drawn. `recenterVoronoi` is called every time the nodes are redrawn, i.e. in each tick of the simulation. Using the nodes as sites, it calculates the vertices of the Voronoi polygon relative to each node's *xy*-position.

The actual graph is drawn on top of the map. To do so, `svg` `line` and `circle` elements are used, analogous to the node polygons.

```
var link = svg.selectAll('.link').data(graph.links);
link.enter()
  .append('line')
  .attr('class', 'link')
  .style('stroke-width', 1)
  .style('stroke-opacity', 0.5);

node.append('circle')
  .attr('r', function(d) { return get_attribute(d, 'vweight');})
  .attr('fill', '#595959');
```

The radii of the circles which represent the nodes are scaled directly proportional to the vertex weight of the node. Recall that the vertex weight represents the number of times this exact opinion or respectively an opinion that is extended by it has been stated. This fits neatly to the map analogy: The circles are like cities, scaled according to the number of citizens.

The country labels are created using another `svg` `group` element. The font size of the country labels is proportional to the number of opinions that are represented by a cluster, that is, the total weight of a cluster: $30px \cdot \log_{25} \sum_{e \in C} w(e)$. As a country may become fragmented, the labels cannot simply be positioned at the mean *xy*-coordinate of a cluster. Instead, they are positioned at the median.

Finally, the simulation needs to be started and a tick function is required that is executed on every simulation tick. In this function, all the coordinates of the nodes, clip points, links and text elements are updated.

```
function initializeSimulation() {
  simulation.nodes(graph.nodes);
  initializeForces();
  simulation.on("tick", ticked);
}

function ticked(e) {
  redrawLink(link);
  redrawNode(node);
}

function redrawLink(link) {
  link.attr('x1', function(d) { return d.source.x; })
  .attr('y1', function(d) { return d.source.y; })
  .attr('x2', function(d) { return d.target.x; })
  .attr('y2', function(d) { return d.target.y; });
}

function redrawNode(node) {
  node.attr('transform', function(d) { return 'translate('+d.x+', '+d.y+')'; })
  .attr('clip-path', function(d) { return 'url(#uclip-'+d.index+')'; });

  var clip = svg.selectAll('.mclip')
  .data(recenterVoronoi(graph.nodes), function(d) {
    return d.point.index; });
  clip.enter().append('clipPath')
  .attr('id', function(d) { return 'uclip-'+d.point.index; })
  .attr('class', 'mclip');
  clip.exit().remove();
  clip.selectAll('path').remove();
  clip.append('path')
  .attr('d', function(d) { return 'M'+d.join('L')+'Z'; });

  clabel.attr("x", function(d) {return median(d.cluster)[0];})
  .attr("y", function(d) {return median(d.cluster)[1];});
}

```

Furthermore, the map generates statistics about the underlying dataset (Figure 6.3), which are periodically displayed and updated. These donut charts are created with D3's pie chart functionality, provided in the module `D3-SHAPE`⁷. As can be seen in Figure 6.4a, these statistics show that the country sizes correspond quite well to the number of opinions that inhabit them.

User Interaction & Dynamics

The processing of a new node in the visualization module is sketched in the red box in Figure 6.1. When a new node is received it is only integrated on the map, if the corresponding position is a consistent sink node in the entailment-DAG that has not been entered before. The following messages are displayed (for variable durations), depending on the case:

- Case 0: If the opinion is logically inconsistent:
We aren't able to interpret your opinion in a consistent way. Please reconsider your answers. (30s)
- Case 1: If the opinion is a non-sink (middle case in the figure):
Your opinion generalizes a number of other opinions, which are highlighted on the map. (60s)

⁷<https://github.com/d3/d3-shape/blob/master/README.md#pies>

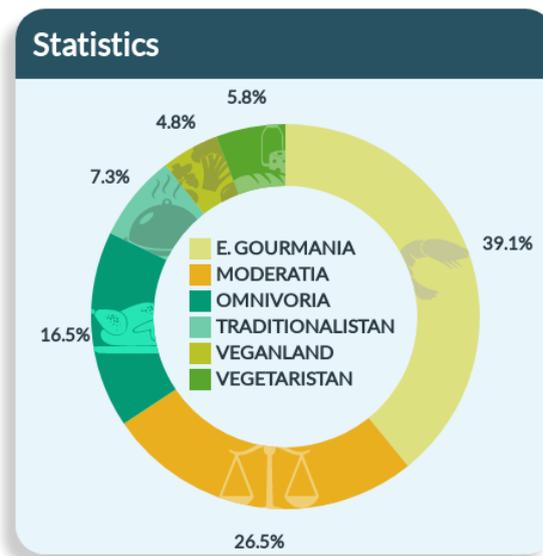


Figure 6.3.: Donut chart depicting the underlying distribution of opinions.

- Case 2: If the opinion is a sink but has been stated before (left case):
Your opinion has already been stated by someone else, it will now be highlighted on the map. (15s) Afterwards, a description of the respective country is shown (60s).
- Case 3: If the opinion is a unique sink, the whole pipeline is triggered, starting with the message:
A new node, that represents your opinion, will now be inserted to the map. (15s) Then, the new node is shown travelling to its destination country. The polygon is coloured gray at first. Once it has reached its country, it is coloured in the corresponding shade. Finally, the country description is displayed as in Case 2.

In case 0 merely the message is displayed. In cases 1 - 3 we use the familiar *Point of Interest* icon for highlighting opinion nodes. In case 1 the current position is too unspecific and is extended by other positions on the map. These are marked with the POI icons (Figure 6.4b). For cases 2 and 3 the map looks like shown in Figure 6.4c.

In case 2 the duplicate opinion is marked on the map and the corresponding country description is displayed for 60s. Case 3 is the only case where a new opinion node is added to the map. Only in that case the Mathematica module calculates edge weights which are sent to the visualization module (c.f. Figure 6.1). The approximation of all the edge weights (c.f. Section 6.2) is calculated almost immediately. On the basis of this estimation a new node along with its adjacent edges is added to the force-directed graph. In order to prevent the node from getting “entangled” on its way to its own cluster, its coordinates are not initialized randomly but set to one of the four corners of the canvas, whichever is closest to the predicted country. Also, we limit the number of edges inserted for every node to 10 by using random filtering. The accurate edge weights start to come in one after another. For each new accurate edge value we check if it was among the randomly chosen edges. If yes, the corresponding estimated weight is replaced by the accurate weight, as long as it is non-negative after adding the global filtering constant. If this is not the case the edge is removed and, if possible, another random edge is selected. If the edge was not included in the selection but would be included with the updated weight, then it should be given the possibility to be selected. Therefore, an index in the range $[0, deg)$ is chosen, where deg is the unfiltered degree. If the index is in the range $[0, 10)$, the edge at that position is replaced with the new edge.

The new node is also marked by a POI-icon, which moves along with the node as it is inserted. For this, another function needs to be added to the tick function, to update the coordinates of the icon according to the coordinates of the node:

```
function ticked(e) {
  redrawLink(link);
  redrawNode(node);
  redrawLoc();
}

function redrawLoc() {
  d3.select("#poi")
    .attr("x", function(d){return newNode.x-locWidth/3;})
    .attr("y", function(d){return newNode.y-locHeight;});
}
```

Once all edges are calculated the clustering is recalculated with Algorithm 6.1. Very rarely, cluster membership changes due to the accurate edges. In that case the description of the actual country is displayed, reading Whoops, our prediction was wrong. in the footer of the message box.

The final product at the Open Codes exhibition is shown in Figure 6.5

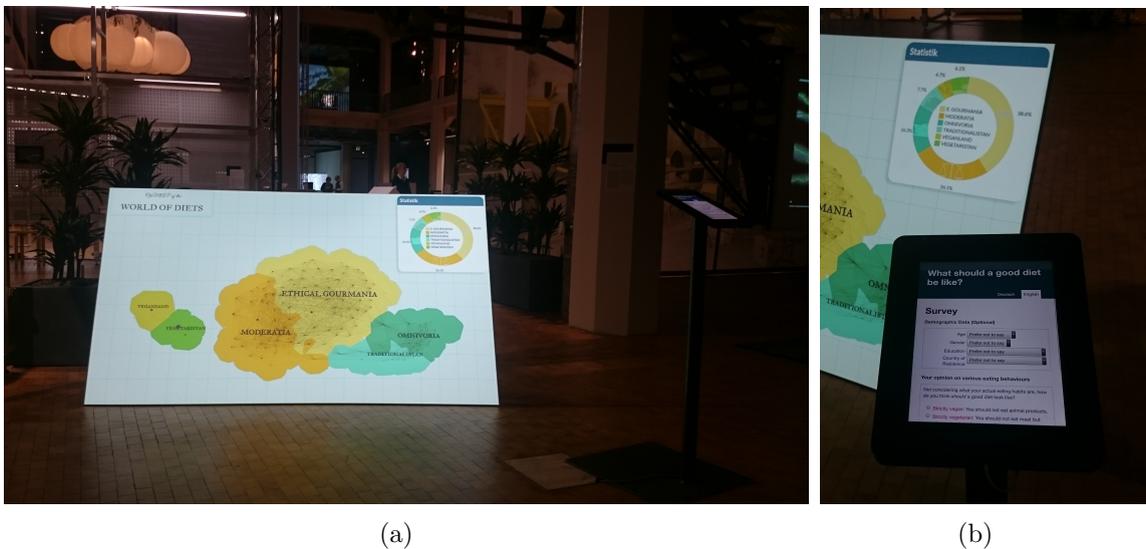
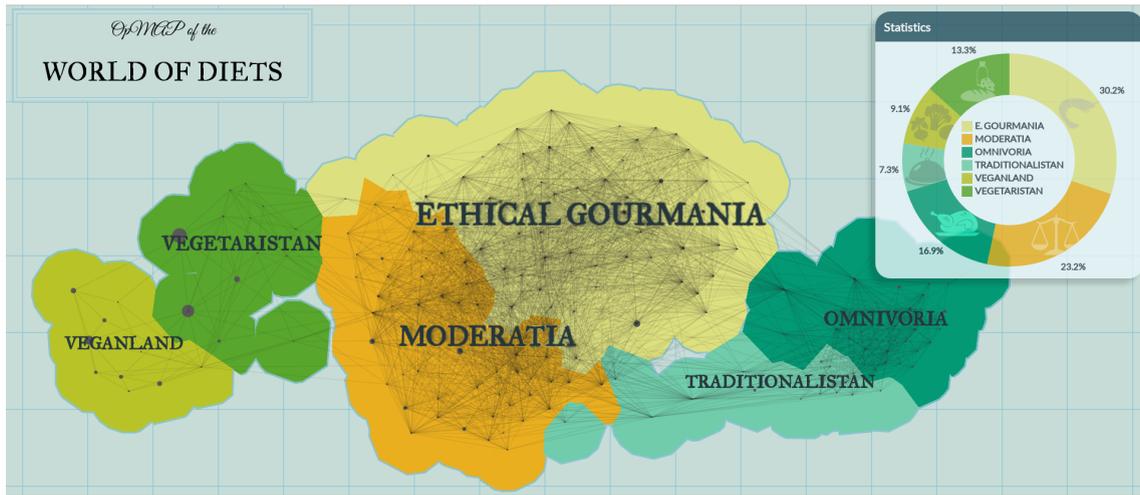
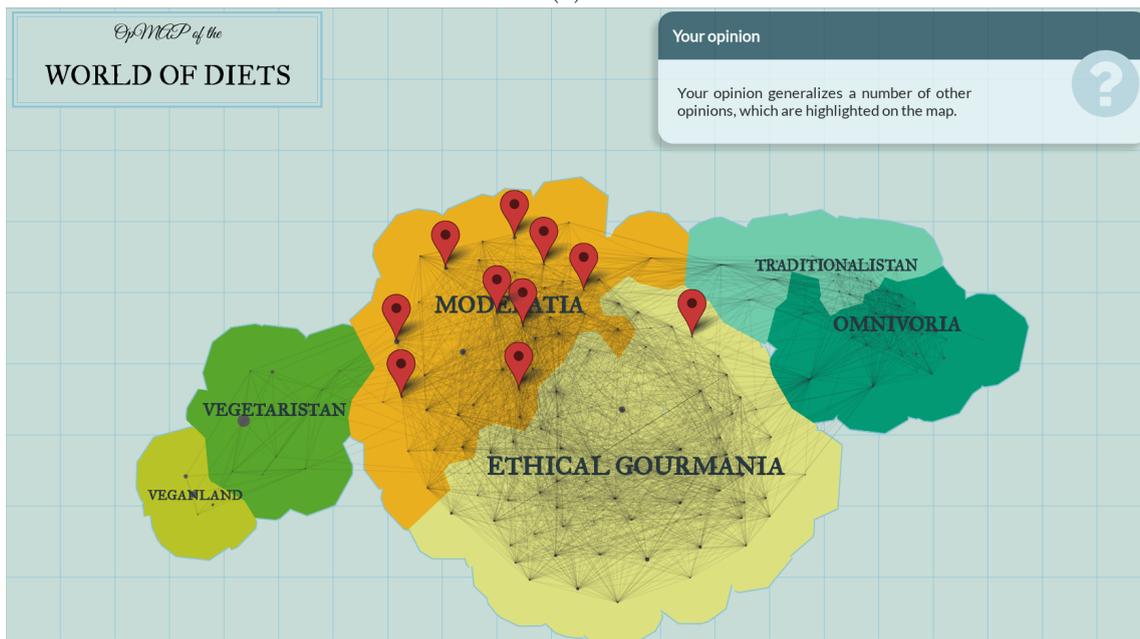


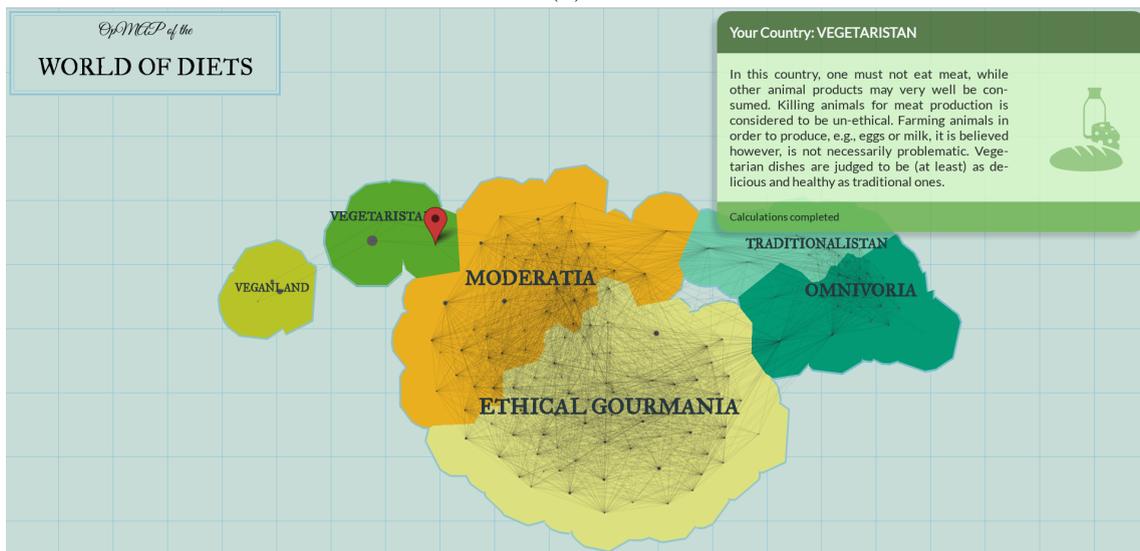
Figure 6.5.: Our installation at Open Codes, ZKM



(a)



(b)



(c)

Figure 6.4.: The dynamic map application. (a) Map showing statistics of the underlying data set. (b) Supersets of a non-sink opinion being highlighted. (c) A sink opinion being localized.

7. Conclusion and Outlook

7.1. Summary

The work presented here integrates several well-studied formal models and algorithms from a wide variety of scientific disciplines—argumentation theory, probabilistic epistemology, network analysis, graph drawing, and computational geometry. We created a working prototype of an interactive dynamic map visualization that is intuitive and easily readable. Thereby we showed that the geography metaphor for visualization of relational information devised by Kobourov et al. in [GHK10] can be applied to visualize opinion landscapes in large public debates.

We first created a formal model of the space of opinions, which are represented as truth-value assignments to statements that are relevant to the debate. We then explored various possibilities to measure semantic distances between opinions—from inexpensive edit distances to Bayesian measures of coherence in concert with probabilistic degrees of justification defined on deductive argumentation. We introduced our opinion-graph where the nodes represent opinions and edges are weighted by the distance measure which turned out to be the most suitable in this context: The degree of mutual coherence.

Employing the debate on dietary habits as a use case, we designed an online survey that elicits users’ stances on the debate’s statements. Thereby, we collected empirical opinion data to build our map from.

For finding structure in the opinion-graph, we investigated different clustering algorithms in combination with vertex and edge filtering methods. Vertices are filtered using domain-specific information from argumentation theory. This is done with the help of an additional structure, the entailment-DAG, which connects opinion vertices according to logical entailment. Only sink nodes of that graph are drawn on the map, whereas other nodes are considered in the form of vertex weights divided among their respective children.

To deal with negative edge weights, our method relies on a linear transformation: Adding a constant and removing all edges whose weights remain negative after that transformation. The constant is determined such that the graph extracted from the initially collected opinion sample remains connected. To aid clustering and to enable a more evenly spaced layout, a local edge filtering is performed. Using Gephi and quantitative statistics about the statements supported and rejected in each cluster, we extensively explored if the clusterings found by various algorithms are semantically

meaningful. The final application uses the infomap algorithm, which reveals interesting patterns in the opinion sample: The majority of participants feel that meat consumption has to be restricted in some way. They are subdivided into two “countries”, both of which are aware of the problems of modern mass farming. One of the countries does not abandon meat consumption altogether, mostly because of the pleasure experience of eating, while the other advocates a drastic reduction of meat or even animal products in general. The meat-eaters are also subdivided into opinions that do not want to restrict meat consumption for reasons of indulgence and those who do not view the advanced effects of meat consumption as problematic. Vegans and vegetarians are each grouped into their own cluster by the algorithm.

To depict the graph while clearly showing the determined cluster structure, the application uses a force-directed graph drawing algorithm with the forces adapted to optimize drawing properties that are in line with the map metaphor: The more similar two opinions, the closer they are. Opinions of the same cluster are positioned in geometrically similar regions, and distances between clusters are not too short.

The actual landmass is generated by binding irregular polygons to the opinion nodes which are clipped using polygons calculated by means of a Voronoi decomposition. The polygons are finally coloured in the respective country colours.

The interactive map allows new opinions to be located as users input their opinion via the survey. The tool we created is relatively general and can be used for any policy debate that can be construed as deductive argumentation. A lot of additional features for exploring the opinion space and various enhancements are conceivable, some of which are addressed in the following section.

7.2. Future Research

Many aspects of our method allow alternative approaches and open up plenty of fascinating research problems in each of the scientific fields involved in creating opinion maps.

Survey Design. Different ways of eliciting and formalizing opinions could be investigated. In our survey design we have strong presuppositions about how the opinion landscape on nutrition behaviour is structured. In particular, we pre-structure our data by first offering a selection of core options. Then, on the next page we exclusively present arguments that might be advanced in favour of the respective option. Therefore, it would be desirable to redesign the survey with the support of sociologists and psychologists, in a manner that all participants would be presented with exactly the same arguments. In order to prevent users from submitting inconsistent positions, one could introduce warnings once inconsistencies arise and ask them to reconsider their choice.

Our survey offers exactly three options for every item, **agree**, **disagree**, and **neutral**. *Likert scales*, proposed by Rensis Likert [Lik32], are another widely used psychometric in questionnaires. Respondents can indicate their level of agreement or disagreement on a symmetric scale, for example **strongly agree** - **agree** - **neutral** - **disagree** - **strongly disagree**. In order to ensure that such a study is psychologically sound, one should perform a pilot study and remove items for which more than 80% of participants indicate the same level of acceptance or rejection etc. (see for example [VTH02]). Consequently, research on which coherence measure is meaningful for opinions formalized in this way would be required. An alternative approach to obtain more accurate opinions would be to stick to the ternary scale but offer users to add a weight to each argument in order to indicate how important it is to them, as it is the case with WAHL-O-MAT.

Application Design. A multitude of tools and features for interactively exploring the map could be implemented. One possibility would be a zooming function which allows

users to inspect different opinions and see which statements are supported. Another idea would be a highlighting function that marks all those opinions that support a certain policy option—in our case a practice such as vegetarianism, veganism, organic diets etc. Opinion maps may even be used for consensus formation, for example by overlaying a heatmap for each policy option, which indicates the extent to which certain areas of the map are compatible or incompatible with the option. Therein, positions that are a good compromise would be unveiled. Hence, the application may be a useful tool in online discussion forums.

Moreover, a desirable feature would be for users to be able to bring up arguments of their own. For instance, our current survey does not consider any religious aspects to eating. This, in turn, would call for an automated method to unveil dialectical structures in the debate, which requires advanced speech comprehension.

Finally, more sophisticated information could be displayed. For example, one could ask the same person at different points in time to see how their opinion changes, and maybe relate these changes to media-effective events, for example as public awareness of certain phenomena, such as the mad cow disease, rises.

Empirical Study. A usability study should be performed, in order to determine the extent to which an opinion map can be used by the intended user population to understand debates and grasp large amounts of opinion data with effectiveness, efficiency and satisfaction. Such a study can take the form of a *think aloud* session with novice users [ES80]. Depending on the provided tools, the user can be asked to perform various statistical tasks, such as to locate him- or herself in the opinion space or summarize the main arguments that are used in a country. While performing the tasks, the user talks about how he or she solves them to the experimenter. The results should then be compared with other modes of presentation, such as text-based or chart-based presentations. Another possibility is a *cognitive walk through* with an expert [WRLP94], either in the field the debate is concerned with or in the fields of deliberation theory and communication science.

Coherence. It might be worthwhile to investigate different ideal measures of coherence and possible heuristics for their approximation. Furthermore, a comprehensive formal theory of mutual coherence is yet to be established, possibly relating the concepts of coherence and entailment using the entailment-DAG. Thereby, graph theoretic properties of the DAG could be related to epistemological properties of the opinion data.

Clustering. In the context of opinion graphs, it would be natural to devise a clustering objective that directly makes use of the logic of the coherence measure. For example, it could be evaluated for each cluster to what degree the union of all statements accepted in positions that form the cluster forms a coherent set. Alternatively, it could be demanded that every pair of opinions within a cluster is coherent to a certain degree. Additionally, a clustering algorithm tailored to coherence graphs could involve *cluster editing*. Herein, the goal is to transform a graph into a disjoint union of cliques, using edge deletions and insertions with minimal cost (see for example [DLL⁺06]).

Clusterings can be tested by means of a *ground-truth* [YL15]. However, in this context it is not entirely clear which opinions should form a cluster. Therefore, investigations on this question are required to generate a ground-truth by hand (similar to our fabricated, highly structured sample 2017-05-13T05-38-29) and then also automatically, once the relevant aspects are known.

Furthermore, one could test alternative clustering paradigms in this context, such as *fuzzy clustering*, where nodes are assigned to clusters with certain probabilities, *overlapping clustering*, where nodes can be assigned to multiple clusters, and *hierarchical clustering*, where the network is divided successively into a series of partitions, from the entire

network to singleton communities. It can easily be seen, how well this last approach lines up with the map metaphor.

Initially, we aimed for an ever-evolving map which would allow new clusters to dynamically emerge. This, however, would lead to a whole new set of challenges: For example, automatic cluster labelling would be required. To solve that problem, one might come up with some simple heuristics that derive a country label from the most frequently accepted statements in a new cluster. However, if one wants to have a humorous label and country description, as we created manually for the exhibition, sophisticated methods from computational linguistics are required. Automatic *topic labeling* is in fact a flourishing branch in that field, for example using *latent dirichlet allocation* [LGNB11].

Visualization. Our current system is highly tuned to the initial dataset. It is not trivial to gain fine-grained control over the force layout, which makes it difficult to guarantee correct country sizes and adjacencies. To overcome these problems, we suggest to experiment with an alternative method that would first generate a map that fulfils all desired properties, and then embed the graph on this map afterwards. A lot of theoretical work has been done on *contact representations* that can be created for planar graphs (for example [CKU13, PR00]). In such a representation, each vertex is represented as a geometrical shape, so that two objects touch if and only if the corresponding vertices are connected by an edge. As countries are k -gons with rather high values of k , it is possible to fix the outer boundary of a country first, to ensure that the country size is representative of the underlying data. It is worth investigating how these theoretical results can be implemented in practice. Regarding the embedding, a modification of the force directed algorithm that allows constraints (for example [DMW08]) could be used. Finally, it would have to be investigated how this alternative implementation works with empirical data, as compared to the current OPMAP implementation.

List of Figures

1.1.	Concept drawing	2
1.2.	Main steps in the mapping algorithm	5
1.3.	Obtaining “landmass”	5
2.1.	Various types of thematic maps	8
2.2.	“Karte des Bücherlandes” by Alphons Woelfle, 1938	9
2.3.	Science map	10
2.4.	Our dataset visualized with GMAP	11
2.5.	Online platform Kialo	11
3.1.	Argument Map of core claims in the Veggie-Debate	14
3.2.	Position tree	16
3.3.	Artificially structured simulated dataset	22
4.1.	Illustration of cut, volume, and internal volume	26
4.2.	Infomap illustration	27
4.3.	Shannon entropy	28
4.4.	Walktrap algorithm	30
4.5.	Per-cluster acceptance/rejection rates in simulated dataset	32
4.6.	Jaccard similarities	35
4.7.	Illustrative entailment-graph	36
4.8.	Opinion-graph in Gephi	37
4.9.	Per-cluster acceptance/rejection rates in the survey data	38
5.1.	A map of the “World of Diets”	39
5.2.	Simulation cooling	45
5.3.	Correlation between edge lengths and edge weights	46
5.4.	Voronoi diagram	47
5.5.	Largest empty circle	48
5.6.	Beach line	49
5.7.	Fortune’s algorithm – site event	49
5.8.	Fortune’s algorithm – circle event	49
5.9.	Binary search tree	50
5.10.	Classic vs. weighted Voronoi diagram	52
6.1.	Overview of the OPMAP architecture	54
6.2.	Interpolating mutual coherence	56
6.3.	Opinion distribution donut chart	64
6.5.	Our installation at Open Codes, ZKM	65
6.4.	The dynamic map application	66

List of Tables

3.1. Description of the simulated datasets	21
4.1. Clustering results for a simulated random dataset	33
4.2. Clustering results for a simulated dataset with constraints	33
4.3. Clustering results for a simulated artificially structured dataset	34
A.2. Survey design	84

Bibliography

- [AA99] Gennady L. Andrienko and Natalia V. Andrienko. Interactive maps for visual data exploration. *International Journal of Geographical Information Science*, 13(4):355–374, 1999.
- [AKV15] Md. Jawaherul Alam, Stephen G. Kobourov, and Sankar Veeramoni. Quantitative measures for cartogram generation techniques. *Comput. Graph. Forum*, 34(3):351–360, 2015.
- [BCB03] Katy Börner, Chaomei Chen, and Kevin W. Boyack. Visualizing knowledge domains. *ARIST*, 37(1):179–255, 2003.
- [Bet12] Gregor Betz. On degrees of justification. *Erkenntnis*, 77(2):237–272, 2012.
- [Bet17] Gregor Betz. Handout on probabilistic epistemology, 2017. Presented at the research seminar of Prof. Beckert, KIT, March 23rd, 2017. Available from the author.
- [BGLL08] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of community hierarchies in large networks. *CoRR*, abs/0803.0476, 2008.
- [BH86] Josh Barnes and Piet Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *nature*, 324(6096):446–449, 1986.
- [BHJ09] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. In *Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, San Jose, California, USA, May 17-20, 2009*, 2009.
- [BKB05] Kevin W. Boyack, Richard Klavans, and Katy Börner. Mapping the backbone of science. *Scientometrics*, 64(3):351–374, 2005.
- [Bos17a] Mike Bostock. D3 example code: Force-directed graph. <https://bl.ocks.org/mbostock/4062045>, 2017. Accessed: 2017-11-27.
- [Bos17b] Mike Bostock. D3-FORCE – documentation. <https://github.com/d3/d3-force>, 2017. Accessed: 2017-12-15.
- [CHvKS10] Sergio Cabello, Herman J. Haverkort, Marc J. van Kreveld, and Bettina Speckmann. Algorithmic aspects of proportional symbol maps. *Algorithmica*, 58(3):543–565, 2010.
- [CKU13] Steven Chaplick, Stephen G. Kobourov, and Torsten Ueckerdt. Equilateral l-contact graphs. In *Graph-Theoretic Concepts in Computer Science - 39th International Workshop, WG 2013, Lübeck, Germany, June 19-21, 2013, Revised Papers*, pages 139–151, 2013.
- [CN06] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695:1–9, 2006.

- [Cou16] Andrew Couch. D3 example code: Better force layout node selection. <http://bl.ocks.org/couchand/6420534>, 2016. Accessed: 2017-11-27.
- [DBCVKO08] Mark De Berg, Otfried Cheong, Marc Van Kreveld, and Mark Overmars. *Computational Geometry: Introduction*. Springer, 2008.
- [DLL⁺06] Frank Dehne, Michael Langston, Xuemei Luo, Sylvain Pitre, Peter Shaw, and Yun Zhang. The cluster editing problem: Implementations and experiments. *Parameterized and Exact Computation*, pages 13–24, 2006.
- [DM07] Igor Douven and Wouter Meijs. Measuring coherence. *Synthese*, 156(3):405–425, 2007.
- [DMW08] Tim Dwyer, Kim Marriott, and Michael Wybrow. Topology preserving constrained graph layout. In *Graph Drawing, 16th International Symposium, GD 2008, Heraklion, Crete, Greece, September 21-24, 2008. Revised Papers*, pages 230–241, 2008.
- [Dor11] Daniel Dorling. Area cartograms: Their use and creation. pages 252–260. Wiley Online Library, 2011.
- [Ead84] PA Eades. A heuristic for graph drawing. *Congressus numerantium*, 42:149–160, 1984.
- [EHKP15] Alon Efrat, Yifan Hu, Stephen G. Kobourov, and Sergey Pupyrev. MapSets: Visualizing Embedded and Clustered Graphs. *J. Graph Algorithms Appl.*, 19(2):571–593, 2015.
- [ES80] K. Anders Ericsson and Herbert A. Simon. Verbal reports as data. *Psychological review*, 87(3):215, 1980.
- [FB07] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.
- [FH16] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *CoRR*, abs/1608.00163, 2016.
- [For87] Steven Fortune. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2:153–174, 1987.
- [fpB] Bundeszentrale für politische Bildung. Wahl-O-Mat. <https://www.bpb.de/politik/wahlen/wahl-o-mat/>. Accessed: 2017-12-15.
- [FR91] Thomas M.J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.
- [GHK10] Emden Gansner, Yifan Hu, and Stephen G. Kobourov. GMap: Visualizing Graphs and Clusters as Maps. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 201–208. IEEE, 2010.
- [GHKV09] Emden Gansner, Yifan Hu, Stephen G. Kobourov, and Chris Volinsky. Putting Recommendations on the Map: Visualizing Clusters and Relations. In *Proceedings of the third ACM conference on Recommender systems*, pages 345–348. ACM, 2009.
- [HSWZ17] Michael Hamann, Ben Strasser, Dorothea Wagner, and Tim Zeitz. Simple distributed graph clustering using modularity and map equation. *CoRR*, abs/1710.09605, 2017.

- [Hu05] Yifan Hu. Efficient, high-quality force-directed graph drawing. *Mathematica Journal*, 10(1):37–71, 2005.
- [JVHB14] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PloS one*, 9(6):e98679, 2014.
- [Kob12] Stephen G. Kobourov. Spring embedders and force directed graph drawing algorithms. *CoRR*, abs/1201.3011, 2012.
- [KW78] Joseph B. Kruskal and Myron Wish. *Multidimensional scaling*, volume 11. Sage, 1978.
- [LGNB11] Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1536–1545. Association for Computational Linguistics, 2011.
- [Lik32] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.
- [Llo82] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Information Theory*, 28(2):129–136, 1982.
- [Nav01] Gonzalo Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88, 2001.
- [NG04] Mark E.J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.
- [Pet15] Keith Peters. Coding math – educational videos that teach the math needed in programming, episodes 36 – 39. <http://www.codingmath.com/>, 2015. Accessed: 2017-12-15.
- [PL05] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *ISCIS*, volume 3733, pages 284–293, 2005.
- [PR00] Tomaz Pisanski and Milan Randic. Bridges between geometry and graph theory. *MAA NOTES*, pages 174–194, 2000.
- [RAB09] Martin Rosvall, Daniel Axelsson, and Carl T. Bergstrom. The map equation. *The European Physical Journal-Special Topics*, 178(1):13–23, 2009.
- [RAK07] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.
- [RB06] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1):016110, 2006.
- [RB08] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.
- [SABW82] William C Swope, Hans C Andersen, Peter H Berens, and Kent R Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics*, 76(1):637–649, 1982.

- [Sha01] Claude E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.
- [Sho99] Tomoji Shogenji. Is coherence truth conducive? *Analysis*, 59(264):338–345, 1999.
- [SPR11] Venu Satuluri, Srinivasan Parthasarathy, and Yiye Ruan. Local graph sparsification for scalable clustering. In Timos K. Sellis, Renée J. Miller, Anastasios Kementsietsidis, and Yannis Velegrakis, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011*, pages 721–732. ACM, 2011.
- [Tal16] William Talbott. Bayesian epistemology. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2016 edition, 2016.
- [TB09] Vincent A. Traag and Jeroen Bruggeman. Community detection in networks with positive and negative links. *Physical Review E*, 80(3):036115, 2009.
- [Ver67] Loup Verlet. Computer “experiments” on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical review*, 159(1):98, 1967.
- [VTH02] Edwin R. Van Teijlingen and Vanora Hundley. The importance of pilot studies. *Nursing Standard*, 16(40):33–36, 2002.
- [WRLP94] Cathleen Wharton, John Rieman, Clayton Lewis, and Peter Polson. The cognitive walkthrough method: A practitioner’s guide. In *Usability inspection methods*, pages 105–140. John Wiley & Sons, Inc., 1994.
- [YL15] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.
- [Zim09] Georg Zimmermann. Karte des Bücherlandes. In *BIS - Das Magazin der Bibliotheken in Sachsen*, pages 184–185, 2009.

Appendix

A. Appendix Section 3

A.1. Veggie-Debate: Sentence Pool

The following table contains the theses and (counter-) arguments featured in the Veggie-Debate, sorted by type (core claim, culinary, health, financial, natural, climate protection, animal protection, world hunger, autonomy or miscellaneous considerations). The indices are the original ones from the sentence pool, that is, the number in the opinion label. Sentences [4],[11],[12],[19] and [40] are not used in the survey.

Core Claims	
[1]	[Meat-OK] There are meat and animal products that you may eat. <i>[thFleischOK] Es gibt Fleisch und Tierprodukte, die man essen darf.</i>
[2]	[Eat-what-you-want] Everybody can eat whatever they like. <i>[thAllesEsser] Man darf Fleisch und tierische Produkte beliebiger Art essen.</i>
[3]	[Organic-meat] You should eat meat and animal products only from sustainable, species-appropriate manufacturers. <i>[thBioFleisch] Man darf Fleisch und tierische Produkte aus artgerechter, ökologischer Tierhaltung essen, aber auch nur solche.</i>
[4]	[No-mass-farming] You should not eat meat products from modern animal breeding and mass farming. <i>[thKeineMassenTierhaltung] Man sollte kein Fleisch aus moderner Tierzucht und Massentierhaltung essen.</i>
[5]	[Strict-veggie] You should not eat meat. <i>[thStriktVeggie] Man sollte kein Fleisch essen.</i>
[6]	[Strict-vegan] You should not eat animal products. <i>[thStriktVegan] Man sollte keine tierischen Produkte essen.</i>
[7]	[Less-meat] You should reduce meat consumption as much as possible. <i>[thWenigerFleisch] Man sollte den Konsum von Fleisch möglichst reduzieren (z.B. nur einmal pro Woche Fleisch essen).</i>
[8]	[Less-animal] You should reduce the consumption of animal products as much as possible. <i>[thWenigerTier] Man sollte den Konsum tierischer Produkte möglichst reduzieren (z.B. Soja- statt Kuhmilch trinken).</i>

Culinary Considerations

- [10] [Culinary-standards] An acceptable diet must also meet culinary standards.
[thKuliStand "Kulinarische Standards"] Eine akzeptable Ernährung muss auch kulinarischen Standards genügen.
- [20] [Cultural-tradition] Completely abandoning meat and/or animal products would put an end to a centuries-old, cultural tradition - the art of cooking.
[argFIKP "Fleischküche ist Kulturerbe"] Der (vollständige) Verzicht auf Fleisch und/oder tierische Produkte würde das Ende einer jahrhundertealten, kulturellen Tradition - der Kochkunst - bedeuten.
- [21] [Pleasure-experience] Completely abandoning meat and/or animal products would lead to the loss of many intense pleasure experiences; the appetite for food, which is a basic human experience of happiness, would be lost; one would merely nourish oneself rather than tasting, indulging and feasting.
[argGAP "Genusserlebnis-Argument"] Der (vollständige) Verzicht auf Fleisch und/oder tierische Produkte bedeutete einen Verlust zahlreicher intensiver Genusserlebnisse; die Lust aufs und am Essen - eine elementare Glückserfahrung des Menschen - ginge verloren; statt zu kosten, zu genießen und zu schlemmen, würde sich nur noch ernährt.
- [22] [Contra-pleasure-experience] From a global point of view, the vegetarian, if not vegan, cuisine has a long tradition, and is in no way inferior to meat cuisine.
[conArgGAP] Die vegetarische, wenn nicht gar vegane Küche hat - global gesehen - eine lange Tradition und steht geschmacklich der Fleisch-Küche in nichts nach.
- [23] [Variety-reduction] Completely abandoning meat and/or animal products reduces the variety of dishes in an unacceptable manner.
[argVielfaltP "Argument aus der Vielfalt"] Der Verzicht auf Fleisch und/oder tierische Produkte reduziert die Vielfalt von Gerichten und Speisen in inakzeptabler Weise.
- [24] [Veggie-cooking-difficult] Many people are simply unable to prepare tasty vegetarian dishes, and - given other professional and private obligations - learning to do so would bring along enormous costs for them.
[argVegSchwP "Vegetarisch kochen schwierig"] Viele Personen können schlicht nicht geschmacklich akzeptabel vegetarisch kochen, und dies zu lernen wäre für sie - angesichts beruflicher und privater Verpflichtungen - mit immensen Kosten verbunden.

Health Considerations

- [11] [Health-risks-unacceptable] Potential deficiency symptoms and health impairments are unacceptable.
[thMangelInakz "Mangelercheinungen-inakzeptabel"] Drohende Mangelercheinungen und gesundheitliche Beeinträchtigung sind inakzeptabel.

- [12] [Unhealthy-fats] Animal fats are harmful to health.
[thSchlFette "Schlechte-Fette"] Tierische Fette sind gesundheitsschädlich.
- [13] [Meat-unhealthy] Excessive meat consumption is unhealthy. To be on the safe side one should consume no meat at all.
[thFlUnges "Fleisch-ungesund"] Zu hoher Fleischkonsum ist gesundheitsschädlich. Sicherheitshalber verzichtet man gleich ganz darauf.
- [14] [Zero-nutritional-risk] Consumption of aliments that are (if consumed excessively) harmful to health should be reduced to zero.
[thNullErRi "Null-Ernährungs-Risiko"] Auf Nahrungsmittel, die (bei übermäßigen Verzehr) gesundheitsschädlich sind, sollte man gänzlich verzichten.
- [15] [Reduce-nutritional-risk] Consumption of aliments that are (if consumed excessively) harmful to health should be reduced as much as possible.
[thErRiRed "Ernährungs-Risiko-reduzieren"] Den Konsum von Nahrungsmitteln, die (bei übermäßigen Verzehr) gesundheitsschädlich sind, sollte man möglichst reduzieren.
- [25] [Physical-fitness] Many people feel physically better (fitter and more energetic) when eating meat regularly.
[argVitalP "Vitalitäts-Argument"] Viele Menschen fühlen sich körperlich besser (vitaler, fitter), wenn sie regelmäßig Fleisch essen.
- [26] [Deficiency-symptoms] Meat is part of a balanced diet. Giving up meat consumption may lead to deficiency symptoms and other health impairments.
[argMangelP "Ernährungsmangel-Einwand"] Fleisch ist Teil einer ausgewogenen Ernährung; ohne Fleischkonsum drohen Mangelerscheinungen und gesundheitliche Beeinträchtigung.
- [27] [Contra-deficiency-symptoms] A diverse vegetarian (vegan) diet is harmless to health (especially when completed with nutritional supplements).
[conArgMangel1] Eine abwechslungsreiche vegetarische (vegane) Ernährung ist gesundheitlich unbedenklich (zumal wenn sie durch Nahrungsergänzungsmittel komplettiert wird).
- [28] [Healthy-cooking-difficult] Many people are simply unable to stick to a healthy vegetarian diet, and - given other professional and private obligations - learning to do so would bring along enormous costs for them.
[argGesSchweP "Gesund kochen schwierig"] Viele Personen können sich schlicht nicht vegetarisch (geschweige denn vegan) gesund ernähren und dies zu lernen wäre für sie - angesichts beruflicher und privater Verpflichtungen - mit immensen Kosten verbunden.
- [29] [No-harmful-fats] Animal fats are unhealthy. Therefore one should not eat animal products at all.
[argKeSchFeP "Keine schlechten Fette"] Tierische Fette sind gesundheitsschädlich. Besser, man verzichtet daher gänzlich auf tierische Produkte.

- [30] [Less-harmful-fats] Animal fats are unhealthy. Therefore one should reduce the consumption of animal products as much as possible.
[argWeSchFeP "Weniger schlechte Fette"] Tierische Fette sind gesundheitsschädlich. Daher sollte man möglichst wenig davon essen.

Financial Considerations

- [9] [Vegetarian-variety] There is a great variety of delicious and reasonably priced vegetarian dishes.
[thVegViel "Vegetarische Vielfalt"] Es gibt viele gute, abwechslungsreiche und zugleich günstige vegetarische Gerichte.
- [31] [Organic-expensive] Many people cannot afford organic products, especially animal products from sustainable agriculture. Organic foods are only for rich people.
[argNuFueReP "Bio nur für Reiche"] Bio-Produkte, insbesondere Tierprodukte aus nachhaltiger Landwirtschaft können sich viele Personen nicht leisten. Bio ist nur was für Reiche.
- [32] [Dietary-change-feasible] An increase in nutritional costs is not unacceptable, as it is reasonable and realistic for households to change their diet (and eat less meat, for instance).
[argErUmZuP "Ernährungsumstellung zumutbar"] Steigende Ernährungskosten sind nicht inakzeptabel, da es zumutbar und realistisch ist, dass Haushalte ihren Speiseplan umstellen (und z.B. weniger Fleisch essen).
- [34] [Veggie-expensive] The ingredients for appetizing vegetarian dishes (for example almond butter, sesame oil, cashew nuts) are expensive, many people can not afford them. If you do not want to live exclusively on potato and carrot soup, veggie food can easily become unaffordable for many.
[argGuVegTeuP "Gutes Veggie-Essen teuer"] Die Zutaten für ansprechende vegetarische Gerichte (z.B. Mandelmus, Sesamöl, Cashewkerne) sind teuer, zahlreiche Personen können sich diese nicht leisten. Wenn man nicht nur Kartoffel-Möhren-Suppe essen will, wird Veggie-Essen schnell für viele unbezahlbar.

Natural Considerations

- [16] [Naturalness-principle] Everything that is natural is good and legitimate, the unnatural is bad and illegitimate.
[thNatuePr "Natürlichkeitsprinzip"] Das, was natürlich ist, ist gut und zulässig, das Unnatürliche ist schlecht und unzulässig.
- [35] [Human-history] Humans have always hunted, farmed and killed animals to exploit and to eat them. Meat consumption can therefore not be strictly reprehensible.
[argMenGeschP "Menschheitsgeschichte"] Menschen haben schon immer Tiere gejagt, gehalten und getötet, um sie zu essen und zu verwerten. Fleischkonsum kann daher nicht strikt verwerflich sein.

- [36] [Natural-food-chains] It is quite natural that animals are hunted and killed - not only humans, but also animals do so. Meat consumption can therefore not be strictly reprehensible.
[argNatNaKeP "Natürliche Nahrungsketten"] Es ist ganz natürlich, dass Tiere gejagt und getötet werden - nicht nur Menschen, auch Tiere tun das. Fleischkonsum kann daher nicht strikt verwerflich sein.
- [37] [Mass-farming-unnatural] Modern livestock breeding and mass farming are unnatural: without precedent in human history and against the nature of the animals.
[argMaTiUnNaP "Massentierhaltung unnatürlich"] Moderne Tierzucht und Massentierhaltung sind unnatürlich: ohne Präzedenz in der Menschheitsgeschichte und wider die Natur der Tiere.

Climate Protection Considerations

- [18] [Ambitious-climate-targets] Ambitious climate targets (for example the two-degree target) have to be satisfied.
[thAmKliZi "Ambitionierte-Klimaziele"] Ambitionierte Klimaziele (z.B. das 2-Grad-Ziel) sollen eingehalten werden.
- [38] [Climate-argument-1] Animal husbandry causes considerable amounts of greenhouse gases directly and indirectly (via land use, power demand, digestive gases). These must be reduced to zero in order to meet ambitious climate goals (for instance the two-degree target).
[argKlimArg1P "Klimaargument-1"] Tierhaltung verursacht direkt und indirekt (über Flächennutzung, Energiebedarf, Verdauungsgase) erhebliche Mengen an Treibhausgasen, die auf Null reduziert werden müssen, um ambitionierte Klimaziele (z.B. das 2-Grad-Ziel) einzuhalten.
- [39] [Climate-argument-2] Animal husbandry causes considerable amounts of greenhouse gases directly and indirectly (via land use, power demand, digestive gases). These must be reduced drastically in order to meet ambitious climate goals (for instance the two-degree target).
[argKlimArg2P "Klimaargument-2"] Tierhaltung verursacht direkt und indirekt (über Flächennutzung, Energiebedarf, Verdauungsgase) erhebliche Mengen an Treibhausgasen, die drastisch reduziert werden müssen, um ambitionierte Klimaziele (z.B. das 2-Grad-Ziel) einzuhalten.
- [40] [Climate-argument-3] = [Climate-argument-2]
- [41] [Nature-destruction] Modern mass farming involves high area and power requirements and goes along with monocultures and overfertilization (manure). Thus, it destroys the landscape as well as valuable natural regions and threatens domestic biodiversity.
[argNaZeP "Naturzerstörung"] Moderne Massentierhaltung hat einen hohen Flächen- und Energiebedarf und geht mit Monokulturen sowie Überdüngung (Gülle) einher. Damit zerstört sie das Landschaftsbild sowie wertvolle Naturräume und bedroht die heimische Artenvielfalt.

Animal Protection Considerations

- [42] [Animal-protection] In modern mass farming, animals endure a painful existence and are brutally and painfully slaughtered. Consumers are jointly responsible for this suffering.
[argTiSchuP "Tierschutz-Argument"] In der modernen Massentierhaltung fristen Tiere ein qualvolles Dasein und werden auf brutale und schmerzvolle Weise getötet. Fleisch-Konsumenten sind für dieses Leid mitverantwortlich.
- [43] [No-pain-experience] Animals do not consciously experience pain in any way comparable to humans.
[conArgTiSchu1a "Kein Schmerzempfinden"] Tiere haben kein bewusstes Schmerzempfinden, das dem menschlichen vergleichbar wäre.
- [44] [Mere-scare-stories] In modern mass farming animals are much better off than many scare stories are supposed to make you believe.
[conArgTiSchu1b "Bloß Schauergeschichten"] In der modernen Massentierhaltung geht es Tieren viel besser, als es viele Schauergeschichten glauben machen wollen.
- [45] [Animal-keepers-responsible] Animal breeders and keepers, but not the consumers, are responsible for the conditions in the barn.
[conArgTiSchu4 "Tierhalter verantwortlich"] Tierzüchter und -halter, nicht aber die Konsumenten, sind für die Zustände im Stall verantwortlich.
- [46] [Right-to-life] Animals have a right to life. Killing them in order to eat them is morally wrong.
[argReAuLeP "Recht auf Leben"] Tiere haben ein Recht auf Leben. Es ist moralisch falsch, sie zu töten, um sie zu essen.
- [47] [Inappropriate-farming] Even in so-called species-appropriate animal farming, which serves only the manufacturing of animal products (such as milk and eggs), animals are confined, limited in their freedom of movement and finally slayed. This is morally wrong.
[argArTiUnP "Artgerechte Tierhaltung Unding"] Selbst in sogenannter artgerechter Tierhaltung, die nur der Herstellung tierischer Produkte (wie Milch und Eier) dient, werden Tiere eingesperrt, in ihrer Bewegungsfreiheit begrenzt und schließlich getötet. Das ist moralisch falsch.
- [48] [Avoid-animal-suffering] As a consumer you can never be quite sure under which conditions the animals live, whose products you want to buy. Thus, it is preferable to not buy and eat animal products at all.
[argTiLeUnVeP "Tierleid unbedingt vermeiden"] Als Konsument kann man nie ganz sicher sein, unter welchen Bedingungen die Tiere leben, deren Produkte man zu kaufen gedenkt. Besser man kauft und isst daher gar keine tierischen Produkte.

World Hunger Considerations

- [19] [Sustainable-nourishment] Nourish a growing world population, which will soon exceed 9 billion people, in a sustainable manner is imperative.
[thWeEr "Weltbevölkerung ernähren"] Es ist zwingend geboten, eine wachsende Weltbevölkerung von bald mehr als 9 Milliarden Menschen auf nachhaltige Weise zu ernähren.
- [49] [World-population-argument-1] A growing world population which will soon exceed 9 billion people can only be nourished in a sustainable way if meat consumption is drastically reduced on a global scale. To do this you should contribute yourself and eat less meat.
[argWeBev1P "Weltbevölkerungs-Argument-1"] Eine wachsende Weltbevölkerung von bald mehr als 9 Milliarden Menschen kann nur auf nachhaltige Weise ernährt werden, wenn der Fleischkonsum global gesehen drastisch reduziert wird. Dazu sollte man selbst beitragen und weniger Fleisch essen.
- [50] [World-population-argument-2] A growing world population which will soon exceed 9 billion people can only be nourished in a sustainable way if meat consumption is drastically reduced on a global scale. It is better to give it up entirely.
[argWeBev2P "Weltbevölkerungs-Argument-2"] Eine wachsende Weltbevölkerung von bald mehr als 9 Milliarden Menschen kann nur auf nachhaltige Weise ernährt werden, wenn der Fleischkonsum global gesehen drastisch reduziert wird. Besser man verzichtet ganz darauf.

Autonomy Considerations

- [33] [Meat-abandonment-infeasible] If everyone is allowed to eat what they want, then the complete abandonment of meat dishes is unacceptable.
[argFlVeUnzP "Fleischverzicht unzumutbar"] Wenn jeder essen darf, was er will, dann ist ein Fleischverzicht unzumutbar.
- [51] [Diet-private-affair] Everyone can decide for themselves what they eat.
[argEsPriSaP "Essen ist Privatsache"] Jeder bestimmt selbst, was er/sie isst.
- [52] [Moralism-reproach] Proponents of vegetarian or vegan diets make morally overstated claims.
[argMoVoP "Moralismus-Vorwurf"] Befürworter vegetarischer oder veganer Ernährung stellen moralisch völlig überzogene Forderungen.
- [53] [No-patronization] Banning certain foods is merely some do-gooders' attempt to patronize other people.
[argGeBevoP "Gegen Bevormundung"] Essverbote sind Bevormundungsversuche von Gutmenschen, sonst nichts.

Miscellaneous Considerations

[17] [Individuals-negligible] Since the individual does not alter existing production methods by giving up certain products no changes of consumer behaviour are required.

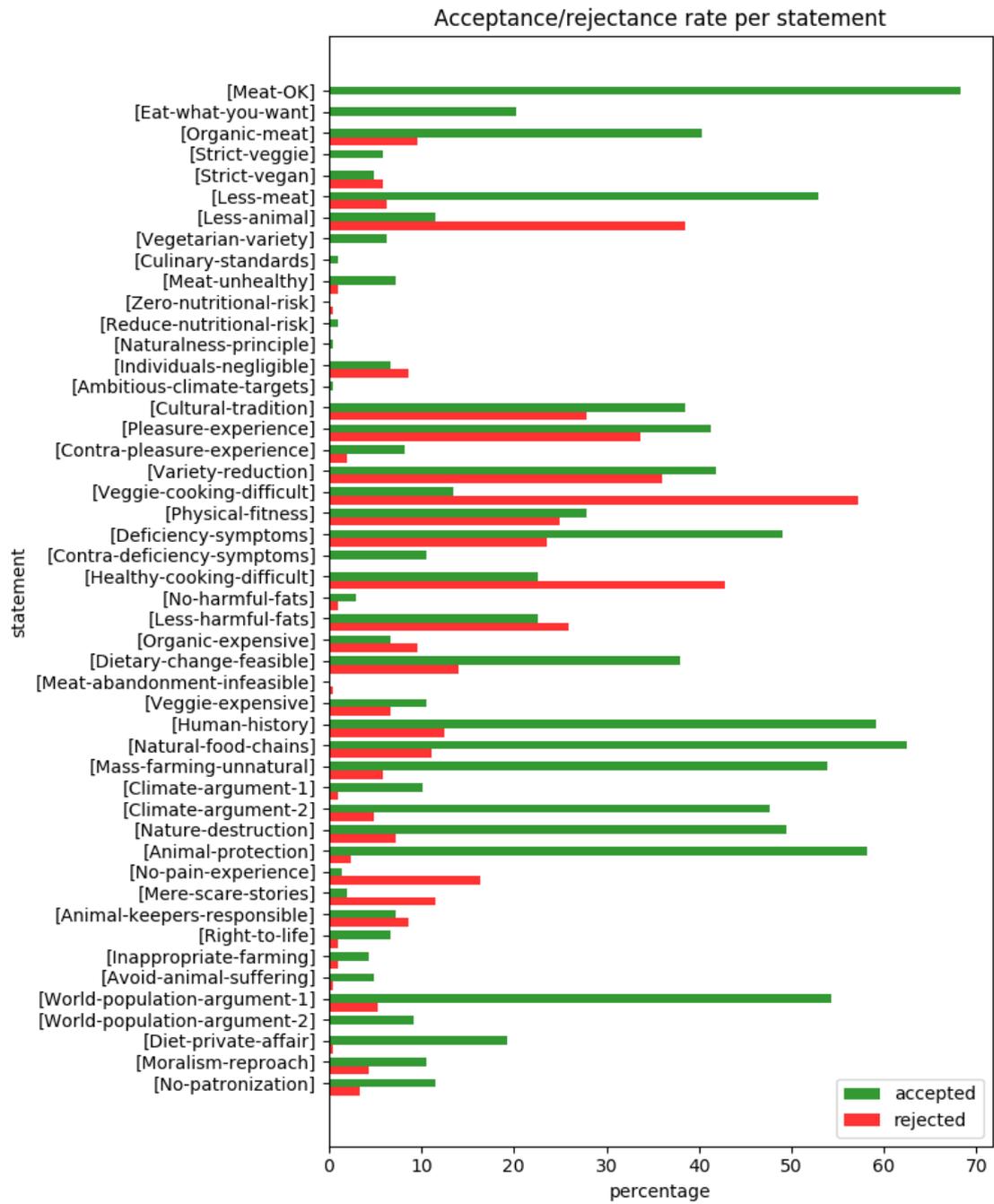
[thEinzUner "Einzelne-Unerheblich-Einwand"] Durch Produktverzicht ändert der Einzelne nichts an bestehenden Erzeugungsweisen. Und nur dann wären Handlungsänderungen geboten.

A.2. Survey Design

	vegan	vegetarian	restricted	omnivore	catch-all
Core	(6)	(5), (¬ 6)	(1), 3, 7, 8	(2)	
Culinary	21	22	20, 21, 23, 24	20, 21, 23, 24	10, 20, 21, 22, 23, 24
Health	13, 29, 27	13, 27	25, 26, 28, 30	25, 26, 28	13, 14, 15, 25, 26, 2, 28, 29, 30
Finance		9	32	31, 34	9, 31, 32, 34
Naturalness			37, 35, 36	35, 36	16, 35, 36, 37
Climate & environment	38	38	39, 41		18, 38, 39, 41
Animal rights	47 48, 46	46	42	43, 44, 45	42, 43, 44, 45, 46, 47, 48
World hunger	50	50	49		49, 50
Autonomy				51, 52, 53	33, 51, 52, 53
Miscellaneous				17	17

Table A.2.: Survey design. Each column is a survey option, the rows indicate the type of consideration. The numbers correspond to the sentences given in the sentence pool in Appendix Section A. Numbers in parentheses indicate (negated) sentences, that are not actually offered in the survey but are implied by the selected initial option.

A.3. Survey Results



The following table shows the top five most frequently accepted and rejected sentences in the collected opinion sample.

Top five most frequently accepted statements		
[1]	[Meat-OK] There are meat and animal products that you may eat.	(68.27%)
[36]	[Natural-food-chains] It is quite natural that animals are hunted and killed - not only humans, but also animals do so. Meat consumption can therefore not be strictly reprehensible.	(62.5%)
[35]	[Human-history] Humans have always hunted, farmed and killed animals to exploit and to eat them. Meat consumption can therefore not be strictly reprehensible.	(59.13%)
[42]	[Animal-protection] In modern mass farming, animals endure a painful existence and are brutally and painfully slaughtered. Consumers are jointly responsible for this suffering.	(58.17%)
[49]	[World-population-argument-1] A growing world population which will soon exceed 9 billion people can only be nourished in a sustainable way if meat consumption is drastically reduced on a global scale. It is better to give it up entirely.	(54.33%)
Top five most frequently rejected statements		
[24]	[Veggie-cooking-difficult] Many people are simply unable to prepare tasty vegetarian dishes, and - given other professional and private obligations - learning to do so would bring along enormous costs for them.	(57.21%)
[28]	[Healthy-cooking-difficult] Many people are simply unable to stick to a healthy vegetarian diet, and - given other professional and private obligations - learning to do so would bring along enormous costs for them.	(42.79%)
[8]	[Less-animal] You should reduce the consumption of animal products as much as possible.	(38.46%)
[23]	[Variety-reduction] Completely abandoning meat and/or animal products reduces the variety of dishes in an unacceptable manner.	(36.06%)
[21]	[Pleasure-experience] Completely abandoning meat and/or animal products would lead to the loss of many intense pleasure experiences; the appetite for food, which is a basic human experience of happiness, would be lost; one would merely nourish oneself rather than tasting, indulging and feasting.	(33.65%)

B. Appendix Section 4

B.1. Country Descriptions

When a user enters their opinion a description of the resulting country is displayed. The corresponding message boxes are shown below. The texts were formulated by Gregor Betz.

Your Country: ETHICAL GOURMANIA

In this country, eating meat is in principle deemed permissible for reasons of taste and cultural tradition. However, the problems of modern mass farming are also acknowledged, which explains the wide-shared belief that only meat from sustainable and organic production may be consumed.



Calculations completed

(a)

Your Country: Once-A-Week-Land (aka MODERATIA)

In this country, there exist a profound awareness and recognition of the diverse problems of mass farming – ranging from violations of animal rights in the farm to global climate change. A (possibly drastic) reduction of meat consumption and a boycott of products from mass farming are considered to be appropriate answers to these problems. Many even demand that the consumption of all animal products whatsoever be reduced.



Calculations completed

(b)

Your Country: TRADITIONALISTAN

In this country, killing and eating animals is seen as something very natural. The traditional human diet is essentially viewed as unproblematic. Any moral demands to change individual eating habits are rejected as paternalistic and out-of-place.



Calculations completed

(c)

Your Country: All-You-Can-(M)eat (aka OMNIVORIA)

In this country, meat is considered to be way too delicious to be abstained from. Animals are not recognized as beings with a right to life. And modern mass farming, it is believed, doesn't represent much of a problem anyway. So, in this country, you may basically eat what you want.



Calculations completed

(d)

Your Country: VEGANLAND

In this country, animal products of any kind are a no-go. In view of all the problems of farming animals and the consumer's irreducible uncertainty about what is really going on in a farm, a vegan diet is unanimously demanded. From a culinary point of view, the vegan cuisine is considered as no less delicious than the traditional one.



Calculations completed

(e)

Your Country: VEGETARISTAN

In this country, one must not eat meat, while other animal products may very well be consumed. Killing animals for meat production is considered to be un-ethical. Farming animals in order to produce, e.g., eggs or milk, it is believed however, is not necessarily problematic. Vegetarian dishes are judged to be (at least) as delicious and healthy as traditional ones.

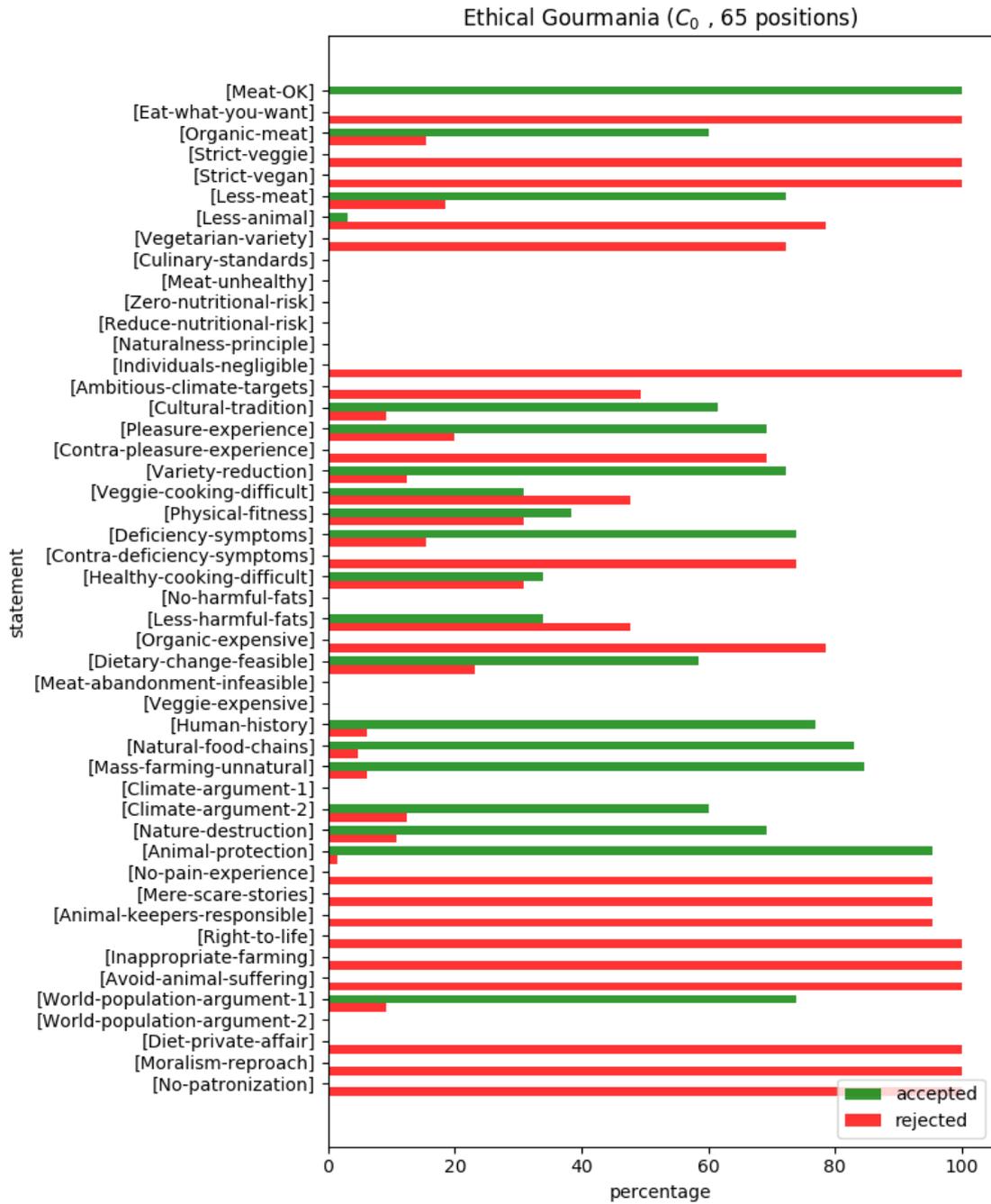


Calculations completed

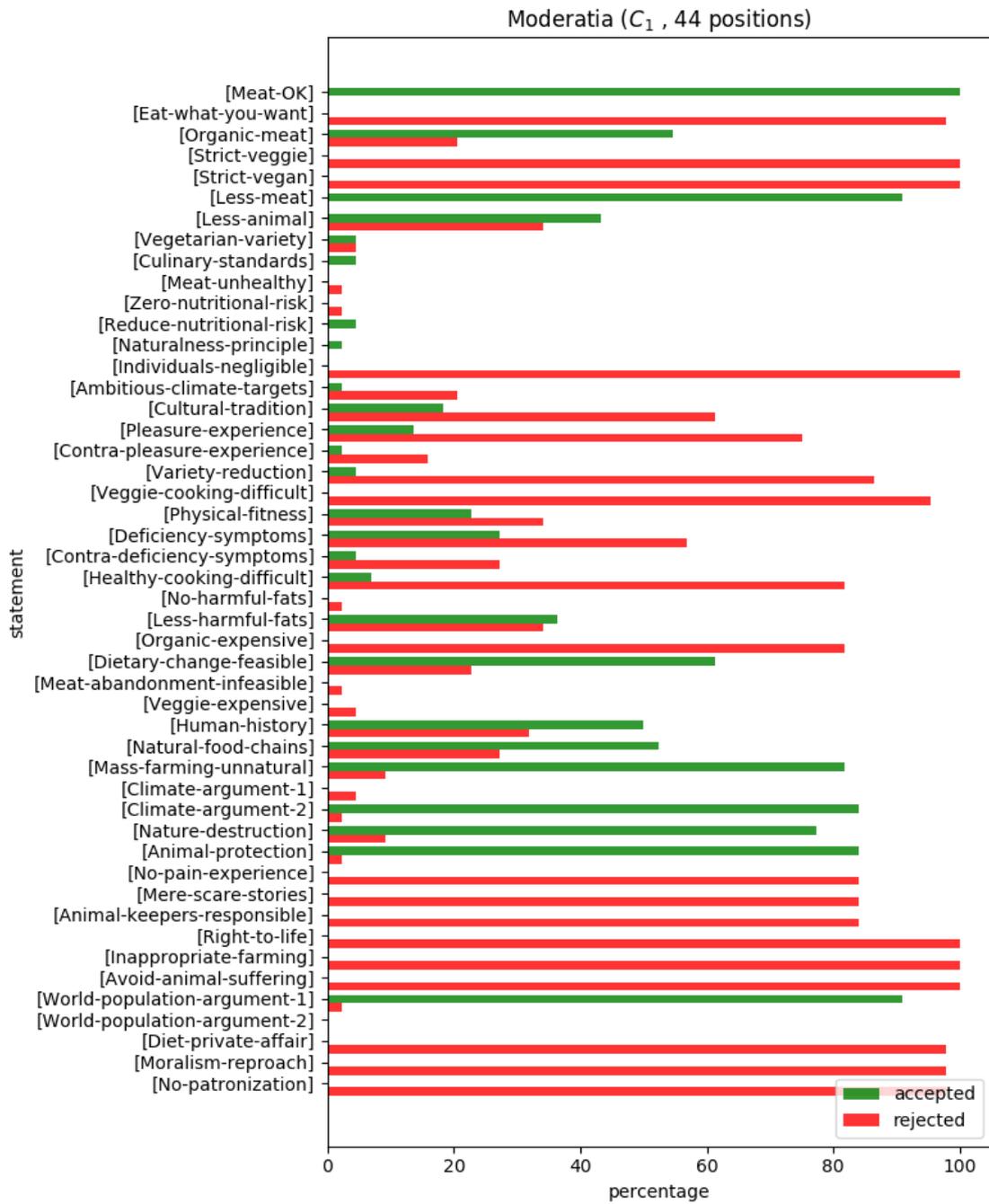
(f)

B.2. Clustering Results

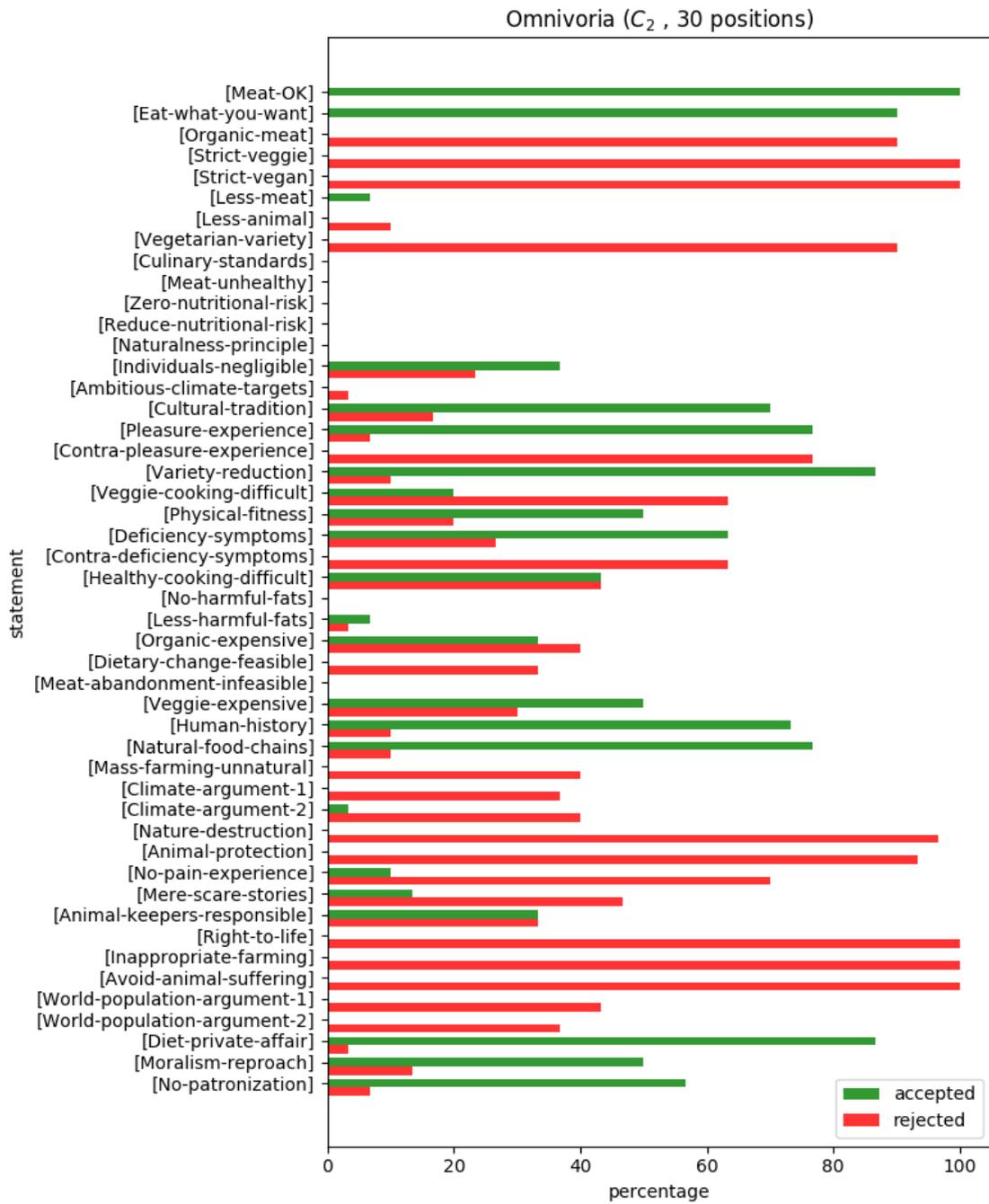
The following figures show the acceptance and rejection rates per cluster. The tables list for each cluster the top five most frequently accepted and rejected statements.



Ethical Gourmania (C_0 , 65 positions)		
Top five most frequently supported statements		
[1]	[Meat-OK] There are meat and animal products that you may eat.	(100.0%)
[42]	[Animal-protection] In modern mass farming, animals endure a painful existence and are brutally and painfully slaughtered. Consumers are jointly responsible for this suffering.	(95.38%)
[37]	[Mass-farming-unnatural] Modern livestock breeding and mass farming are unnatural: without precedent in human history and against the nature of the animals.	(84.62%)
[36]	[Natural-food-chains] It is quite natural that animals are hunted and killed - not only humans, but also animals do so. Meat consumption can therefore not be strictly reprehensible.	(83.08%)
[35]	[Human-history] Humans have always hunted, farmed and killed animals to exploit and to eat them. Meat consumption can therefore not be strictly reprehensible.	(76.92%)
Top five most frequently rejected statements		
[2]	[Eat-what-you-want] Everybody can eat whatever they like.	(100.0%)
[5]	[Strict-veggie] You should not eat meat.	(100.0%)
[6]	[Strict-vegan] You should not eat animal products.	(100.0%)
[17]	[Individuals-negligible] Since the individual does not alter existing production methods by giving up certain products no changes of consumer behavior are required.	(100.0%)
[46]	[Right-to-life] Animals have a right to life. Killing them in order to eat them is morally wrong.	(100.0%)



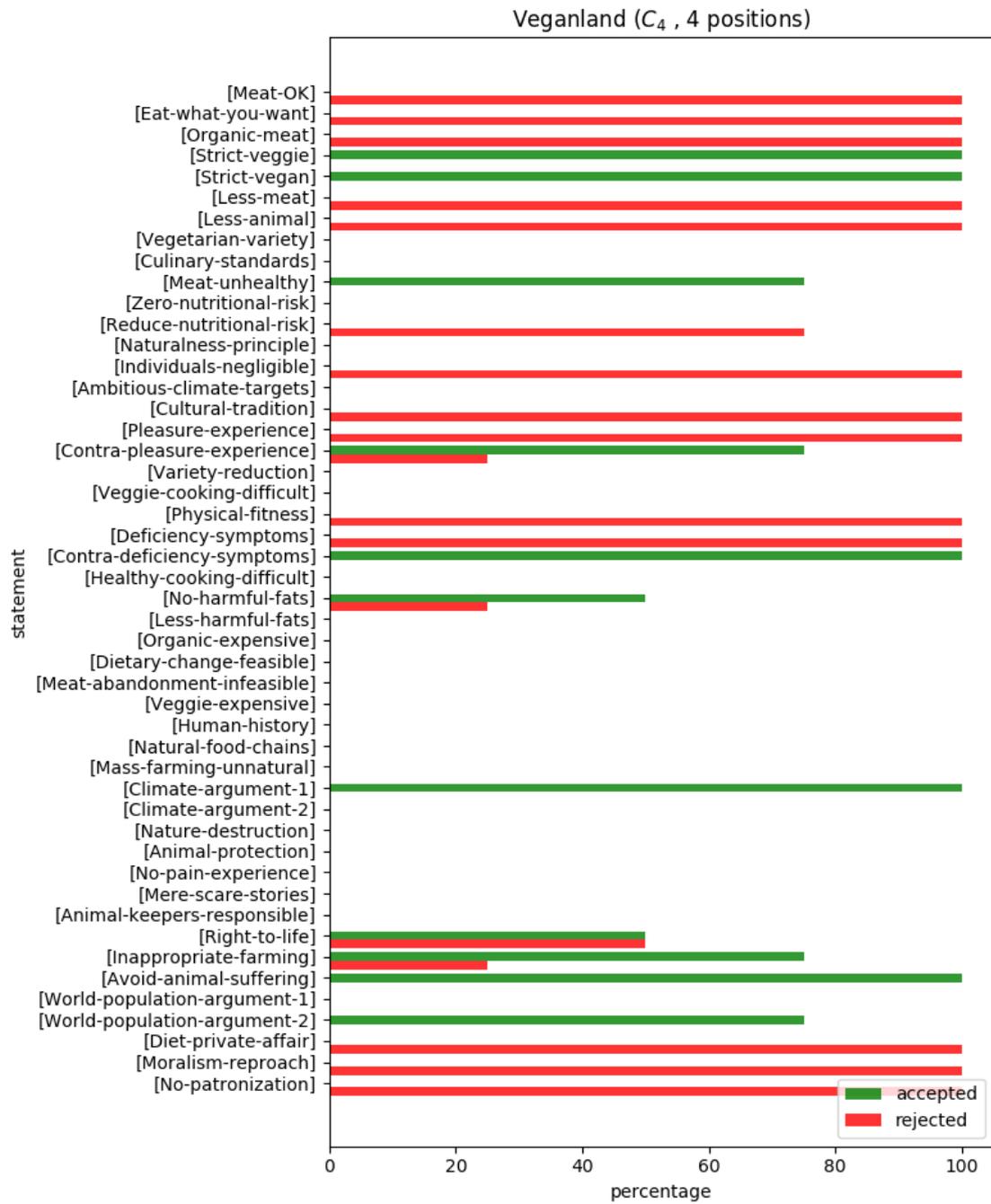
Moderatia (C_1 , 44 positions)		
Top five most frequently supported statements		
[1]	[Meat-OK] There are meat and animal products that you may eat.	(100.0%)
[7]	[Less-meat] You should reduce meat consumption as much as possible.	(90.91%)
[49]	[World-population-argument-1] A growing world population which will soon exceed 9 billion people can only be nourished in a sustainable way if meat consumption is drastically reduced on a global scale. It is better to give it up entirely.	(90.91%)
[39]	[Climate-argument-2] Animal husbandry causes considerable amounts of greenhouse gases directly and indirectly (via land use, power demand, digestive gases). These must be reduced drastically in order to meet ambitious climate goals (for instance the two-degree target).	(84.09%)
[42]	[Animal-protection] In modern mass farming, animals endure a painful existence and are brutally and painfully slaughtered. Consumers are jointly responsible for this suffering.	(84.09%)
Top five most frequently rejected statements		
[5]	[Strict-veggie] You should not eat meat.	(100.0%)
[6]	[Strict-vegan] You should not eat animal products.	(100.0%)
[17]	[Individuals-negligible] Since the individual does not alter existing production methods by giving up certain products no changes of consumer behavior are required.	(100.0%)
[46]	[Right-to-life] Animals have a right to life. Killing them in order to eat them is morally wrong.	(100.0%)
[47]	[Inappropriate-farming] Even in so-called species-appropriate animal farming, which serves only the manufacturing of animal products (such as milk and eggs), animals are confined, limited in their freedom of movement and finally slayed. This is morally wrong.	(100.0%)
[48]	[Avoid-animal-suffering] As a consumer you can never be quite sure under which conditions the animals live, whose products you want to buy. Thus, it is preferable to not buy and eat animal products at all.	(100.0%)



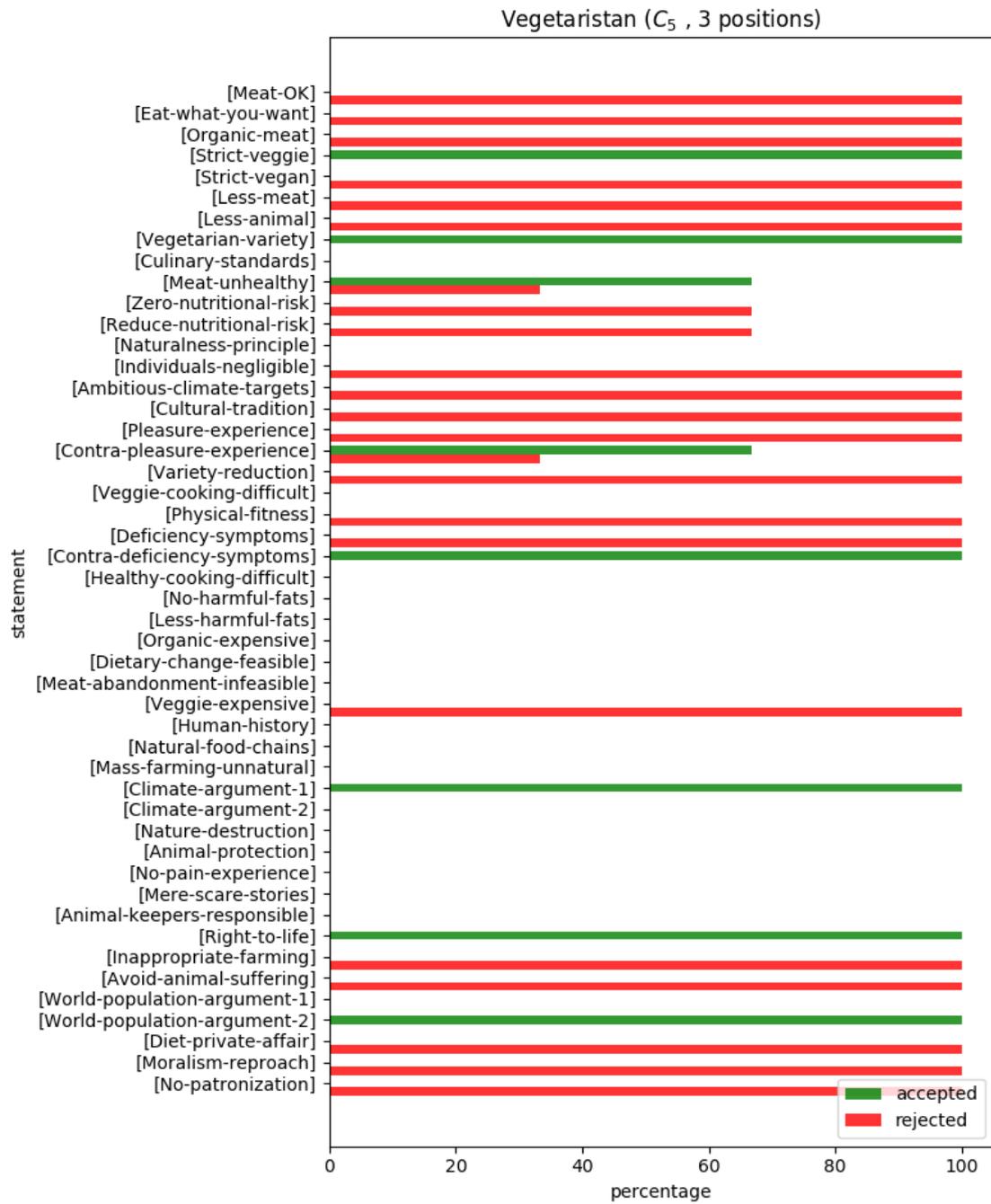
Omnivoria (C_2 , 30 positions)		
Top five most frequently supported statements		
[1]	[Meat-OK] There are meat and animal products that you may eat.	(100.0%)
[2]	[Eat-what-you-want] Everybody can eat whatever they like.	(90.0%)
[23]	[Variety-reduction] Completely abandoning meat and/or animal products reduces the variety of dishes in an unacceptable manner.	(86.67%)
[51]	[Diet-private-affair] Everyone can decide for themselves what they eat.	(86.67%)
[21]	[Pleasure-experience] Completely abandoning meat and/or animal products would lead to the loss of many intense pleasure experiences; the appetite for food, which is a basic human experience of happiness, would be lost; one would merely nourish oneself rather than tasting, indulging and feasting.	(76.67%)
Top five most frequently rejected statements		
[5]	[Strict-veggie] You should not eat meat.	(100.0%)
[6]	[Strict-vegan] You should not eat animal products.	(100.0%)
[46]	[Right-to-life] Animals have a right to life. Killing them in order to eat them is morally wrong.	(100.0%)
[47]	[Inappropriate-farming] Even in so-called species-appropriate animal farming, which serves only the manufacturing of animal products (such as milk and eggs), animals are confined, limited in their freedom of movement and finally slayed. This is morally wrong.	(100.0%)
[48]	[Avoid-animal-suffering] As a consumer you can never be quite sure under which conditions the animals live, whose products you want to buy. Thus, it is preferable to not buy and eat animal products at all.	(100.0%)



Traditionalistan (C_3 , 15 positions)		
Top five most frequently supported statements		
[1]	[Meat-OK] There are meat and animal products that you may eat.	(100.0%)
[36]	[Natural-food-chains] It is quite natural that animals are hunted and killed - not only humans, but also animals do so. Meat consumption can therefore not be strictly reprehensible.	(86.67%)
[35]	[Human-history] Humans have always hunted, farmed and killed animals to exploit and to eat them. Meat consumption can therefore not be strictly reprehensible.	(80.0%)
[2]	[Eat-what-you-want] Everybody can eat whatever they like.	(73.33%)
[26]	[Deficiency-symptoms] Meat is part of a balanced diet. Giving up meat consumption may lead to deficiency symptoms and other health impairments.	(66.67%)
Top five most frequently rejected statements		
[5]	[Strict-veggie] You should not eat meat.	(100.0%)
[6]	[Strict-vegan] You should not eat animal products.	(100.0%)
[46]	[Right-to-life] Animals have a right to life. Killing them in order to eat them is morally wrong.	(100.0%)
[47]	[Inappropriate-farming] Even in so-called species-appropriate animal farming, which serves only the manufacturing of animal products (such as milk and eggs), animals are confined, limited in their freedom of movement and finally slayed. This is morally wrong.	(100.0%)
[48]	[Avoid-animal-suffering] As a consumer you can never be quite sure under which conditions the animals live, whose products you want to buy. Thus, it is preferable to not buy and eat animal products at all.	(100.0%)



Veganland (C_4 , 4 positions)		
Top five most frequently supported statements		
[5]	[Strict-veggie] You should not eat meat.	(100.0%)
[6]	[Strict-vegan] You should not eat animal products.	(100.0%)
[27]	[Contra-deficiency-symptoms] A diverse vegetarian (vegan) diet is harmless to health (especially when completed with nutritional supplements).	(100.0%)
[38]	[Climate-argument-1] Animal husbandry causes considerable amounts of greenhouse gases directly and indirectly (via land use, power demand, digestive gases). These must be reduced to zero in order to meet ambitious climate goals (for instance the two-degree target).	(100.0%)
[48]	[Avoid-animal-suffering] As a consumer you can never be quite sure under which conditions the animals live, whose products you want to buy. Thus, it is preferable to not buy and eat animal products at all.	(100.0%)
Statements rejected by more than 50		%
[1]	[Meat-OK] There are meat and animal products that you may eat.	(100.0%)
[2]	[Eat-what-you-want] Everybody can eat whatever they like.	(100.0%)
[3]	[Organic-meat] You should eat meat and animal products only from sustainable, species-appropriate manufacturers.	(100.0%)
[7]	[Less-meat] You should reduce meat consumption as much as possible.	(100.0%)
[8]	[Less-animal] You should reduce the consumption of animal products as much as possible.	(100.0%)



Vegetaristan (C_5 , 3 positions)		
Top five most frequently supported statements		
[5]	[Strict-veggie] You should not eat meat.	(100.0%)
[9]	[Vegetarian-variety] There is a great variety of delicious and reasonably priced vegetarian dishes.	(100.0%)
[27]	[Contra-deficiency-symptoms] A diverse vegetarian (vegan) diet is harmless to health (especially when completed with nutritional supplements).	(100.0%)
[38]	[Climate-argument-1] Animal husbandry causes considerable amounts of greenhouse gases directly and indirectly (via land use, power demand, digestive gases). These must be reduced to zero in order to meet ambitious climate goals (for instance the two-degree target).	(100.0%)
[46]	[Right-to-life] Animals have a right to life. Killing them in order to eat them is morally wrong.	(100.0%)
Top five most frequently rejected statements		
[1]	[Meat-OK] There are meat and animal products that you may eat.	(100.0%)
[2]	[Eat-what-you-want] Everybody can eat whatever they like.	(100.0%)
[3]	[Organic-meat] You should eat meat and animal products only from sustainable, species-appropriate manufacturers.	(100.0%)
[6]	[Strict-vegan] You should not eat animal products.	(100.0%)
[7]	[Less-meat] You should reduce meat consumption as much as possible.	(100.0%)
