

Chapter 4

Bias by Default?



A Means for A Priori Interface Measurement

Joseph A. Cottam and Leslie M. Blaha

4.1 Introduction

Does all data in an application have an equal chance of being seen? The answer to this question is likely “no”, and that is not necessarily a bad thing. We deliberately influence what is visible and what is not based on many goals or intentions with an information representation. In fact, we rely on such imbalances as part of the data exploration process to keep the information content tractable for human memory and reasoning [16]. Any time something “just pops out” or is “obvious” in a display, there is an element of bias at play. However, does the interface naturally bias in a way that supports or impedes the tasks it was designed to support? How much of that bias is inherent in the interface, and how much is the result of the ways the interface interacts with a specific dataset? How much is the result of the user crafting the interface for personal needs and interests? This chapter proposes Markov modeling as an approach to begin teasing apart the sources of bias in visual analytic systems.

Friedman and colleagues defined bias in computers systems as a slant which produces systematic and unfair discrimination against certain individuals or groups, particularly when that discrimination is paired with unfair outcomes [9, 10]. They defined three types of biases: pre-existing, technical and emergent. Although we disagree that bias only produces unfair outcomes, we find these classes useful for thinking about the bias that can estimated about the system with and without user interactions. Pre-existing bias reflects how a system embodies cultural norms, practices and attitudes that exist in the environment in which the system was developed, programmed or deployed. Pre-existing biases in visual analytics might reflect the culture of the company or research group that developed the system. They could be

J. A. Cottam (✉) · L. M. Blaha
Pacific Northwest National Laboratory, 902 Battelle Blvd, Richland, WA 99354, USA
e-mail: joseph.cottam@pnnl.gov

L. M. Blaha
e-mail: leslie.blaha@pnnl.gov

as simple as the default interface elements, like default color schemes or variable placements on axes. For example, the www.SmartMoney.com *Map of the Market* was a popular example of a treemap compactly representing stock market values [20]. The default color scheme for the map was on a red-green spectrum, with green representing positive trending stocks (gains) and red representing negative trending stocks (losses). The red-green spectrum plays off the cultural norm of green for go and red for stop, adopted from traffic signals. The treemap offered an alternative yellow-blue color map, particularly as an alternative for people with red-green color blindness; however, we have no compelling a priori cultural association for whether yellow or blue should be assigned to gains or losses. Without the pre-existing bias, we lose some intuition for interpreting the visualization.

Other system biases are technical biases. These arise from technical constraints or considerations in the design process, such as choice of hardware or peripherals, which shape the capabilities of the system. Technical bias in visual analytic systems can influence the initial layout, the available algorithms or the options for interaction techniques. Interaction options have implications for the amount of information that needs to be available on the screen. For example, hover and roll-over functions may not be enabled without a mouse or touchpad. Without a hover option, tooltips may not be possible, so information that might have been available on demand may need to be readily available in other ways or on the screen at all times. Or the burden can be placed on the user to query for the information; however, if the user is inexperienced with the system or poor at formulating queries, then some information may not be queried and so may not be seen. Another form of technical bias can be seen in the specific algorithms provided in a tool. They are often chosen based on expected performance on reference hardware for anticipated datasets. As hardware advances, previously intractable algorithms can be implemented, and as new datasets are approached with a tool, different algorithms may be preferred.

A third class of system biases are emergent biases that result from the interactions of users with the system. These are very much of interest to visual analytic systems which are meant to facilitate extensive interactions for data exploration [3]. However, we suspect that emergent biases can only be measured from user interactions with the system. This is because each user has unique biases from attitudes, experience and task goals that will shape the emergent biases [17, 18]. Whether the goal of measurement is online or post hoc bias assessment, it is hard to predict emergent biases in the absence of specific user characteristics and interaction behavior data.

Thus, the goal of this chapter is to propose a framework by which we can measure the biases of an interface from the design of the system, including choices of visualizations and interactions. This may include elements of both technical and pre-existing biases, which do not require the collection of user interaction data for assessment. Of particular interest, at present, is predicting if the system design will steer users into system states where information is systematically unavailable or hard to recover, which will bias their exploratory reasoning and inference processes. Identifying the biases a priori helps (1) identify when and which biases are important, (2) compensate for biases when they hinder task performance and (3) constructively employ biases when they help.

The chapter is structured as follows. First, we review how our concept for a priori bias compares to analytic provenance and interaction sequences. Then we overview the Gapminder tools for the Gapminder World data,¹ which we use as a running example to demonstrate measuring a priori bias. Then we introduce Markov modeling for capturing interfaces as a state space model and introduce two models for a priori system bias. Finally, we present an analysis of Gapminder visualizations. We end with discussions of other potential formalisms and next steps in using this approach to capture biases in analytic tools.

4.2 Relationship to Analytic Provenance

Modeling a priori system bias provides an important complement to analytic provenance modeling. The goal of provenance modeling is to leverage the sequence of user actions to characterize a user's analytic process [13, 21]. Xu et al. [21] argue that there are two important uses for analytic provenance: users can plan further analyses and systems can suggest related but unexamined data. If captured and interpreted automatically, rather than through intensive manual annotation, a mixed-initiative system could incorporate analytic provenance into intelligent recommendations, as illustrated by Endert et al. [6] and Cook et al. [2]. Notably, Dabek and Caban [4] use captured actions to automatically build Markov-model-like automata that form the basis of their intelligent recommender system.

Additionally, when used post hoc, provenance enables analysts to study their own and others' processes. Toward this end, there have been efforts to develop visualizations for showing analytic provenance. GraphTrail [5] uses a graph visualization approach where the states of the analytic system are nodes, and the links illustrate the analyst's transition path between the visualizations. Those links could be enriched by identifying the types of actions they represent in the analytic process, using the catalog of activity developed by Gotz and Zhou [13], for example.

From a system design perspective, analytic provenance analysis allows designers to inspect how design choices and interface elements were used throughout task completion. Our proposed Markov chain model for interface and exploration biases offers a predictive analysis for what might happen. This analysis can be conducted before the system is given to users; it can be engaged early and often in the design process. Importantly, our proposed interface and exploration bias computations are common across users, because they are about the system structure, not the specific user interactions or tasks. Thus, the emergent system biases introduced by the user interactions with the system may be teased apart from the other system biases by leveraging a combination of Markov-chain-based interface analyses and analytic provenance modeling.

Analytic provenance can then capture what a user *actually* does with a system, which can be compared to the predicted provenance from the Markov chain. We

¹<http://gapminder.org>.

propose that modeling the system independent of user interactions is also valuable. Such modeling targets the potential biases in the system that would influence the ways a user *could* or *should* use the system. In many ways, this may be considered task-independent modeling of the potential interaction sequences. Yet, from the perspective of pre-existing bias, this process is also capturing the way system structure and readily available interactions contribute to all tasks attempted with the system. Technical or pre-existing biases may create some systems states that are not useful or would strongly sway the analytic process. While we can observe if or when analysts navigate into those states using analytic provenance, a priori modeling may help us to predict or prevent states unhelpful to the sensemaking process, or that might be compounded by user biases to create strong emergent biases.

4.3 Gapminder

Throughout the rest of this chapter, we will use the Gapminder tools as example visual analytic interfaces. Gapminder is a Swedish organization that curates data and statistics about the world, made available for research and education purposes on <http://www.gapminder.org>. The Gapminder World data includes variables like the population size, income per capita and life expectancy. The organization offers a set of web browser-based interactive visualizations for exploring the Gapminder World data. Figure 4.1 shows an example of the Maps visualization, which has data points plotted as color circles overlaid on the map of the world, one circle per country. In this view, the data are taken from the year 2015, with color indicating world region and the size of the circles representing Income per Person. Possible interactions in this system include changing the variables and settings, selecting countries either by clicking on the circles or on the country name list and watching the data over time through playback controls.

4.4 Markov Models

We propose that Markov chains can be used to model user interfaces and reveal potential biases in those interfaces. That is, we can model interface changes as a probabilistic sequence through a system's state space. We focus on the visual states that can be observed, leaving aside state changes that are only based on hidden internal representation changes.

A general *Markov model* is a statistical process that can be represented as a sequence of states and transition probabilities between those states (i.e., a state machine). Formally, let S_i for $i = 1, \dots, n$ be a set of n possible states, and we define $P(S_i|S_j) = p_{ji}$ as the transition probability from state S_j to state S_i . A sequence of states may be thought of formally as $\{S_i, S_j, S_k, S_i, \dots\}$, where a repeated state, like S_i represents re-visiting a state. All Markov models adhere to the *Markov property*,



Fig. 4.1 Gapminder Maps interface showing the initial state of the interface with 2015 data. See text for more details (Images from <https://www.gapminder.org>, CC-BY license)

which means transitions only depend on the current state (also called being “memoryless”). We represent this as the state of the system at time t being only a function of the state at time $t - 1$, $P(S_t|\{S_1, S_2, \dots, S_{t-1}\}) = P(S_t|S_{t-1})$.

The state machine model is the basis for other Markov processes. For example, a Markov chain is a path through a Markov model [15]. Hidden Markov models are Markov models that maximize the probability of observed chains when the underlying state space and probabilities are not known [1]. Markov models are “simple” in that they are amenable to many different kinds of analyses that yield useful information. Therefore, building a Markov model that faithfully reproduces system behaviors can lead to useful insights about expected behaviors under other circumstances.

The sequence of states in a Markov chain can represent a sequence of states the visual interface can go through. Those state changes maybe driven by direct user actions, streaming data updates or mixed-initiative analysis as it makes recommendations. The complete set of states in the Markov model is comprised of the union of all valid chains. This concept is illustrated in the Gapminder Bubbles visualization in Fig. 4.2. The three screenshots show a progression of states in the system. Figure 4.2(top) shows the initial view of the data when the year 2015 is selected. Figure 4.2(middle) shows the interface after the country India is selected by clicking on the India circle. Figure 4.2(bottom) shows the interface after the circle for Switzerland has been hovered over with the mouse. We note that a display changes can result from two types of changes. The first is a change in content/data produced by replaying the data over time with the playback controls. The second is a change in

the layout or design parameters, resulting from a reconfiguration of the visualization through the right-side panel. To supply the transition probabilities, and thereby complete the Markov model, we assume that possible states of the interface are states in the Markov model and transition probabilities are derived from the screen presence of interface elements. A user session is a Markov chain, drawn from the probability space defined by the model. Analyzing the Markov model state machine provides insight into possible and probable user session patterns.

We can gain insight about potential system biases, pre-existing and technical, by examining the structure of connections between and understanding the relative likelihoods of interface states. For example, some states may not be reachable without a specific sequence of user actions, making them less likely to occur. Other states may have likelihoods that change over time because of certain design or algorithm choices. Still, others may be dependent on the default settings (the initial conditions) of the system. Modeling the user interface independent of actual user actions provides a basis for comparing interfaces to each other. Additionally, examining user interface actions in light of interface bias can tell you if observed biases came from the tool or from the operator. It allows us to distinguish the potential technical and pre-existing biases from the emergent biases in interactive visual analytic systems.

4.5 Interface Models

There are at least two conditions to interface modeling: with and without data loaded. With a dataset loaded, we propose to construct the Markov model with three key features: (1) each link is a possible action; (2) each node is an interface state that results from an action; and (3) links are weighted proportional to the target area on the screen. The above procedure captures the essential idea, but it probably needs to be tempered in some cases. Figure 4.3 illustrates some of the network shapes that result from applying this process by hand to parts of the Gapminder “Bubbles” interface. Linear dependencies are evident, showing that moving large distances in time incurs many step costs, biasing the user to make comparisons in near neighborhoods.

Applying the same procedure to the Gapminder “Map” interface (Fig. 4.1) yields similar patterns BUT with different weights. For example, in the “Bubbles” interface it is possible to directly select the Switzerland bubble, as in Fig. 4.2(top). However, in the map view, Switzerland is completely occluded by neighboring data. A data-dependent analysis of the interface would directly reveal this bias against such data points by examining the weights derived from screen space. Similarly, data-dependent analysis could reveal if the bias *toward* particular data points is proportional to bias in the dataset.

We have only done a partial analysis of the Gapminder interfaces, but we expect similar patterns to be components of full-application analysis. Just observing structural patterns, these patterns can illustrate potential biases. For example, isolated groupings show areas that may be difficult to move between - a bias for staying with the current representation. Moving to more algorithmic analysis, it would be possible to identify unreachable and difficult-to-access data.

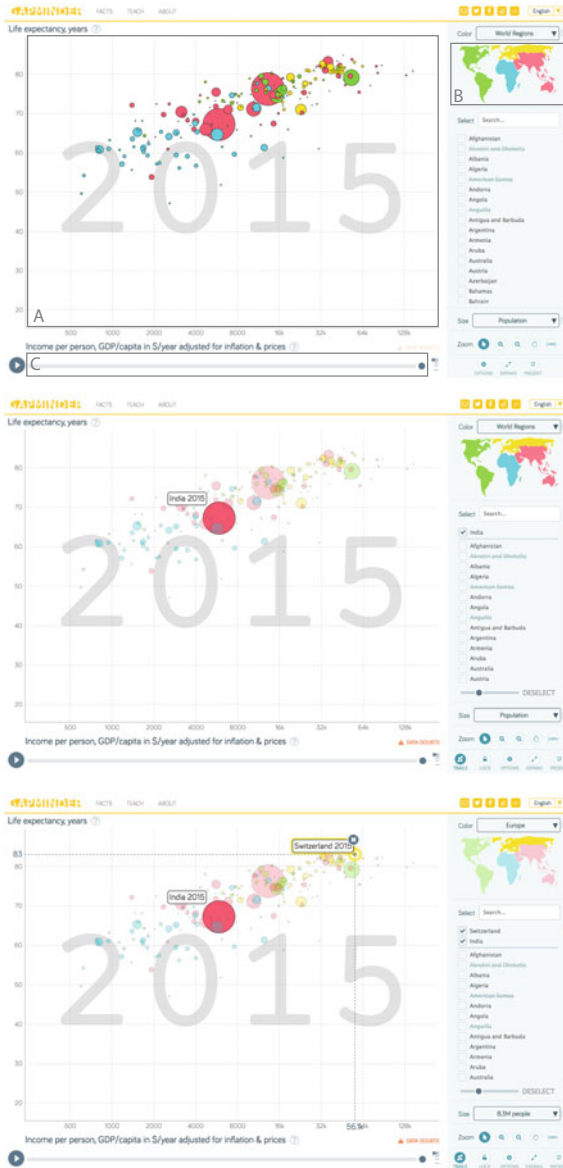


Fig. 4.2 Series of images of a Gapminder “Bubbles” view in sequential states: (top) initial 2015 data, (middle) select India, (bottom) hover Switzerland. Images from gapminder.org, CC-BY license. In the top image, (A) indicates the map plot window, (B) are the interface settings controls, and (C) is the playback controls to show animations over time. The data is shown with Income per Person on the x-axis and Life Expectancy (in years) on the y-axis. The size of the circles represent population, and the circles are colored by region

Modeling with weights in proportion to a target’s area is at least partially justified by the Shannon entropy interpretation of Fitts’ Law [8]. In brief, if a longer sequence of actions (or a sequence of more unlikely actions) is required to reach a state, that state is less likely to be encountered by chance. A sequence of user actions can be viewed as a string that encodes the address of an interface state. In terms of information, if bits of information must be supplied to “address” a state, the likelihood of an error increases. If there are more redundant paths, it is analogous to encoding redundancy and the state is more likely.

The Markov chain conceptualization for data-dependent biases derives the transition probabilities, p_{ji} , from this weighting schema. We are capturing biases where the transition probabilities shape Markov chains to end up in a particular part of the state space or make some transitions more likely than others. With data in the system, we are measuring some of the technical biases. The data representations reflect the results of the underlying encoding/embedding schemes and choice of machine learning or analytic algorithms. These technical choices can bias the data available in the system. Pre-existing biases may come into play if the system is applied to data types for which it was not designed, because the norms and practices will not properly apply. This would occur, for example, if numerical techniques are applied ineffectively to encode text data. But predominantly, data-dependent Markov chains capture technical system biases.

This preliminary analysis makes it evident that the basic procedure naïvely applied yields a combinatorial explosion of states. For example, sequential data selection is done when picking specific countries in the Gapminder “Bubbles” chart. A full model is a lattice of all possible combinations of selections (A, B, C, A&B, A&B&C, A&C, B&C, etc.). For all but trivial examples, this is likely to be computationally intractable. Tempering full data dependence is probably necessary and is the focus of the next section. In truth, a mixture of data-dependent and data-independent modeling is likely to yield the best *tractable* models. Some of the simplifications used in Dabek and Caban [4] reduce the impact of redundant combinations may also have analogous simplifications for this a priori modeling.

Data Independent Modeling

Interesting patterns in the interface may be revealed by ignoring details of the data presentation. In the data-independent scenario, the resulting model is simplified but necessarily more abstract. It is constructed in the same way as the data-dependent bias case but with two simplifications. First, all interactions that directly involve the data are collapsed into a single link by type. For example, instead of a selection-related link for each data point, there is a single data-selection link. This necessarily implies that data-related states are also compressed together. The general transformation is shown in the difference between the top and bottom row in the left column of Fig. 4.3. Second, because we are no longer considering the data representation, we can no longer use screen-space to weight the links. Instead, we propose to make all links that leave a node equally likely. This is termed a *regular* Markov chain, with the transition probability matrix $P = \left[\frac{1}{n} \right]$. This initial assumption provides a baseline against which we can study a system.

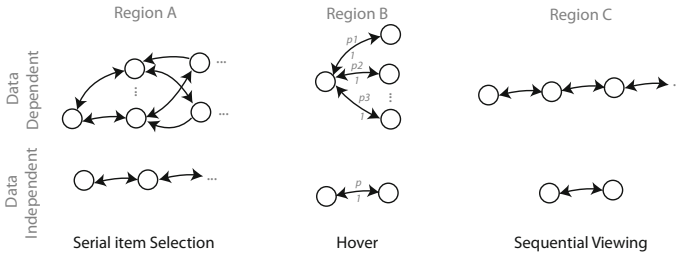


Fig. 4.3 Markov model structures from Gapminder “Bubbles” regions noted in Fig. 4.2(top). The difference between the data-dependent and data-independent cases is evident in the difference of complexity between the rows

Data-independent Markov chains have transition probabilities that are regular or are shaped by the initial conditions of the system. If the transition probabilities are dependent on initial conditions, we are capturing a pre-existing bias in the system. That is, the assumptions made by the designer as to default settings produced a bias toward data availability that changed when those default settings were adjusted to some alternative initial configuration. Additional pre-existing biases are captured in the overall design elements in the display or choices of representation implemented, because all reflect some methodological attitude or cultural norm for that system. Technical biases can also be revealed if the data-independent display incorporates structures output from some internal algorithm, or the structure reflects technology choices on which the system is implemented. But we argue that data-independent Markov chains serve to capture pre-existing system biases.

Modeling an interface with a specific dataset represented is likely to be more directly actionable than the data-independent model. However, the models are likely to be large relative to the data-independent case because many common interface patterns are combinatoric in the elements of the dataset. Working with the data-independent model has the effect of reducing the size the model significantly, but it makes the results more abstract and thus more difficult to interpret.

4.6 Application: Gapminder Analysis

We prototyped the bias measurement procedure on the Gapminder world map visualization. The target application is show in Fig. 4.1. The data-based components were recreated using Gapminder’s demographic data [12] and geographic centroids [11]. Countries are represented by circles, the areas of which correspond to the income variable. Figure 4.4 shows the basic result of this abstraction.

Estimating bias according to the procedure outlined in Sect. 4.5 requires measuring the proportion of pixels allocated to various interactive elements. This can be accomplished by assigning each interactive element a unique color and counting how



Fig. 4.4 Re-representation of Gapminder Maps with random colors for measuring bias. Circle size represents Income, and position of each circle is the same as in the Maps view in Fig. 4.1

many pixels in the end image contain each color. To properly measure the interface bias, an image must be measured for the interactive state of each interface element. In the Gapminder Map, the main map interaction is selecting countries. When a country is selected its label is rendered top-most and can be used for selection in the same way that the circles can be. Therefore, the measurement process creates a separate image for each interactive state. In this case, each image corresponds to selecting a different country and includes a label box for the selected country. The label box is filled by the same color as the country because, in the Gapminder map, country labels behave as selection targets in the same way the country's circle does. The underlying map is not directly interactive and is thus omitted from Fig. 4.4. The various controls on the periphery are also omitted from this analysis.

With an image similar to that in Fig. 4.4 generated for each country, the number of pixels allocated for each country can be counted directly. Because the background color is the most common color (comprising 88% of the image), it was omitted from this model. However, in modeling other interfaces it may be valuable to include. Each image corresponds to a state in the Markov model and the percent of pixels for each country corresponds to the transition probabilities.

With the Markov model defined, analysis can proceed. There are two basic measures: the baseline probability and the stable distribution. The baseline probability is the average probability across all possible transitions. Shown in Fig. 4.5 in comparison to the population distribution, the distributions are distinctly different.

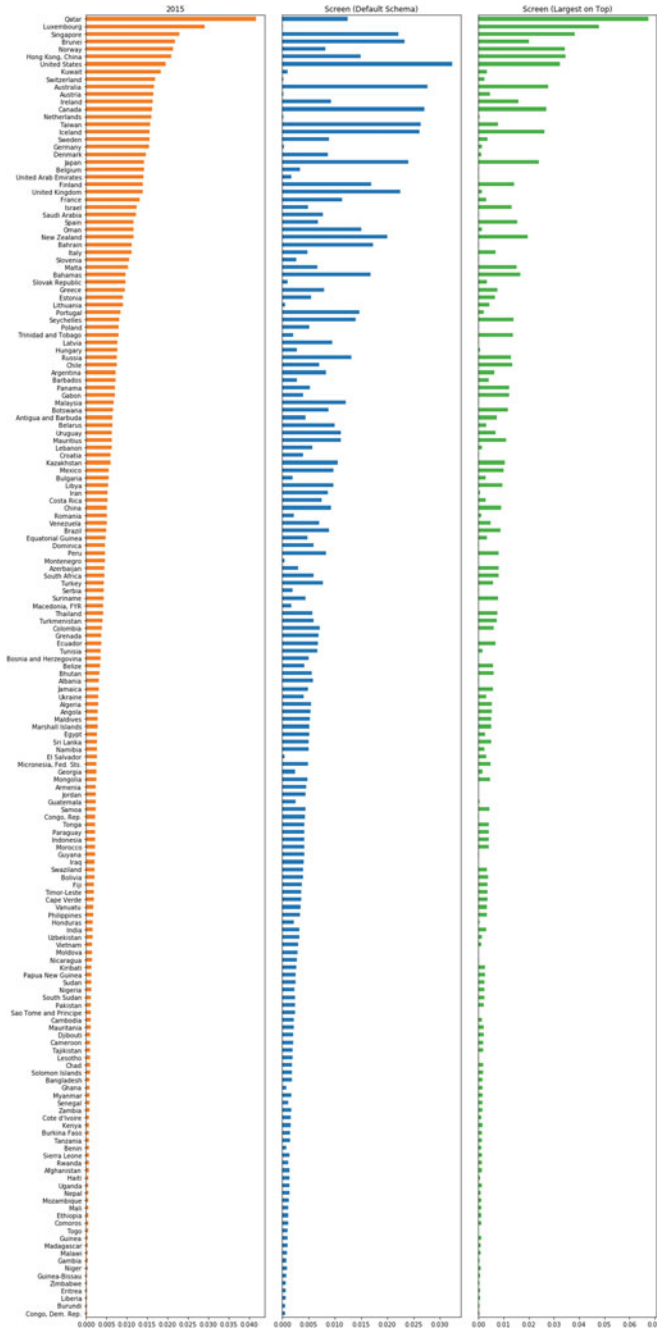


Fig. 4.5 Screen space and proportional incomes compared (ordered by screen space). If screen space were allocated proportional to income, the blue and green series of bars would both monotonically decrease. Because they do not, there is disproportionate representation

Treating screen and data proportion as a sorted list, the relationship between the two can be measured with Spearman's rank correlation (i.e., Spearman's ρ). ρ ranges from -1 to 1 , corresponding to inversely ordered to identically ordered. A value of 0 indicates that the orders are unrelated, and is the null hypothesis. With an ordering of countries based on screen proportion and another based on data proportion, $\rho = -0.02$, with $p = 0.75$. Relative to the common type I error rate $\alpha = 0.05$, this result indicates that the orders cannot be distinguished from random. We must conclude that any bias in the visualization is not related the distribution in the source income data.

The other basic measure is to look at the stable distribution, essentially modeling what random walks across the interface would produce. The data-dependent case, where the actual data values are used to scale the circles, is shown in Fig. 4.6. It is clear that there is a significant bias towards specific countries, but that bias is not matched by the per-capita income of those countries. In fact, several highly populated countries are at the bottom of the distribution (Luxembourg, Switzerland, Austria, Netherlands, Germany, Montenegro, El Salvador), but all are in regions of the world with many political boundaries close together such as Europe and Central America. In contrast, the top of the distribution (United Kingdom, Brunei, Japan, Iceland, Taiwan, Canada, Australia, United States) is made of geographically isolated countries, even though they do not have the highest per-capita incomes.

Interpreting these results requires knowing what the *desired* outcome is. The argument for approximately equal distributions is that each item is equally selectable. Our analysis indicates that the Gapminder interface essentially supports this type of analysis when used interactively, but only when used interactively. In contrast, if a bias that follows the data distribution is desired (that the answers should "pop out") this layout fails both interactively (where distributions are too even) and statically (where the image does not allocate pixels proportional to the source data).

The data-independent analysis reveals limits about the interface regardless of exact data values. In this analysis, images were generated where each country was given the same value. Exploring different cases involved using different assigned values. To provide an even distribution statically or in the stable distribution at common screen (100 dpi) or print resolutions (300 dpi), the circle for each country would need to be smaller than a single pixel. This is impractical, and thus we conclude that the map layout provided is incapable of providing an even bias.

Our re-implementation of the Gapminder interface is not perfect. There are three main differences. First, Gapminder's actual interface uses an area-preserving geographic projection (or a compromise project that includes area-preservation as a partial criterion). For simplicity, we used an equi-rectangular projection. This does not affect the procedural validity, but it likely influences the exact weights in the Markov model as overlapping regions may shift around. It is likely that our analysis reports less bias than a matching projection because much of the bias is found in Europe, which is more compressed in most area-preserving projections than in equi-rectangular. Second, We have omitted the controls surrounding the main map and the background map itself. The background map was omitted because it is essentially non-interactive. Other controls were omitted for simplicity of analysis.

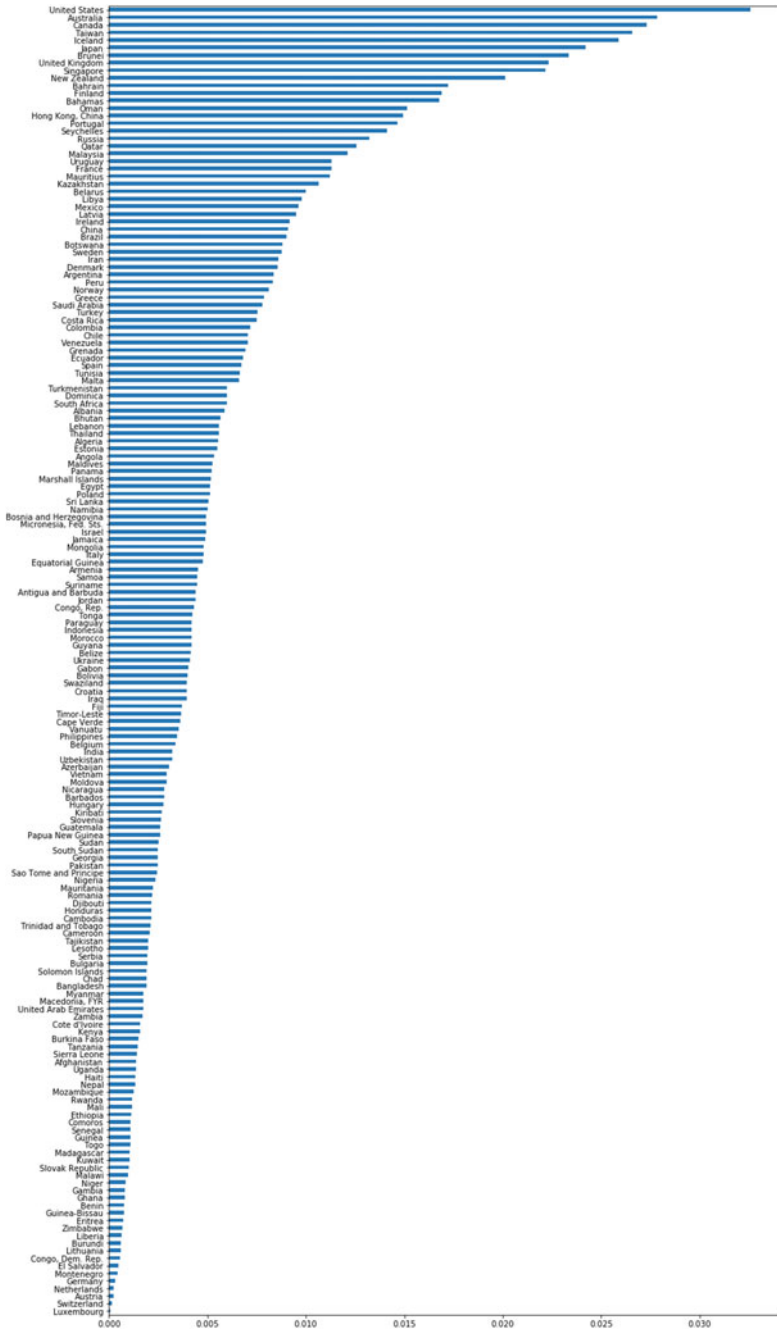


Fig. 4.6 Interactive, data-dependent Markov-modeled bias of the Gpmindr Maps interface

Finally, the scaling factor used for circles approximates that of the Gapminder interface, but is not a perfect match.

Our analysis can be used to directly explore alternative implementation decisions. For example, to faithfully reproduce the Gapminder Maps interactive interface elements, country circles are rendered such that smaller values lay on top of larger values. This makes it more likely that small countries will be selected than their proportion of the data would indicate. Using our analysis techniques, we can also measure what the bias *would be* if countries were rendered in other orders. If large income countries were rendered on top, the interactive case appears more like the background data distribution (see Fig. 4.5, right column), but not sufficiently to be statistically significant (Spearman's $\rho = 0.06$, $p = 0.38$). This indicates that the bias is dominated by something other than rendering order.

Rendering the order by other data values would also provide other bias profiles, some of which may be useful for specific contexts (e.g., conditioning income maps by population may bias the interface towards discovering patterns in poverty). This approach provides opportunities to explore interface decisions and how they may be made in context-specific ways.

4.7 Discussion

Classic Markov modeling is a “memory free” technique. It only takes the current state into consideration when making a transition. However, data exploration necessarily includes human memory [16]. Modeling multi-step memory with static Markov models is cumbersome at best (and practically impossible in combinatoric cases). However, compressing combinatoric cases into abstract chains (as discussed earlier) can be seen as a simple memory model. A similar compression technique might be used to model a simple form of memory. An alternative to combinatoric compression of states would be to use a model that includes memory in a structured way. Dynamic Markov, Push-down automata, and RAM-based automata (with limited RAM) are also viable options. Each has a finite state space and a well-developed field of analysis.

Our proposed weighting scheme is simple, and may not be sufficient to illuminate some bias patterns. There are some interesting challenges. For example, in the data-dependent construction, the size-based weighting is derived from Fitts' Law. However, Fitts' Law does not account for convention or attention. Therefore, some interface elements may be relatively large by convention but the probability that they will be interacted with is not proportional to their size. For example, menu bars have a size and position dictated by the interface guidelines of the platform, and that may be significantly larger than the representation of a single data point. Capturing such differences in the interaction probabilities requires reaching beyond Fitts' Law for transition probabilities.

In the data-dependent Markov modeling, *only* the screen real-estate is used to model direct data interactions. Logical extensions include using visual similarity (along many retinal dimensions) to up-weight or down-weight items. This could be extended further with a dynamic Markov model, so weights change based on what

states have been visited in earlier interactions. Proper dynamic weighting requires knowledge of the task as well as the visual representation. It makes sense to up-weight similar things when the retinal variables correspond to the desired task but to (possibly) down-weight similar items when the retinal variable does not have a bearing on the task. Also, exploration versus verification probably has different interaction patterns. Such modeling may be achieved using a Markov Decision Process. In addition to a transition probability, the model is extended with a payoff matrix and a “discount” factor. Payoffs are provided when a specific transition is taken. The discount factor determines whether immediate payoffs or future expected payoffs are prioritized. Decisions are still based on the information observable in the current state, but the probability of a transition is made a factor of the base probability, the payoff, the expected future payoff and the discount factor. Payoff and discount factors can be adjusted to model different goal-directed behaviors. Similar dynamic re-weighting is done in Dabek and Caban [4], captured in their “ideology” factors.

Analytic provenance models suggest another approach to Markov modeling. In particular, if a provenance tracking system records information about the state of the interface, we could use a hidden Markov model to derive the Markov chain of the original interface state space [7]. This might be helpful in cases where we have incomplete information about the structure or state space of an interface. This inference process could leverage existing graph modeling systems for analytic provenance, as in GraphTrail [5], to interpret the hidden model states. This approach bears some similarity to Jankun-Kelly’s [14] P-set Model of visualization exploration. He defines two key concepts. A P-set is a set of parameters that define a visualization system, and visualization transformation is an operation on the P-set that creates a particular visualization view. Each set of parameter values (P-set) defines a state space with weighted connections (transformations) between the states. The difference between our Markov chain approach is that our links between the states quantify the probability of moving between states, rather than defining the parameter transformations themselves. An interesting direction for future work is to relate the transformations to transition probabilities between parameter states to capture emergent bias.

4.8 Conclusion

We note that methods for measuring information content in a visual analytic system remain an open challenge for the field [19]. Such measures are important for the overall evaluation of systems, particularly for calibrating our expectations for how much information users may be able to extract from a system. We propose that measurement of information availability and the interface biases that may shape that information availability should be modeled in systems before they are put into human-in-the-loop evaluations. Markov models, as proposed herein, provide a promising direction for conceptualizing the state space of a visual analytic system and understanding system-level biases through the transition probabilities over the state space.

Acknowledgements This research was sponsored by the Analysis in Motion Initiative at the Pacific Northwest National Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

1. Ballard DH (1997) An introduction to national computation. MIT Press, Cambridge, MA
2. Cook K, Cramer N, Israel D, Wolverton M, Bruce J, Burtner R, Endert A (2015) Mixed-initiative visual analytics using task-driven recommendations. In: 2015 IEEE conference on visual analytics science and technology (VAST). IEEE, New York, pp 9–16
3. Cook KA, Thomas JJ (2005) Illuminating the path: the research and development agenda for visual analytics. IEEE Computer Society, Los Alamitos, CA
4. Dabek F, Caban JJ (2017) A grammar-based approach for modeling user interactions and generating suggestions during the data exploration process. *IEEE Trans Visual Comput Graphics* 23(1):41–50
5. Dunne C, Henry Riche N, Lee B, Metoyer R, Robertson G (2012) Graphtrail: analyzing large multivariate, heterogeneous networks while supporting exploration history. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, New York, pp 1663–1672
6. Endert A, Fiaux P, North C (2012) Semantic interaction for sensemaking: inferring analytical reasoning for model steering. *IEEE Trans Visual Comput Graphics* 18(12):2879–2888
7. Endert A, Ribarsky W, Turkay C, Wong B, Nabney I, Blanco ID, Rossi F (2017) The state of the art in integrating machine learning into visual analytics. *Comput Graphics Forum*. <https://doi.org/10.1111/cgf.13092>
8. Fitts PM (1954) The information capacity of the human motor system in controlling the amplitude of movement. *J Exp Psychol* 47(6):381–391
9. Friedman B (1996) Value-sensitive design. *Interactions* 3(6):16–23
10. Friedman B, Nissenbaum H (1996) Bias in computer systems. *ACM Trans n Inf Syst (TOIS)* 14(3):330–347
11. Gapminder Foundation. Gapminder.org: Geography. <https://www.gapminder.org/data/geo/> (2018). Accessed 01 Apr 2018
12. Gapminder Foundation. Gapminder.org: List of indicators in gapminder world. <https://www.gapminder.org/data/> (2018). Accessed 01 Apr 2018
13. Gotz D, Zhou MX (2009) Characterizing users’ visual analytic activity for insight provenance. *Inf Visual* 8(1):42–55
14. Jankun-Kelly T (2008) Using visualization process graphs to improve visualization exploration. In: International provenance and annotation workshop. Springer, Berlin, pp 78–91
15. Kemeny JG, Snell JL (1960) Finite Markov chains, vol 356. van Nostrand, Princeton, NJ
16. Patterson RE, Blaha LM, Grinstein GG, Liggett KK, Kaveney DE, Sheldon KC, Havig PR, Moore JA (2014) A human cognition framework for information visualization. *Comput Graphics* 42:42–58
17. Wall E, Blaha L, Franklin L, Endert A (2017) Warning, bias may occur: a proposed approach to detecting cognitive bias in interactive visual analytics. In: IEEE visual analytics science and technology (VAST). IEEE, New York
18. Wall E, Blaha L, Paul CL, Cook K, Endert A (2017) Four perspectives on human bias in visual analytics. In: Ellis G (ed) Cognitive biases in visualizations, Chap. 3. Springer, Berlin
19. Ward MO, Grinstein G, Keim D (2010) Interactive data visualization: foundations, techniques, and applications. CRC Press, Natick, MA
20. Wattenberg M (1999) Visualizing the stock market. In: CHI’99 extended abstracts on human factors in computing systems. ACM, New York, pp 188–189
21. Xu K, Attfield S, Jankun-Kelly T, Wheat A, Nguyen PH, Selvaraj N (2015) Analytic provenance for sensemaking: a research agenda. *IEEE Comput Graphics Appl* 35(3):56–64