

A view from computational journalism.**BY NICHOLAS DIAKOPOULOS**

Accountability in Algorithmic Decision Making

EVERY FISCAL QUARTER, automated writing algorithms churn out thousands of corporate earnings articles for the Associated Press based on little more than structured data. Companies such as Automated Insights, which produces the articles for the AP, and Narrative Science can now write straight news articles in almost any domain that has clean and well-structured data: finance, sure, but also sports, weather, and education, among others. The articles are not cardboard either; they have variability, tone, and style, and in some cases readers even have difficulty distinguishing the machine-produced articles from human-written ones.⁴

It is difficult to argue with the scale, speed, and labor-saving cost advantage that such systems afford. But the trade-off for media organizations appears to be nuance and accuracy. A quick search on Google for “‘generated by Automated Insights’ correction” yields results for thousands of articles that were automatically written, published, and then had to have corrections issued.

The errors range from relatively innocuous ones about where a company is based, to more substantial wrong word choices—*missing* instead of *beating* earnings expectations, for example. Were any of these market-moving errors? Was the root cause bad data, a faulty inference, or sloppy engineering? What is the right way to post corrections?

Algorithmic curation of content is also behind some of the most important and influential news-dissemination platforms that virtually all of us use. A recent Pew study found Facebook is a source of news about government and politics for 61% of millennials,¹⁸ yet a majority of the public is not aware the Facebook newsfeed is algorithmically curated.¹¹ This becomes a lot more problematic when you consider Facebook can affect voter turnout in elections based merely on the amount of hard news promoted in an individual’s news feed.²⁴ This bit of information, together with recent research showing biased search results can shift the voting preferences of undecided voters,¹⁰ points to the need to start asking questions about the degree to which such curation and ranking systems can affect democratic processes.

These are just a few examples of algorithms influencing our media and information exposure. But the impact of automated decision making is being felt throughout virtually all strands of industry and government, whether it be fraud-detection systems for municipalities managing limited resources, a formula that grades and ranks teacher performance, or the many ways in which dynamic product pricing is done by Amazon, Airbnb, or Uber.⁸ It is time to think seriously about how the algorithmically informed decisions now driving large swaths of society should be accountable to the public. In the face of important or expensive errors, discrimination, unfair denials of public services, or censorship, when and how should algorithms be reined in?

Computer science and engineering



professionals have a role to play here. While autonomous decision making is the essence of algorithmic power, the human influences in algorithms are many: criteria choices, optimization functions, training data, and the semantics of categories, to name just a few. Often a human operator is involved in a final decision only to have been influenced by the algorithm's nudging and suggestions along the way.

Algorithmic Decision Making

It is helpful first to get the lay of the land in terms of the different types of atomic decisions that algorithms make. These include processes that prioritize, classify, associate, and filter.

Prioritizing is something we do on a daily basis to cope with the information onslaught. As beings with limited time and attention, we cannot ignore the need to economize. Algorithms prioritize information in a way that emphasizes or brings attention to certain things at the expense of others; by

definition prioritization is about discrimination. As a result, there may be ramifications to individuals or other entities that should be considered during design. Search engines are canonical examples, but there are many other consequential rankings—for everything from the quality of schools and hospitals, to the riskiness of illegal immigrants on watch lists.¹⁴ The criteria used in a ranking, how they are defined and datafied, and their weighting are essential design decisions that deserve careful consideration and scrutiny.

Classification decisions mark a particular entity as belonging to a given class by considering key characteristics of that entity. Class membership can then drive all kinds of downstream decisions. The opportunities for bias, uncertainty, or outright mistakes are plentiful in automated classification. The training data that is the basis for supervised machine-learning algorithms is an important consideration, given the human biases that may be

lurking there. Recently published research by Shilad Sen and collaborators underscores the need to consider the cultural community from which training data is collected.²³ Data crowdsourced from Mechanical Turk may be useful for widely shared and agreed-upon knowledge but introduces discrepancies in other cases. The bottom line, as they write, is this: "When collecting a gold standard researchers and practitioners must consider the audience of the gold standard, the system or algorithm that uses it, and the type of knowledge."

In developing classification algorithms, designers must also consider the accuracy of the classifications: the false positives and false negatives, and the consequences to stakeholders of either of those types of errors. For example, in Boston a man classified as having a fraudulent driver's license (a false positive in this case) was unable to work until the matter was resolved. Classification algorithms can be tuned

to make fewer of either type of mistake, but as one goes down the other goes up. Tuning can grant privilege to different stakeholders and outcomes in a decision, implying that designers make an essential value judgment when balancing error rates.¹⁶

Association decisions revolve around creating relationships between entities. The semantics of those relationships can vary from the generic “related to” or “similar to” to distinct domain-specific meanings. These associations lead to connotations in their human interpretation. For example, when a man in Germany searched for his name on Google and the autocomplete suggestions connected him to “scientology” and “fraud,” this was both meaningful and unsettling to the man, leading to a defamation lawsuit that he ultimately won.⁶ Collaborative filtering is a popular class of algorithm that defines an association neighborhood (a cluster, really) around an entity and uses those close ties to suggest or recommend other items.¹³ The quantification bugbear torments associations just as it does rankings and classifications. The criteria that are defined and measured, and the similarity metrics that dictate how closely two entities match, are engineering choices that can have implications for the accuracy of an association, both objectively and in terms of how that association is interpreted by other people.

One issue with the church of big data is its overriding faith in correlation as king. Correlations certainly do create statistical associations between data dimensions. But despite the popular adage, “Correlation does not equal causation,” people often misinterpret correlational associations as causal. The man whose name was associated with fraud on Google may or may not be the cause of that association, but we certainly read it that way. This all indicates a challenge in communicating associations and the need to distinguish correlative vs. causal associations.

Filtering decisions involve including or excluding information according to various rules or criteria. Often this emerges at the user-interface level in, for example, news-reading applications such as Facebook or Flipboard. Since there is practically always some troll or miscreant willing to soil the sandbox, moderation and filtering

are crucial elements when publishing social media. Online comments are sometimes filtered algorithmically to determine whether or not they are anti-social and therefore unworthy of public consumption. Of course, the danger here is in going too far—into censorship. Censorship decisions that may be false positives should be carefully considered, especially in cultures where freedom of speech is deeply ingrained.

Ultimately, when considering the various decisions and predictions that an algorithm may make, particularly ones that can affect people, but also those that affect property, you must consider the errors and potential for discrimination and censorship that can arise. Just read ACM’s ethics policies.^{1,2}

The ACM Code of Ethics for software engineering lists eight principles that are aspirations for professional behavior. First and foremost is that software engineers should act in the *public interest*: to be accountable and responsible for their work, to moderate private interests with public good, to ensure safety and privacy, to avoid deception, and to consider the disadvantaged. The general moral imperatives of ACM include “avoid harm to others,” “be fair and take action not to discriminate,” and “respect the privacy of others.”

Let that sink in.

Have you ever programmed an algorithm that could violate any of these mandates? It may not have been intentional, but there are side effects you might have noticed if you had done more thorough benchmarking or considered the human contexts in which the output of your algorithmic creations would be used. Is it possible that you used a protected trait such as race, ethnicity, religion, nationality, gender, sexuality, disability, marital status, or age in an inappropriate way? The point is these ethical ideals need to be incorporated throughout the engineering process, for people to be constantly reconsidering: What are the consequences of the unlikely false positive, or the impacts of how criteria are measured and defined in training datasets? Helen Nissenbaum was far ahead of the curve when, almost two decades ago, she recommended the development of explicit standards of care including rigorous engineering guidelines that would consider these issues.²¹

Government vs. Private Sector Accountability

The mandate for accountable algorithms (and for the accountability of the people behind them) for government usage is a bit different from that for the private sector. In the case of the modern democratic state, citizens elect a government that provides social goods and exercises its power and control in a way that is moderated through norms and regulation. The government is legitimate only to the extent it is accountable to the citizenry. But algorithms are largely unregulated now, and they are indeed exercising power over individuals or policies in a way that in some cases (for example, hidden government watch lists) lacks any accountability whatsoever. A recent academic review of Social Security Administration models used to predict life expectancy and solvency found systematic underestimation, implying that funds were on firmer ground than warranted.¹⁵ We, the governed, should find it unacceptable there is no transparency or even systematic benchmarking and evaluation of these forecasts, given the important policy decisions they feed.

Corporations, on the other hand, do not have the same mandate for public accountability, though they may sometimes be impelled to act through social pressure (for example, boycotts). Perhaps more compelling is the capitalist argument that higher data quality and thus better inference will lead to more satisfied customers. The most clear-cut way to do this is to design processes that adjudicate and facilitate the correction of false positives by end users. Allowing users to inspect, dispute, and correct inaccurate labels in the data would improve overall data quality for machine-learning applications.


Transparency can be a mechanism that facilitates accountability, one that we should demand from government and exhort from industry. Corporations often limit their transparency out of fear of losing a competitive advantage from a trade secret or of exposing their systems to gaming and manipulation. Complete source-code transparency of algorithms, however, is overkill in many if not most cases. Instead, the disclosure of certain key pieces of information, including aggregate results

and benchmarks, would be far more effective in communicating algorithmic performance to the public.


When automobile manufacturers disclose crash-test results, they do not tell you the details of how they engineered the vehicle. When local municipalities publish restaurant inspection scores, they do not disclose a restaurant's unique recipes. The point is there are models for transparency that can effectively audit and disclose information of interest to the public without conflicting with intellectual property and trade secrets. In some cases fear of manipulation and gaming of disclosed criteria are unfounded. For example, criteria not based on user behaviors offers no mechanism for gaming from individuals who have no direct control over those attributes. In some cases gaming or manipulation of an algorithm might even be a good thing. For example, if credit-rating agencies disclosed the criteria they used to score individuals, wouldn't it be a great thing if everyone gamed their score? They would have to act financially responsibly in that game.

For government, the Freedom of Information Act (FOIA) and similar laws in the U.S. and many other jurisdictions compel disclosure of government records and data when requested, though there are, of course, exceptions, such as when a government integrates a third-party system protected by trade secrets. There has been at least one successful use of an FOIA request to compel the disclosure of government source code.¹⁹ In another FOIA case decided in 1992, the Federal Highway Administration resisted disclosing the algorithm it used to compute safety ratings for carriers, but ultimately lost in court to a plaintiff that successfully argued the government must disclose the weighting of factors used in that calculation.⁹

Thus, FOIA is of some use in dealing with the government use of algorithms, but one of the issues with current FOIA law in the U.S. is it does not require agencies to create documents that do not already exist. Hypothetically, a government algorithm could compute a variable in memory that corresponds to some protected class such as race and use that variable in some other downstream decision. As long as that



Allowing users to inspect, dispute, and correct inaccurate labels in the data would improve overall data quality for machine-learning applications.



variable in memory was never directly stored in a document, FOIA would be unable to compel its disclosure. Audit trails could help mitigate this issue by recording stepwise correlations and inferences made during the prediction process. Guidelines should be developed for when government use of an algorithm should trigger an audit trail.³

It may be time to reconsider FOIA regulation along the lines of what I propose be called the Freedom of Information Processing Act (FOIPA). FOIPA would sidestep the issues associated with disclosing formulas or source code and instead allow the public to submit benchmark datasets the government would be required to process through its algorithm and then provide the results. This would allow interested parties, including journalists or policy experts, to run assessments that prod the government algorithm, benchmark errors, and look for cases of discrimination or censorship. For example, you could take two rows of data that varied on just one piece of sensitive information like race and examine the outcome to determine if unjustified discrimination occurred.

An Algorithmic Transparency Standard

So far we have covered a lot of ground: from the decisions that algorithms make, to the stakes of errors, and the ethics of responsibly engineering these systems. But you may still be asking yourself this overriding question: What can we and should we be disclosing about our algorithms?

To help answer that question I led the organization of a workshop on Algorithmic Transparency in the Media at Columbia University's Tow Center for Digital Journalism in spring 2015. About 50 people from the news media and academia convened to discuss how to work toward ideas that support a robust policy of news and information stewardship via algorithms. We discussed case studies on "Automatically Generated News Content," "Simulation, Prediction, and Modeling in Storytelling," and "Algorithmically Enhanced Curation," and brainstormed dimensions of the various algorithms in play that might be disclosed publicly.


Based on the wide array of ideas generated at the workshop, we came up

with five broad categories of information that we might consider disclosing: human involvement, data, the model, inferencing, and algorithmic presence.


Human involvement. At a high level, transparency around human involvement might involve explaining the goal, purpose, and intent of the algorithm, including editorial goals and the human editorial process or social-context crucible from which the algorithm was cast. Who at your company has direct control over the algorithm? Who has oversight and is accountable? Ultimately we want to identify the authors, or the designers, or the team that created and are behind this thing. In any collective action it will be difficult to disaggregate and assign credit to exactly who did what (or might be responsible for a particular error),²¹ yet disclosure of specific human involvement would bring about social influences that both reward individuals' reputations and reduce the risk of free riding. Involved individuals might feel a greater sense of public responsibility and pressure if their names are on the line.

Data. There are many opportunities to be transparent about the *data* that drives algorithms in various ways. One avenue for transparency here is to communicate the quality of the data, including its accuracy, completeness, and uncertainty, as well as its timeliness (since validity may change over time), representativeness of a sample for a specific population, and assumptions or other limitations. Other dimensions of data processing can also be made transparent: how was it defined, collected, transformed, vetted, and edited (either automatically or by human hands)? How are various data labels gathered, and do they reflect a more objective or subjective process? Some disclosure could be made about whether the data was private or public, and if it incorporated dimensions that if disclosed would have personal privacy implications. If personalization is in play, then what types of personal information are being used and what is the collected or inferred profile of the individual driving the personalization?

The model itself, as well as the modeling process, could also be made transparent to some extent. Of high importance is knowing what the model



The main challenge moving forward is to determine appropriate mechanisms for disclosure that are beneficial but do not kill usability.



actually uses as input: What features or variables are used in the algorithm? Often those features are weighted: What are those weights? If training data was used in some machine-learning process, then you would characterize the data used for that along all of the potential dimensions described here. Some software-modeling tools have different assumptions or limitations: What were the tools used to do the modeling?

Of course, this all ties back into human involvement, so we want to know the rationale for weightings and the design process for considering alternative models or model comparisons. What are the assumptions (statistical or otherwise) behind the model, and where did those assumptions arise? And if some aspect of the model was not exposed in the front end, why was that?

Inferencing. The inferences made by an algorithm, such as classifications or predictions, often leave questions about the accuracy or potential for error. Algorithm creators might consider benchmarking against standard datasets and with standard measures of accuracy to disclose some key statistics. What is the margin of error? What is the accuracy rate, and how many false positives versus false negatives are there? What kinds of steps are taken to remediate known errors? Are errors a result of human involvement, data inputs, or the algorithm itself? Classifiers often produce a confidence value, and this, too, could be disclosed in aggregate to show the average range of those confidence values as a measure of uncertainty in the outcomes.

Algorithmic presence. Finally, we might disclose if and when an algorithm is being employed at all, particularly if personalization is in use, but also just to be made aware of, for example, whether A/B testing is being used. Other questions of visibility relate to surfacing information about which elements of a curated experience have been filtered away. In the case of Facebook, for example, what are you not seeing, and, conversely, what are you posting (for example, in a news feed) that other people are not seeing.

Technical systems are fluid, so any attempt at disclosure has to consider the dynamism of algorithms that may be continually learning from new data. The engineering culture must

become ingrained with the idea of continual assessment. Perhaps new multidisciplinary roles relating to algorithmic risk modeling or transparency modeling need to be created so these questions receive dedicated and sustained attention.

In the case of information disclosure that would make an entity look bad or be substantially damaging to its public image, I am enough of a pragmatist not to expect voluntary compliance with any ethical mandate. Entities use information disclosure to engage in strategic impression management. Instead, we might look toward regulations that compel information disclosure or at least routine audits around key algorithmically influenced decisions, such as credit scoring.³ The dimensions of information disclosure articulated here could also feed such regulatory designs.

In other cases, a more adversarial approach may be necessary for investigating black-box algorithms. In the journalism domain, I refer to this as algorithmic accountability reporting,⁵ and it involves sampling algorithms along key dimensions to examine the input-output relationship and investigate and characterize an algorithm's influence, mistakes, or biases. This is an extension of traditional investigative accountability journalism, which for many years has had the goal of exposing malfeasance and misuse of power in government and other institutions.

To provide a flavor of this type of reporting, I started investigating in early 2015 the much-publicized Uber surge-pricing algorithm.⁸ The ride-sharing app uses dynamic pricing to "encourage more drivers to go online" and try to match supply and demand. It is an easy line to buy and appeals to basic economic theories. My investigation, based on an analysis of a month's worth of pricing data in Washington, D.C., indicated that instead of motivating a fresh supply of drivers to get on the road, surge pricing instead redistributes drivers already on the road. This is important because it means the supply of drivers will shift toward neighborhoods offering higher surge prices, leaving other neighborhoods undersupplied and with longer waiting times for a car. Uber cars are rival goods, and the analysis raises ques-

tions about which neighborhoods end up with better or worse service quality. Higher prices and better service for some means worse service for others.

Challenges Ahead

There is still much research to be done to understand when and how best to act responsibly and be transparent about the algorithms we build. Deciding what to disclose is just a start; the communication vehicle also needs to be explored. Human-computer interaction as well as machine learning and software engineering have roles to play here.

This article has broadly articulated classes of information that might be disclosed about algorithms: the human element, data, the model, inferences, and the algorithmic presence. Practically speaking, however, each algorithm is a bit different and must be understood in context to determine what can be disclosed. This is both a technical process as well as a human-centered one. We must develop a process of information-disclosure modeling that includes thinking about how the public would use any particular bit of information disclosed.

Providing transparency and explanations of algorithmic outputs might serve a number of goals, including scrutability, trust, effectiveness, persuasiveness, efficiency, and satisfaction. Ultimately, we need to do user modeling and think through a series of questions such as what we want to accomplish with each bit of disclosed information, and what behavior we are trying to affect. What are the decisions the public would make based on that information? What information could be disclosed that would make those decisions more effective, or mitigate risks? How might a user respond to this information?²⁵

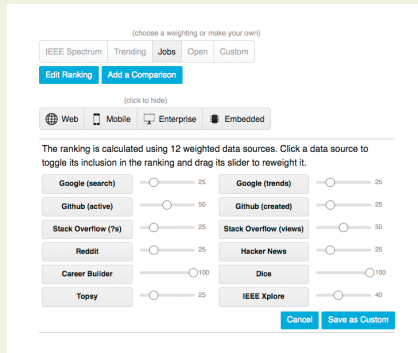
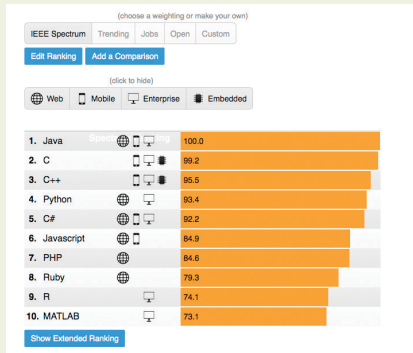
Another dimension to the human-interface challenge is to design effective user experiences for transparency information. Recent research has shown algorithmic transparency information can lead to a better outcome but comes at the expense of enjoyable and reassuring usage.²² The main challenge moving forward is to determine appropriate mechanisms for disclosure that are beneficial but do not kill usability. Additionally, some users may simply not care, while oth-

ers are deeply interested, raising the design challenge of accommodating the needs of many publics, while not polluting the user experience with a surfeit of information for the uninterested. Of course, algorithmic transparency need not be directly integrated into the user experience. For example, corporations or governments might issue algorithmic transparency reports on a quarterly or yearly basis that would disclose aspects of the five dimensions discussed previously.

One approach gaining some attention in the research community is to develop machine-learning methods that can be explained in ways that humans can readily understand. For example, the Bayesian Rule List (BRL) technique learns a series of human-readable rules that when chained together offer a human-readable explanation of the classifier.¹⁷ Other methods are being developed in natural language generation (NLG) to output text that explains why or how a decision was reached. Imagine if your favorite machine-learning library, say scikit-learn, could explain in a sentence why a particular input case was classified the way it was. That would be useful for debugging, if nothing else.

On the other hand, we might consider integrated presentation strategies that leverage data visualization to succinctly communicate the workings of an algorithm. For example, early research has shown that salient visual explanations such as histograms can be effective for communicating recommendation explanations.¹³ In a collaboration with *IEEE Spectrum*, I built a data-driven app ranking top programming languages that feeds off of 12 different weighted data inputs to arrive at a ranking.⁷ Instead of making it a static, fixed ranking, however, like, say, the annual *U.S. News & World Report* College Rankings, we defined several different weightings. So, for example, you could quickly rank languages weighted toward job listings or open source projects, and you could create your own custom ranking by deciding which data inputs were important to you and reweight them accordingly (see the accompanying figure). You could also visually compare your rankings to do a sensitivity analy-

IEEE top programming languages ranking and reweighting interfaces.



sis and see how a change in a factor would impact the resulting output ranking. Based on 1,285 tweets that people shared about the app, we found about one in six indicated people were reweighting the ranking in various ways. While it is too early to claim victory in designing dynamic and transparent ranking interfaces, this is at least a step in the direction I envision for interactive modeling.

There are technical challenges here, too. In particular, concerns often arise over manipulation and gaming that may be enabled by disclosing information about how systems work. A certain amount of threat modeling may be necessary if transparency is required. If a particular piece of information were made available about an algorithm, how might that be gamed, manipulated, or circumvented? Who would stand to gain or lose? Manipulation-resistant algorithms also need to be designed and implemented. Feature sets that are robust and difficult to game need to be developed.

The software engineering of algorithms also needs to consider architectures that support transparency and feedback about algorithmic state so they can be effectively steered by people.²⁰ Algorithm implementations should support callbacks or other logging mechanisms that can be used to report information to a client module. This is essential systems work that would form the basis for outputting audit trails.

Finally, we must work on machine-learning and data-mining solutions that directly take into account provisions for fairness and anti-discrimination. For example, recent research has explored algorithmic approaches that can iden-

tify and correct for disparate impact in classifiers by statistically transforming the input data set so that prediction of protected attributes is not possible.¹² Additional research is needed in this space as different types of models and data types may demand different technical approaches and adaptations.

Conclusion

Society must grapple with the ways in which algorithms are being used in government and industry so that adequate mechanisms for accountability are built into these systems. The ideas presented here about acting ethically and responsibly when empowering algorithms to make decisions are important to absorb into your practice. There is much research still to be done to understand the appropriate dimensions and modalities for algorithmic transparency, how to enable interactive modeling, how journalism should evolve, and how to make machine learning and software engineering sensitive to, and effective in, addressing these issues. C

Related articles on queue.acm.org

Online Algorithms in High-frequency Trading

Jacob Loveless, Sasha Stoikov, and Rolf Waeber
<http://queue.acm.org/detail.cfm?id=2534976>

AI Gets a Brain

Jeff Barr and Luis Felipe Cabrera
<http://queue.acm.org/detail.cfm?id=1142067>

Other People's Data

Stephen Petschulat
<http://queue.acm.org/detail.cfm?id=1655240>

References

1. ACM. Software Engineering Code of Ethics and Professional Practice, 2015; <https://www.acm.org/about/se-code#full>.

2. ACM Code of Ethics and Professional Conduct. 1992; <https://www.acm.org/about/code-of-ethics>.
3. Citron, D. and Pasquale, F. The scored society: due process for automated predictions. *Washington Law Review* 89 (2014).
4. Clerwall, C. Enter the robot journalist. *Journalism Practice* 8, 5 (2014): 519–531.
5. Diakopoulos, N. Algorithmic accountability: Journalistic investigation of computational power structures. *Digital Journalism* 3, 3 (2015): 398–415.
6. Diakopoulos, N. Algorithmic defamation: the case of the shameless autocomplete. Tow Center for Digital Journalism, 2014.
7. Diakopoulos, N., et al. Data-driven rankings: the design and development of the IEEE Top Programming Languages news app. *Proceedings of the Symposium on Computation + Journalism*, 2014.
8. Diakopoulos, N. How Uber surge pricing really works. *Washington Post Wonkblog* (Apr. 17, 2015).
9. *Don Ray Drive-A-Way Co. v. Skinner*, 785 F. Supp. 198 (D.D.C. 1992); <http://law.justia.com/cases/federal/district-courts/FSupp/785/198/2144490/>.
10. Epstein, R. and Robertson, R.E. The search engine manipulation effect (SEME) and its possible impact on the outcomes of elections. In *Proceedings of the National Academy of Sciences* 112, 33 (2015).
11. Eslami, M. et al. 'I always assumed that I wasn't really that close to [her]': Reasoning about invisible algorithms in the news feed. In *Proceedings of the 33rd Annual ACM SIGCHI Conference on Human Factors in Computing Systems*, 2015.
12. Feldman, M., et al. Certifying and removing disparate impact. In *Proceedings of the 21st ACM International Conference on Knowledge Discovery and Data Mining*, 2015, 259–268.
13. Herlocker, J.L. et al. Explaining collaborative filtering recommendations. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 2000, 241–250.
14. Kalhan, A. Immigration policing and federalism through the lens of technology, surveillance, and privacy. *Ohio State Law Journal* 74 (2013).
15. Kashin, K. et al. Systematic bias and nontransparency in US Social Security Administration forecasts. *Journal of Economic Perspectives* 29, 2 (2015).
16. Kraemer, F. et al. Is there an ethics of algorithms? *Ethics and Information Technology* 13, 3 (2010), 251–260.
17. Letham, B. et al. Building interpretable classifiers with rules using Bayesian analysis. *Annals of Applied Statistics*, 2015.
18. Mitchell, A. et al. Millennials and Political News. Pew Research Center, Journalism and Media (June 1, 2015); <http://www.journalism.org/2015/06/01/millennials-political-news/>.
19. Muckrock. Source code of HEAT SAFETY TOOL, 2011; <https://www.muckrock.com/foi/united-states-of-america-10/source-code-of-heat-safety-tool-766/>.
20. Mühlbacher, T. et al. Opening the black box: strategies for increased user involvement in existing algorithm implementations. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014) 1643–1652.
21. Nissenbaum, H. Accountability in a computerized society. *Science and Engineering Ethics* 2, 1 (1996): 25–42.
22. Schaffer, J. et al. Getting the message?: a study of explanation interfaces for microblog data analysis. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, 2015, 345–356.
23. Sen, S. et al. Turkers, Scholars, 'Arafat' and 'Peace': Cultural communities and algorithmic gold standards. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2015, 826–838.
24. Sifry, M. Facebook wants you to vote on Tuesday. Here's how it messed with your feed in 2012. *Mother Jones* (Oct. 31, 2014); <http://www.motherjones.com/politics/2014/10/can-voting-facebook-button-improve-voter-turnout>.
25. Tintarev, N. and Masthoff, J. A survey of explanations in recommender systems. *Proceedings of the International Conference on Data Engineering*, 2007, 801–810.

Nicholas Diakopoulos is an assistant professor at the University of Maryland, College Park, Philip Merrill College of Journalism, with courtesy appointments in the College of Information Studies and Department of Computer Science. He is also a fellow at the Tow Center for Digital Journalism at Columbia University

Copyright held by author.
 Publication rights licensed to ACM. \$15.00.