"Good” isn’t good enough

Ben Green
Harvard University
bgreen@g.harvard.edu

Abstract

Despite widespread enthusiasm among computer scientists to contribute to “social good,” the field’s efforts to promote good lack a rigorous foundation in politics or social change. There is limited discourse regarding what “good” actually entails, and instead a reliance on vague notions of what aspects of society are good or bad. Moreover, the field rarely considers the types of social change that result from algorithmic interventions, instead following a “greedy algorithm” approach of pursuing technology-centric incremental reform at all points. In order to reason well about doing good, computer scientists must reflexively evaluate their normative commitments, consider the long-term impacts of technological interventions, evaluate algorithmic interventions against alternative reforms, and no longer prioritize technical considerations as superior to other forms of knowledge.

1 Introduction

Across the broad world of computer science, “social good” (or just “good”) has become a term du jour. The aspiration among computer scientists to do good is both commendable and exciting. But because the field lacks the language and methods to consider the complexities of actually achieving positive social change, this well-intentioned movement suffers from several underdeveloped principles. First, computer science lacks robust theories and discourse regarding what “good” actually entails. As a result, the field typically adopts a narrow approach to politics that involves making vague (almost tautological) claims about what social conditions are desirable. Second, computer science lacks an articulation of how to evaluate or navigate the relationship between technological interventions and social impact. The movement to promote social good thus tends to take for granted that technology-centric incremental reform is an appropriate strategy for social progress. Considered from a perspective of substantive equality and anti-oppression, it is not clear that these efforts to do good are, in fact, consistently doing good.

2 There is no universally agreed upon “good”

The computer science community has not developed (nor even much debated) any working definitions of “social good” to guide its efforts. Instead, the field seems to operate on a “know it when you see it” approach, relying on rough proxies such as crime=bad, poverty=bad, and so on. The notable exception is Mechanism Design for Social Good, which articulates a clear research agenda “to improve access to opportunity, especially for communities of individuals for whom opportunities have historically been limited” [1].

In fact, the term “social good” lacks a thorough definition even beyond the realm of computer science. It is not defined in dictionaries like Merriam-Webster, the Oxford English Dictionary, and Dictionary.com, nor does it have a page on Wikipedia, where searching for “social good” automatically redirects to the page for “common good”—a term similarly undefined in computer science parlance [5]. To find a definition one must look to the financial education website Investopedia, which defines AI for Social Good workshop at NeurIPS (2019), Vancouver, Canada.
social good as “something that benefits the largest number of people in the largest possible way, such as clean air, clean water, healthcare and literacy” [29].

This lack of grounding principles manifests in computer science “for (social) good” projects spanning a wide range of political characters. For example, some work under this umbrella is explicitly developed to enhance police accountability and promote non-punitive alternatives to incarceration [4, 9], while other work uses data to predict and classify crimes to aid police investigations [41, 10]. That such politically disparate and conflicting work could be part of the same movement should prompt a reconsideration of the core terms and principles. When the movement encompasses everything, it stands for nothing.

The point is not that there exists a single optimal definition of “social good,” nor that every computer scientist should agree on one set of principles. Instead, there is a multiplicity of perspectives that must be openly acknowledged to surface debates about what “good” actually entails. Currently, however, the field lacks the language and perspective to sufficiently evaluate and debate differing visions of what is “good.” This allows computer scientists to make broad claims about solving social challenges while avoiding rigorous engagement with the social and political impacts.

USC’s Center for Artificial Intelligence in Society (CAIS) is emblematic of how computer science projects labeled as promoting “social good” can cause harm by wading into hotly contested political territory with a regressive perspective. One of the group’s projects involved deploying game theory and machine learning to predict and prevent behavior from “adversarial groups.” Although CAIS motivated the project by discussing “extremist organizations such as ISIS and Jabhat al-Nusra,” it quickly slipped into focusing on “criminal street gangs” [43]. In fact, the project’s only publication was a controversial paper that used neural networks to classify crimes in Los Angeles as gang-related [28, 41]. This conflation of gang members and terrorists echoes the language of “superpredators” used in the 1990s to justify harsh policing and sentencing practices [44] and is part of a long lineage of military ideas and practices being transferred to local police departments for use in poor and minority neighborhoods [3]. Moreover, the paper took for granted the legitimacy of the Los Angeles Police Department’s gang data—a notoriously biased type of data [17] from a police department that has a long history of abusing minorities in the name of gang suppression [45].

Whether or not the computer scientists behind this and similar projects recognize it, their decisions about what problems to work on, what data to use, and what solutions to propose involve normative stances that affect the distribution of power, status, and rights across society. They are, in other words, engaging in political activity. And although these efforts are intended to promote “social good,” that does not guarantee that everyone will consider such projects beneficial. Despite their label, projects like CAIS’ gang classification paper are many people’s version—most notably, the communities subject to gang-preventive police tactics—of a distinct and severe “bad.”

Most dangerously, while computer science’s vague framing of social good appears to result from a failure to recognize that such claims could be contested rather than from an explicit attempt to stifle these debates, this approach nonetheless allows those already in power to present their normative judgments about what is “good” as neutral facts that are difficult to challenge. Broad cultural conceptions of science as neutral entrench the perspectives of dominant social groups, who are the only ones entitled to legitimate claims of neutrality [11, 25, 26, 32, 34, 36]. Thus, if the field does not openly reflect on the assumptions and values that underlie essential aspects of computer science—such as identifying research questions, proposing solutions, and defining “good”—the assumptions and values of dominant groups will tend to win out. Projects that purport to enhance social good without a reflexive engagement with social and political context are likely to reproduce the exact forms of social oppression that many working towards “social good” seek to dismantle.

3 Incrementalist “good” can lead to long-term harm

Although efforts to promote “social good” can be productive [15], computer science has thus far not developed a rigorous methodology for considering the relationship between algorithmic interventions and long-term social impact. The field takes for granted that, even if machine learning cannot provide perfect solutions to social problems, it can nonetheless contribute to “good” by making many aspects of society better. In fact, some computer scientists emphasize these immediate improvements over

---

1Reflexivity refers to the practice of treating one’s own scientific inquiry as a subject of analysis [6].
long-term considerations: arguing, for example, that “we should not let the perfect be the enemy of the good” [42]. This position assumes that because we all agree that crime, poverty, discrimination, and so on are problems, we should applaud any attempts to alleviate those issues. This orientation to producing technical reforms treats the “perfect” as an unrealistic utopia that, on account of its impossibility of being realized, is not worth articulating or debating.

Pursuing social good without considering the long-term impacts can lead to great harm, however: what may seem good in an immediate, narrow sense can be actively harmful in a broader sense. In other words, the dichotomy between the idealized perfect and the incremental good is a false one: it is only through debating and refining our imagined conditions of the perfect society—an essential component of politics—that we can conceive of and evaluate potential incremental goods. Because there is a multiplicity of imagined perfects, which in turn suggest an even larger multiplicity of incremental goods, any incremental good must be evaluated based on what type of society it promotes in both the short and long term.

Evaluating the relationship between incremental goods and long-term social change is an essential task, for not all incremental reforms are made equal or push society down the same path. As social philosopher André Gorz proposes, we must distinguish between “reformist reforms” and “non-reformist reforms” [19]. “A reformist reform,” explains Gorz, “is one which subordinates its objectives to the criteria of rationality and practicability of a given system and policy.” A non-reformist reform, on the other hand, “is conceived not in terms of what is possible within the framework of a given system and administration, but in view of what should be made possible in terms of human needs and demands.” Because of the distinct ways that these two types of reforms are conceived, pursuing one or the other can lead to widely divergent social and political outcomes.

The solutions proposed by computer scientists are almost entirely reformist reforms. The standard logic of algorithmics—grounded in accuracy and efficiency [22, 33]—tends to require accepting and working within the parameters of existing systems to promote the achievement of their goals. Computer science interventions are therefore typically proposed to improve the performance of a system rather than to substantively alter it. And while these types of reforms can have value under the right conditions, such an ethos of reformist reforms is unequipped to identify and pursue the larger changes that are necessary across many social and political institutions (and may even serve to entrench and legitimize the status quo). When reform is conceived in this way, “only the most narrow parameters of change are possible and allowable” [35].

In this sense, the field’s current strategy of pursuing a reformist, incremental good resembles a greedy algorithm: at every point, the strategy is to make immediate improvements in the local vicinity of the status quo. But although a greedy strategy can be useful for simple problems, it is unreliable in complex search spaces: we may quickly find a local maximum, but will be stuck there, far from a broad terrain of better solutions. Computer scientists would never accept a greedy algorithm for complex optimization problems, and similarly should not accept a reformist strategy for complex political problems—where “the optimum solution demands ‘structural reforms’ which modify the relationship of forces, the redistribution of functions and powers, [and] new centers of democratic decision making” [19].

3.1 Case study: the dangers of “good” reforms in the criminal justice system

The U.S. criminal justice system, a domain where computer scientists are increasingly striving to do good, exemplifies the limits of a reformist mindset. The problem is that most technical efforts to contribute “good” are grounded in the existing logics of crime and safety. Even if they lead to incremental improvements, such reforms tend to reinforce and reproduce the criminal justice system’s structural racial violence [8, 20, 22, 39].

Because criminal justice reform can be “superficial and deceptive” [31], it is particularly important to couch reform efforts within a broader vision of long-term change. This is the emphasis articulated by the movement for prison abolition [16, 38]. Recognizing the violence inherent to confining people

2In many contexts, of course, it is not possible achieve perfect solutions through optimization techniques. But even in these settings, computer scientists approach the problem with rigor, developing and characterizing approximation algorithms. The same logic applies in political contexts, where the optimal solution is rarely achievable (if even definable): it is necessary to fully characterize the problem space and to evaluate how robustly and effectively differently approaches can lead to desired outcomes.
in cages and to controlling people’s lives through the threat or enactment of force, prison abolition aims to create a world without prisons. Notably, with this goal in mind, prison abolitionists object to reforms that “render criminal law administration more humane, but fail to substitute alternative institutions or approaches to realize social order maintenance goals” [37]. It is not enough that reforms produce immediate improvements to the criminal justice system; instead, only reforms that reduce or replace carceral responses to social disorder are pursued.

Pretrial risk assessments exemplify how computer scientists’ reformist reforms can make it harder to achieve structural social change. In response to the injustices of cash bail [13], groups including computer scientists [12], criminal defense organizations [19], U.S. senators [27], and state legislatures [40] have proposed replacing money bail with risk assessments that determine who should be detained before trial based on each defendant’s predicted likelihood to be rearrested before trial or fail to appear for trial. Yet such calls for an algorithmic reform overlook the ways in which seemingly “good” (and “fair”) criminal justice algorithms can reinforce carceral logics and outcomes, whether through legitimizing unjust policies [22], distorting deliberative processes [20], biased uses by practitioners [2] [14] [23], shifting control of governance toward unaccountable private actors [7] [30] [46], or allowing public officials to claim credit for embracing reform even as they ignore or squash more impactful alternatives [31]. Meanwhile, many activist groups and legal organizations are pursuing an entirely separate incremental, abolitionist, non-reformist, and non-technological reform: ending cash bail and pretrial detention.

Although adopting pretrial risk assessments and abolishing pretrial detention appear to respond to the same problem, they derive from conflicting visions of the “perfect.” One envisions a just world as one that includes pretrial detention, believing that the issue with pretrial detention is not that it is itself bad, merely that it is determined badly; accordingly, we should remedy the means by which people are selected for pretrial detention. Meanwhile, the other envisions a just world as one without pretrial detention; accordingly, we should abolish the practice altogether. Thus we see that the debate about risk assessments has little to do with technical matters such as fairness and accuracy or pragmatic considerations about the perfect versus the good, and instead hinges on normative questions about how the criminal justice system should be structured. It is only by articulating our imagined perfects that we can even recognize the underlying tension between these two incremental reforms, let alone properly debate which one to choose.

4 Conclusion

There is much more to be said, beyond the scope of this paper, about why computer science efforts to do good tend to have these attributes and how computer science can ground its interventions in rigorous theories of social change [21] [24]. There are numerous reforms that can help computer scientists reason well about doing good. If computer science is to productively contribute to creating a better society, it must develop a rigorous methodology that considers what it means to do good and how to choose among competing goods. This requires, first and foremost, a political orientation for algorithmic practice. Rather than referring to “social good,” computer scientists should more explicitly consider and articulate the normative commitments behind their work (whatever they may be). Second, in order to actualize these commitments, computer science needs a praxis that engages contextually with the relationship between technological interventions and social impact in both the short and long term. This requires looking to the lessons forged and debated by generations of social thinkers and activists regarding how to actually achieve positive social change. Such reasoning can help computer scientists consider the role of algorithms in improving society, how algorithms can generate unintended impacts when they interact with the social and political world, and when other forms of political action are necessary in conjunction with or instead of algorithms. Third, rather than presuming that algorithms provide an appropriate solution for every problem, the field must evaluate algorithmic interventions against alternative reforms. This also means finding new types of algorithmic interventions that better align with long-term pathways of social change. Many of these imperatives draw on the expertise of other fields, necessitating the need for an algorithmic practice that is interdisciplinary at its core, no longer prioritizing technical considerations (such as accuracy) as superior to or more essential than other forms of knowledge.

Acknowledgments

I thank Salomé Viljoen and the three anonymous reviewers for their constructive suggestions.
References


