European Research Council grant agreement n° 313173

Market-based Initiatives as Solutions to Techno-Scientific problems (MISTS) project

Algorithmic IF…THEN rules and the conditions and consequences of power

Daniel Neyland
Norma Möllers
Sociology, Goldsmiths, University of London
Sociology, Queen's University

2016

Corresponding author

d.neyland@gold.ac.uk

To cite this paper, a final version has been published as

Neyland D., Möllers N. (2016). Algorithmic IF … THEN rules and the conditions and consequences of power. *Information, Communication & Society*.
DOI: 10.1080/1369118X.2016.1156141

Algorithmic IF…THEN rules and the conditions and consequences of power

Daniel Neyland and Norma Möllers

Introduction

Power is a major theme in the recent literature on algorithms. Concerns refer, for example, to the roles of algorithms in processes of discrimination (Barocas and Selbst, forthcoming), the sorting and ordering of populations (Lyon 2003), or the (e)valuation and governance (Aneesh, 2009; Rouvroy, 2012) of social life. Furthermore, work on algorithms points to their relative inaccessibility, which renders both analysis and political intervention notoriously difficult. Although we share these concerns, we also find that scholarship on algorithms is often based on implicit, underlying assumptions that there is 'something special' about algorithms which makes them powerful. Lumped together in these assumptions are questions of the ontology, agency, and ability of algorithms. In this paper we seek to focus on how we can understand algorithms' agency and power.

What makes algorithms powerful entities? The way we pose this question already points to the argument we set out in this paper: by asking 'what makes algorithms powerful?' we question who and what need to be drawn together to yield effects that are recognizable as powerful. We argue that the 'social power' of algorithms, just like that of any other artefact, is an effect and not a cause of events, and which is not given but needs to be achieved (cf. Latour 1986, 2005). Rather than understanding algorithms as having power, an agency through which they create an effect, we argue that power derives from algorithmic association. By 'algorithmic association' we mean the assemblage of people, things, resources and other entities held together by practice and process. From this perspective, what algorithms do and how algorithms accomplish effects is inextricably tied to the situations in which they operate and which they help to reproduce.

We set out our argument drawing on two ethnographies of the development of algorithmic surveillance systems. These systems were designed to alert security personnel in transportation hubs (i.e. airports, train stations) in case of an undesirable event. Central to these systems were algorithmic IF…THEN rules which established the conditions (IF) and consequences (THEN) required to produce an effect (such as alert security personnel). However, in our field sites neither conditions, nor consequences were inherent to the system; both needed to be achieved through the careful plaiting of relatively unstable associations of people, things, processes, documents and resources. The continuous work needed to achieve conditions and consequences suggests that we cannot attribute 'power' solely to the algorithm as a single entity. Rather, the algorithms in our study operated through distributed agency among an array of people and things. This perspective shifts attention from purported impacts of algorithms on society to the variety of algorithmic associations that need to be established, as well as the conditions which render them more or less stable.

The structure of this paper is as follows. We begin by discussing how algorithms might be implicated in power relationships, drawing on the idea of 'associations' as a concept. We then introduce a study of two algorithmic surveillance systems in order to consider algorithmic rules, their design, development and possible effects. By analysing the conditions and consequences of these rules, we show the associations needed to achieve conditions and consequences. We conclude by suggesting that 'looking for trouble' – moments of breakdown, failure or other problems – is productive for emphasising the centrality of associations, rather than social power, to achieving algorithmic effects.

Algorithms and power

A number of authors have recently considered algorithmic power. For example, Lash (2007, p. 71) argues: "power is increasingly in the algorithm." Beer (2009, p. 994) further suggests: "algorithms have the capacity to shape social and cultural formations and impact directly on individual lives." Spring (2011) argues that algorithms trap individuals and control their lives, while Slavin (2011, n.p.) is clear that algorithms: "acquire the status of truth...They become real." Within these accounts of algorithmic activity, it is the algorithms which are the entities of concern; the algorithms are noted as powerful, agential and central to the distribution of consequences. These concerns lead to various calls (Kitchin, forthcoming; Seaver, 2013; Diakopolous, 2013) for algorithms to be held to account, governed and regulated. However, the challenge in accomplishing such a task is made complicated by proprietary interest in keeping algorithms enclosed, the technical difficulties involved in making an algorithm transparent (Slavin, 2011) or the problems involved in removing algorithms from their black boxes (Bucher, 2012).

This suggests two problematic issues: first, algorithms are agential, powerful and consequential; second, algorithms are almost impossible to know. However, several recent studies point toward possibilities for engaging up close with algorithms in order to explore their technicalities and consequences (as this paper will also do) and recent studies have begun to question some of the power attributed to algorithms. Hence, Drucker (2013: n.p.) argues that although algorithms are instructions for processes "whose outcomes may usually be predictable," algorithms can also be "as open to error and random uncertainties in their execution as they are to uncertain outcomes in their use." Diakopoulos (2013, p. 2) also suggests algorithms can be beneficial, but also involved in mistakes. This focus on algorithmic mistakes might not diminish the concerns we have with algorithms (their mistakes may be just as consequential and difficult to account for), but it does seem to modify somewhat the ways we might engage with algorithmic power. Instead of treating the algorithm as agential and consequential, we might also have to explore its limitations and the problems it causes for those people, organisations and activities which have become arrayed through algorithms.

Other work pushes these points further. For example, Kushner (2013, p. 1242) suggests that in the case of translation, algorithms need human "help" in order to become a "freelance translation machine, an assemblage of circuits and flesh that transforms text from one language to another with a computer's efficiency and the sensitivity of the human mind." Furthermore, Hallinan and Striphas (2014) argue, in their study of an online recommendation system, that the algorithm cannot work with various oddities in customer preference and instead of being resolved, these oddities need to be worked around. In these accounts, not only is the algorithm made accessible to research, it is also decentred in its agential consequences. The algorithm needs assistance.

Taken together with recent studies of algorithms in STS (Gillespie, 2011, 2014; Neyland, 2015) and geography (Kitchin, forthcoming), this suggests we need to understand the algorithm-in-action as situated (Suchman et al, 2002) among a variety of people, things, processes, documents, resources and technologies. It suggests we could use this situated-ness to explore a distinct approach to power which shifts attention away from the algorithm as *the* agential and consequential entity. Treating the algorithm as the agential entity requires an approach to power predicated on an asymmetrical distribution of the ability to create consequences for others. In this way algorithms would hold power over those subject to algorithmic decision making through this asymmetry. If we instead treat asymmetry as an achieved effect, we can explore how asymmetries are composed.

One means to do so would be to extend Latour's (2005) work on association into algorithms. Latour (2005) suggests that associations are forms of interaction through which things take a social shape. However, according to Latour, one should not jump from recognising the presence of an interaction, to considering that interaction is characterised by a social force (2005: 65). In this way, power (as social force) does not precede interaction, neither is it a property of ossified societal structures, nor is it a context "which makes the many participants in the action move," (2005, p. 83). Instead: "Power and domination have to be produced, made up, composed." (Latour, 2005, p. 64). Rather than treating power as resulting from an asymmetrical distribution of the ability to create a consequence for others, this line of argument suggests making sense of asymmetries through close study of the on-going associations through which an asymmetrical effect is achieved. This approach does not then deny that algorithms might participate in producing asymmetrical consequences, but instead explores how asymmetrical effects are achieved. To understand the ways in which algorithms are tied up with forms of power, we might thus explore their associations. If we follow Latour (2005), we would not look to explain power as an agential characteristic of algorithms, but instead seek to make sense of asymmetries composed through the associational life of algorithms. However, in recognising the situated character of algorithms, we need to provide precise detail on the nature of such algorithmic associations. In order to further explore the ways in which situated algorithms produce asymmetrical effects through their associations we will turn attention to our empirical study of the development of two algorithmic surveillance systems.

Algorithmic surveillance

This paper engages with two, 3 year-long projects which experimented with video analytic algorithms. Video analytics is a developing field in which algorithms and associated software/code sift through streams of digital video data, selecting out data that fits within prescribed patterns of relevance. Such patterns are often referred to as moments of ‘event detection’ in which algorithms and associated software/code demarcate relevant from irrelevant data and draw the relevant data to others’ attention. In our pursuit of algorithmic association and the means by which asymmetrical effects are achieved, this initial distinction of relevance and irrelevance is key – to be deemed ‘relevant’ rather than ‘irrelevant’ can often mean to be noted or not within a video analytic surveillance system. Yet to produce such an asymmetry and maintain the relevance-irrelevance demarcation requires continuing effort, as we will go on to explore.

Briefly stated, the two projects that we will consider were designed to work in the following way. [Author1]’s project involved a management consultancy firm as co-ordinators, a large technology firm, two teams of academic computer scientists and a team of social scientists given ethnographic access to the project. The project also involved a national European rail operator and a large city airport where the system would be developed and tested. An experimental aim of the research was to work through the possibilities of using algorithms to detect events such as people moving in the wrong direction (counter flow) through airport security or through entry and exit points of train stations, moving into inappropriate areas (intrusion) such as train tracks or closed airport offices, and abandoned luggage.

[Author 2]’s project involved behaviour analysis software for video surveillance systems. The project incorporated four teams of academic researchers (computer scientists, geoscientists, and legal scholars), two private research institutes (the members of which were mainly computer scientists by training), a consulting agency that carried out cost-benefit analyses, an IT company which was supposed to integrate the system for technology transfer, as well as officers from regional police crime units who were expected to share their expertise in detecting criminal behaviour. The group’s goal outlined in the grant proposal was to mechanize surveillance processes in order for the system to identify ‘dangerous’ situations and behaviour automatically and in real-time and send alerts to operators who would no longer have to watch screens at all times. The idea was to facilitate intervention before the fact, and would also reduce personnel cost through automation. The project was funded by the German government.

In the following sections we consider three areas of activity that provide insight into the situated character of algorithmic systems and enable us to explore more precisely the forms of association involved and how these go toward building asymmetries. We will begin by looking at the algorithms themselves and the means by which they establish conditions and consequences. We will then explore the further work required to achieve these conditions (through classifications and maps) and consequences (through bricolage and demonstrations).

Algorithmic IF…THEN rules

The algorithms for event detection used in video analytic systems are a designed product. They take effort and work and thought and often an amount of re-working. The algorithms establish a set of rules which are designed to contribute to demarcating relevant from irrelevant video data. In this way, such rules could be noted as central to the kinds of algorithmic power that generate asymmetries between people and things that can be ignored and people and things that might need further scrutiny. If such a focus could hold together, the rules would be central to the 'power' of algorithms. The following algorithmic rules were developed for detecting abandoned luggage in [Author1]'s project:

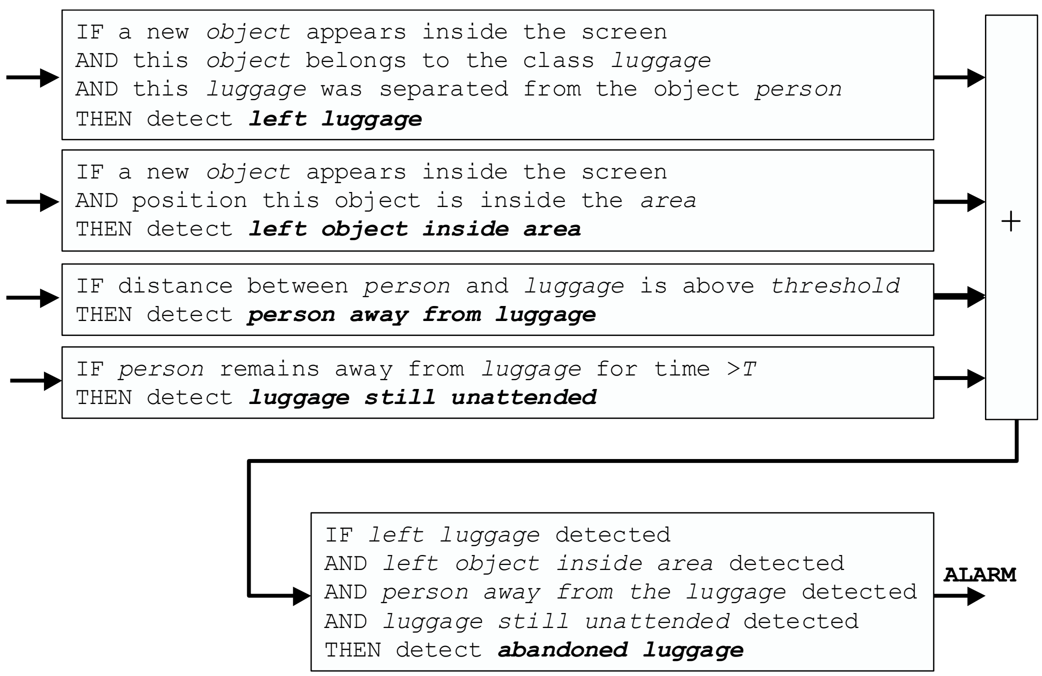

Figure 1: IF…THEN rules for abandoned luggage

What seems most apparent in these rules is the IF…THEN structure. At its simplest, the 'IF' acts as a condition and the 'THEN' acts as a consequence. In this particular algorithm the IF…THEN rules were designed to operate in the following way. IF an object was detected within an area (for example a train station or airport under surveillance), THEN the object could be tentatively allocated the category of potentially relevant. IF that same object was deemed to be in the class of objects 'luggage' THEN that object could be tentatively allocated the category of potentially relevant item of luggage. IF that same luggage shaped object was separate from a human shaped object, THEN it could maintain its position as potentially relevant. IF that same luggage shaped object and human shaped object were beyond the distance threshold currently set by the system (say 2 or 10 metres) and the same objects were beyond the temporal threshold currently set by the system (say 30 seconds or 1 minute) – that is, if the luggage and person were sufficiently far apart for sufficiently long, THEN an alarm could be sent to surveillance operators.

If this structuring and division of various entities (humans, luggage, time, space, relevance and irrelevance), occurred straightforwardly and endured, it might be tempting to argue that this is where the social power of algorithms is located or made apparent. A straightforward short-cut would be to argue that the algorithm structures the social world and through this kind of statement we could then find Lash's (2007) powerful algorithm, and Beer's (2009) algorithm which shapes the social world. As a result of an asymmetrical distribution of the ability to cause an effect, the algorithm would have social power. However, such a short-cut creates a leap from algorithmic rules to their consequences. If instead we pay attention to the situated work required for algorithmic conditions and consequences to be achieved, what we find is not that the algorithm structures the social world. Instead, efforts are made (and fail, are re-made, re-worked and then sometimes fail again or work to a small extent) to constitute the conditions required for the structure or the structure gets re-written to fit new versions of the conditions. This continual re-writing and efforts to achieve conditions and consequences is axiomatic to computer science but is also where associations are made, and are made available for scrutiny. It is also where the asymmetry between relevant and irrelevant data is continuously in the process of being made. In the following analysis of algorithmic conditions and consequences we will explore the associational effort required to constitute a structure that gives asymmetrical effect to algorithmic rules.

Achieving Conditions

In the two projects we consider in this paper, for the IF conditions of an event detection algorithm to be achieved required co-ordinated work to bring together everyday competences (among surveillance operators and computer scientists), the creation of new entities (including lines of code), the further development of components (from algorithmic rules to new forms of classification) and the development of particular theories of the world in which the algorithms would operate (such as a train station or airport), but also subtle changes in that world. These continual associations were the basis for achieving conditions. In order to illustrate the complexity of achieving conditions, we will focus here on a comparison of the two projects' work on classification and maps.

Achieving Conditions – Classifications: Both projects depended on classification work to bring about a condition necessary for the algorithmic system to asymmetrically divide relevant data from irrelevant data. However, classification systems are necessarily incomplete – there is always something that 'falls through the cracks' (Bowker and Star, 2000). This incompleteness means that ambiguity about the objects and subjects to which algorithms refer is inserted into the development process. To varying degrees, developers then need to create workarounds to compensate for such incompleteness. In extreme cases, the world the algorithms are supposed to refer to has to be remodelled (and simplified) according to the availability of data.

In [Author1]'s project for an event detection algorithm to pick out, for example, an item of suspicious luggage and issue an alarm to operators, various classifications had to be made. Objects had to be classified, for example, as luggage shaped or human shaped and the states of those objects also required classification, for example, as moving or not moving. Object classification involved algorithmically sifting through the stream of digital data produced by the video surveillance system in the airport and train stations. Classification depended upon models that were built to parameterise potential objects. This involved establishing edges around what a human-shaped object was likely to be (in terms of height, width and so on). Other models then had to be built to parameterise other objects, such as luggage, cleaners' trolleys, sign posts and other non-permanent attributes of the settings under surveillance. The models relied on 200-point vector analysis to set in place what made up the edges of the object under consideration and then to which model those edges suggested the object belonged. This was designed to produce rapid, real-time classifications within the airport and train station.

Parameterisation was presented by the computer scientists as a form of classification that the developing algorithmic system could manage without using too much processing power (as it also had to take on other tasks) and without having to take too much time. In this way, parameterisation would act as an initial but indefinite basis for object classification that could be confirmed by surveillance operators when shown images of, for example, an apparently suspicious item of luggage.

In [Author1]'s project object classification did not just depend on parameterisation. Object tracking was required to ascertain the state of the objects being classified. To achieve the conditions established in the IF…THEN rules of Figure 1, the system had to identify that a potential item of luggage was no longer moving and its human owner had moved at least a certain distance from the luggage and for a certain time.

In order to track objects that had been given an initial and hesitant classification, human shaped objects and luggage shaped objects would be given a bounding box. This was a digitally imposed stream of meta-data that would create a box around the object according to its already established edges. The box would then be given a metadata identity according to its dimensions, location within the airport or train station (for example, which camera it appeared on) and its direction and velocity. IF a human shaped object split from a luggage shaped object, IF the human shaped object continued to move, IF the luggage shaped object remained stationary, IF the luggage shaped object and human shaped object were over a certain distance apart and IF the human shaped object and luggage shaped object stayed apart for a certain amount of time, THEN this would achieve the conditions under which the algorithmic system could issue an alert. In place of the algorithm (in the form of IF…THEN rules) having power, it seems instead that a series of people, things, actions and relations have to come about to provide a set of conditions through which an asymmetry can be composed, relevant data divided from irrelevant data and an event detected. In this sense, the algorithmic system comes to be what it is through a carefully maintained series of associations (Latour, 2005); it is the associations through which any sense of power might be said to operate and any asymmetry between relevant and irrelevant data achieved.

The importance of these associations can be made clear by comparing the forgoing analysis with classification in [Author2]'s project. Here, the focus was not specifically on objects and people, but rather the actions of people in relation to other people. The main event detection algorithm in [Author2]'s project detected patterns of aggregated movements across the monitored space through what the computer scientists termed 'unsupervised learning', analysing 'what most people do'. The underlying idea was the following: If a single individual behaves significantly differently from the majority of people in a given space, then there is an increased chance that this person is exhibiting the kind of behaviour the system is supposed to detect. In this theory, 'conformity' means 'what most people do,' and deviance is then designated as everything else. Classification would thus operate in the following manner: If a moving object was detected within the monitored area, and it could be assigned the class of person, and this person's movement trajectory diverged significantly from the movements of most people in this space, then the camera would zoom in on this person, sending an alarm and the live video feed to the surveillance operators' screens. In this second project, then, the algorithmic system is nothing without the associational effort to asymmetrically divide normality from abnormality.

Before one jumps too swiftly to assume that algorithms 'have' associational power and that this creates effects, it should be noted that this particular arrangement (using unsupervised learning to distinguish normal and abnormal behaviour) itself emerged from failure. Initial project aims involved classifying actions into 'dangerous' and 'safe' behaviour. Here the researchers had struggled with three interrelated problems: First, they recognized that social behaviour is indexical and can only be meaningfully understood in its context. For them, this meant that they were not able to teach the algorithm how to meaningfully interpret behaviour in order to make definitive decisions. The behaviour was situated. One example they frequently raised was how they would be able to tell whether converging movement trajectories (two people moving towards each other) meant that these people were engaging in, for example, illegal substance trade or simply having a friendly conversation. Second, the police officers in the project were not able to turn their implicit police knowledge (they had learned in years of training) into knowledge explicit enough to satisfy the software requirements for well-defined categories and rules (cf. Sacks, 1972; Goodwin, 1994). Third, because the researchers consequently had more discretion over the definition of 'dangerous' behaviour than they wished for, they did not want to assume responsibility for potentially making wrong decisions. In this sense, to assume that the algorithms 'have' power is to overlook the entangled associations of people, things, decisions, processes and resources within which the algorithm is situated and from which effects are noted (or not), and, from which moment, changes are made (or not). As one researcher noted:

> "You know, the rules don't fall from the sky, someone has to specify them. Ideally, the person who runs the transportation hub puts in his knowledge. But really, the core of the problem is this: What is aberrant behaviour, and what is the concrete situation? … There is definitely a knowledge gap between theory and practice, the police and us. Experts are capable of certain classifications that you simply cannot imitate with technical means." ([Author2]'s field notes, September 2011)

What we can note here is that the associations are not a straightforward condition to be achieved – there are various ways in which algorithmic rules and software/code might be written, various aims put forward and changed, various early test results that require changes, different claims to expertise that require consideration and expectations of expertise that require revision. How different associations are made (between different people and objects, between assumptions about what a stream of video data does tell us or can tell us about suspicion or normalcy), operates in conjunction with a continual reconsideration of the conditions to be achieved. The conditions to be achieved will change with the associations. In this sense it is too simplistic to suggest that algorithms 'have' power; any accomplishment of effect emerges from continual reconfigurations of the entities involved. It is an occasional accomplishment of effect (in other words, the system manages to do something) that can lead to either a conformation or a re-writing of the conditions to be achieved. Asymmetries, such as demarcations of relevant from irrelevant data, can then also change as an upshot of transformations in the conditions achieved through the work of the project participants.

Achieving Conditions – Mapping: The contrast between achieving a condition and maintaining it as part of the algorithmic system and not achieving a condition and so re-writing the basis for what ought to count as an effect, can be seen in comparisons between the two projects' efforts to map space. In [Author1]'s project, in order for an object to be classified (as in the preceding analysis), first the very notion of a moving object had to be identified and to do this, the computer scientists looked to use a standard technique in video analytics: background subtraction. This method for identifying moving objects was somewhat time consuming and processor-intensive, but these efforts could be 'front-loaded' prior to any active work completed by the system. 'Front-loading' in this instance meant that a great deal of work would be done to produce an extensive map of the fixed attributes of the setting (airport or train station) prior to attempts at classification work. Mapping the fixed attributes would not then need to be repeated unless changes were made to the setting (such changes included in this project a change to a shop front and a change to the lay-out of the airport security entry point). Producing the map provided a basis to inform the algorithmic system what to ignore. Fixed attributes were thus nominally collated as non-suspicious in ways that people and luggage, for example, could not be, as these latter objects could not become part of the map of attributes (the maps were produced based on empty airports and train stations). Having a fixed map then formed the background from which other entities could be subtracted. Anything that the system detected that was not part of the map, would be given an initial pixel mask which could then feed into aforementioned processes of object classification and tracking.

In contrast to this approach to mapping, in [Author2]'s project, the teams of researchers were not able to implement their software in the fixed environment needed to make maps. This was because the group worked out of different organizations distributed all over Germany, and only met every three months to integrate their work; and because they did not cooperate with a transportation hub which would put the infrastructure of a monitored space at their disposal. This meant that all of the technical equipment had to be moved and reconfigured from lab to lab, and to the spaces in which it was to be tested and presented. The many 'moving parts' in this project highlight the efforts of communication and collaboration and the careful combination of people, things, processes and resources, which needed to be aligned to achieve the conditions for the algorithms to 'do' anything.

The basis for demarcating relevance from irrelevance in both projects was thus distributed between various different entities (computer scientists and their understanding of spaces such as airports, maps that might be programmed to ignore for a time certain classes of objects, classification systems that might then also – if successful – provide a hesitant basis for selecting out potentially relevant objects). Here again it becomes clear that anything that algorithms were able to 'do' was situated in continual reconfigurations of the entities involved, and that the agency of algorithms thus needs to be understood as distributed. The 'power' of the algorithmic system to asymmetrically divide relevant data from irrelevant data was an upshot of these associations and not a pre-condition for achieving effects.

What we suggest is that an understanding is required of the achievement of conditions in order to make sense of algorithms. This approach is important for emphasising the situated efforts required to design a set of algorithmic rules and achieve conditions for their operation (through, for example, classification and mapping). We have also tried to show the ways in which algorithms are part of quite precarious arrangements at times, with the very conditions to be achieved subject to re-writing according to the effects a system manages to show or not show. By considering the effects of an algorithmic system as an upshot of its associations and by emphasising the work required to make and maintain and also change those associations, we have attempted to move away from any assumption that it is the algorithm itself that is in some way powerful. Our suggestion is that asymmetrical effects – such as demarcations between relevance and irrelevance – are achieved (if at all) through associations, rather than through a pre-condition characteristic of algorithms which gives them the power to act independently on people and things. In the next section, we will explore this further through the second part of algorithmic IF…THEN rules, moving from conditions to consequences.

Achieving Consequences

Although we have somewhat separated algorithmic conditions and consequences in this paper, we would like to stress that in practice achieving conditions is inseparable from achieving consequences. Even a cursory look at Figure 1 will reveal that various IF-conditions and THEN-consequences are nested within the single set of algorithmic rules. Moreover, the consequences of the algorithmic system (for example, issuing an alarm to operators of the system) cannot be achieved without the conditions (making the maps, classifying objects), but the conditions are also meaningless without the consequences (the maps are made to aid in the issue of alarms and problems with the conditions may lead to a re-writing of anticipated consequences). What is clear in the two projects, is that achieving consequences through algorithmic systems involved distinct types of activity in comparison to achieving conditions. We will compare two examples of trouble in achieving consequences across the two algorithm projects to highlight two different forms of activity that occur in response. Assessing moments where the projects ran into trouble is particularly useful as these are where detailed considerations of associations, their strengths and failures are made apparent to everyone in the project. Running into trouble provides a basis for considering the fragility of such associations. Trouble was also, as we will show, a continual matter of concern for the participants in these two algorithm projects.

In [Author1]'s project, the main consequence to be achieved was to produce alerts for system operators of such matters as abandoned luggage. The computer scientists sought to test out if the new algorithmic system could produce such alerts more effectively, or at least as effectively, as the conventional video surveillance system in the end user sites (train stations and an airport). Taking the example of abandoned luggage at the airport, operators would conventionally scan the monitors in their control room in the airport and seek out items which appeared out of place. Items would then be given some scrutiny and if they appeared to be luggage which had been abandoned, operators would radio through to security on the terminal floor who would move to inspect the item. Operators suggested, on average, they would detect one such object per hour. This set a benchmark for the new algorithmic system. The computer scientists set up the system to run for 6 hours taking a live feed from the airport cameras and expected to discover around 6 objects. The system was left to run for 6 hours in the airport as a redundant system (conventional surveillance would continue to operate and provide the basis for any necessary interventions). The computer scientists were present in the airport to collect and analyse results. They were interested in the number of correct alerts issued, but also the number of false positives (seeing things that were not there) and false negatives (not seeing things that were there). In the 6 hours that the system ran, in place of detecting approximately 6 items of potentially lost or abandoned luggage, the algorithmic system detected 2654 potentially suspicious items.

The working assumption of the computer scientists was that there were likely to be around 2648 false positives. In later checking of a random sample of alerts, it turned out the system was detecting as abandoned luggage such things as reflective surfaces, sections of wall, a couple embracing and a person studying a departure board. The computer scientists worked through the results and identified ‘confounding’ variables such as camera angles not favoured by the system (for example being too low and hence unable to see through crowds), changing lighting conditions (which cast shadows in different directions, changing the system’s view of the edges of objects, their initial classification but also their tracking between frames and cameras), and different flooring materials in different parts of the airport (which meant that certain objects stood out against the background in different ways in different spaces). Running into trouble in this way was particularly significant for the project team, as at the time of this test they were only weeks away from giving a final demonstration of the technology ‘live’ to the project funders. The carefully plaited associations between various entities, which each made demands on the system to asymmetrically divide relevant from irrelevant footage, were now to be made available for reconsideration.

This kind of trouble was not unusual in algorithm projects. [Author2]’s project ran into a number of issues. For example, having no single fixed location in which to demonstrate the technology (such as a transport hub) and having to transport technology to the relevant location on each occasion when it would be demonstrated to the funding institution and industry posed significant difficulties for the project. Every single part of the system had to be reconfigured to each other and to the physical space in which they wanted to demonstrate the technology – associations were thus made and re-made frequently and at pace, as this note from [Author2]’s field work shows:

> Dennis approaches me and asks me what I'm up to. "I just wanted to see the work on the cameras, and Marco just explained to me that they're configuring the cameras." Dennis nods and explains: "Yeah, their problem is that the cameras vibrate. I think they always work with fixed cameras and you don't have that problem with those." I don't understand. "Well, we only were allowed to attach the cameras with strings, and in order not to damage the columns we put foam in between; but that's not helping the problem, either. And they're not able to calculate the errors out". ([Author2]'s field notes, November 2011)

In the above case, vibrating cameras meant that the tracking algorithms mistook their own movement for movement in the monitored space, and this in turn meant that it would track every single pixel in the image. In a quite literal sense no asymmetry could be drawn between relevant and irrelevant data as every pixel was constituted as potentially in need of further scrutiny. Further trouble emerged when construction workers changed the pattern of the floor tiles in the days preceding a project demonstration (see below), which was a serious problem for synchronizing the cameras with the other system components. Finally, other events in the hall during the preceding weeks repeatedly changed the appearance of the monitored space which meant that producing the maps needed to reduce processing resources was continuously interrupted. In other words, the architecture, construction workers, other events in the hall, and a variety of other factors intervened with the functioning of the algorithmic system. Associations were made radically contingent by the continual uncertainties posed by moving components from one location to another; as the vibrating cameras showed, components and thus their ability to associate, were transformed by changing locations. Encountering these kinds of trouble caused problems for the project's attempt to demonstrate the utility of the technology to the funders and a wider audience:

> Everyone is gathered around the big touch screen in the hall. Other than the researchers, there are industry representatives, police officers, security services companies, and the funding institution representatives present. Dennis introduces the "use case": he explains that someone will steal the painting they put in the middle of the hall, and that the system will send an alarm to the screen when that happens. This would be based on the analysis of the video material. "So who wants to steal the painting? Jakob?" Before Jakob can answer, a very excited looking Professor Bode jumps up, raises her hand, and walks towards the painting. Everyone's heads turn back and forth between her and the screen, looks of anticipation on their faces. The professor finally reaches out to the painting and takes it from its easel. Everyone's attention shifts to the screen. Anticipation turns into extreme awkwardness when nothing happens. The researchers look at the screen with horror. Dennis tries to salvage the situation with some self-

> deprecating humour: "I forgot to say that this is also an opportunity to see work in progress" ([Author2]'s field notes, November 2011)

In a similar manner to the computer scientists in [Author1]'s project who had to post-rationalise failure in their abandoned luggage test, here the computer scientists had to work to come up with a means to narrate the collapse of associations. They suggested that tiny errors in the messaging formats – such as a missing hyphen – could cause malfunction as the analysis algorithm would send a message to the graphical user interface which could not be read. Both projects, then, faced problems in achieving the kinds of consequences they had initially anticipated – both primarily focused on issuing alerts to operators.

We can note across both projects two common types of response to trouble. A first type of response can be most clearly illustrated through [Author2]'s project. This response to trouble occurred both in their everyday work, as well as on occasions of demonstrations, and involved continual tinkering with code. A kind of on-going bricolage (MacKenzie and Pardo-Guerra, 2014) took place through which ideas of what the surveillance system was supposed to achieve were translated into rules and commands that could be read and executed by a computer. And when trouble emerged in achieving consequences, computer scientists would search for bits of code in open source libraries, piece these together and modify them, test them, and if they did not work the first time, look for the source of error. This was dull, routine work, often not welcomed by the computer scientists.[1] It was also irritatingly complicated for the computer scientists; as the code ran to hundreds of thousands of lines and was drawn together on occasions from existing non-proprietary sources which were not fully understood by the computer scientists, searching for errors was very time-consuming. Associations within the hundreds of thousands of lines of code had to be dragged to attention by computer scientists, inspected and passed or altered. The following event illustrates a typical situation of what happened when this kind of trouble was encountered:

> I ask Marco if they also thought to include parameters such as age and gender. He declines, but says that he's working on such algorithms in a different project. He gets really excited and offers to demonstrate the algorithm. "And how does it work?" I ask. I'm interested whether the algorithm is based on machine learning, or classifications made by developers. "Well the computer learns what features to look for" he replies. He moves his laptop to a different desk and connects to one of the PTZ cameras in the room. He runs the program and a window pops up on the screen, showing the camera footage of Marco and me looking at his laptop screen. Marco adjusts the camera so my face is centred on the

[1] In [Author2]'s project not every computer scientist understood coding as part of their work, and troubleshooting, debugging, and tinkering often were seen as nuisances. Such differential valuation of tasks in developing algorithmic systems is a topic which deserves investigation in its own right.

> screen. He points to numbers and letters which appear next to my face, and to my great amusement the algorithm tells us that I'm a man in my mid-40s [despite being female and in my mid-twenties]. While I'm still laughing, I see that Marco isn't so amused. He opens the window with the code and starts looking through the lines of code, mumbling "hmmm...but this looks pretty good...why doesn't it work... I can comment this line out, although it's nice, but...". He crouches behind his laptop, starts typing and mumbling to himself and the screen, and stops paying attention to me. ([Author2]'s field notes, May 2011)

This kind of routine response to trouble involved continual iterations of translating consequences into code, by piecing together bits of software the computer scientists took from open source libraries, fitting them together, modifying them to their needs, testing the software, looking for bugs, troubleshooting, and then doing it all over again. Rather than understanding coding as a straightforward, planned, and throughout intentional activity, we need to understand it as importantly including tinkering, troubleshooting, debugging, and finding workarounds – a kind of bricolage (MacKenzie and Pardo-Guerra, 2014) through which associations between entities are continually reconfigured and the nature of entities also called to consideration, prior to any consequences at some point being accomplished.

A second type of response to trouble can be illustrated through [Author 1]'s project. The project team looked to respond to the problems met in attempting to identify abandoned luggage 'live' in the airport. The disappointing results of these tests might appear to undermine assertions regarding the 'power' of the algorithm to achieve much at all. However, within a few weeks of the abandoned luggage tests being carried out, the algorithmic system was successfully demonstrated 'live' in the airport to research funders. The response to trouble in this project did not focus so much on bricolage as what we might term the production of a contained effect (drawing inspiration from the work of Muniesa and Callon, 2007). A laboratory, according to Muniesa and Callon (2007), creates a set of controlled and contained conditions idealised for the demonstration of a specific effect. Any attempt to achieve this same effect in moving out into the world beyond the laboratory would require extending the controlled conditions of the laboratory out into the world. Within the controlled and contained conditions of the laboratory, the associations would be at their purest, with no confounding variables. To transform a busy airport into such a controlled environment in order to demonstrate an algorithmic contained effect would also require that the airport in a small, contained way would become the algorithmic system's purified laboratory.

Making the system work, creating a contained effect, required careful management of the different components and their associations. The 'confounding variables' of crowds, distinct flooring materials and inconsistent lighting that had been held responsible for the previous poor results, would have to be managed away from the system in transforming the airport into the laboratory. The airport was monitored by the project co-ordinators with advice from the airport's security manager to ascertain when there were fewest crowds. If fewer people moved through the airport there was a reduced chance that objects the system needed to parameterise, classify and track (such as luggage shaped objects) would be occluded (for example, by multiple legs walking between the camera and object). Also, if the abandoned luggage object were to be initially tracked in the moments prior to abandonment between certain locations and cameras where the lighting conditions and flooring conditions were consistent (for example, in those areas towards the rear of the terminal away from natural light where the floors were covered with non-reflective grey tiles) this might 'idealise' conditions for a test. If the person shaped object 'abandoning' the luggage shaped object could also leave the luggage in a location where it was not occluded from cameras (by, for example, a pillar) and then the person shaped object were to also leave via a route with consistent lighting conditions and flooring materials, this would further 'idealise' the airport in order that it might become the algorithmic system's laboratory. Finally, if the luggage shaped object being abandoned fitted a particular set of parameters, this might aid the system in accomplishing object classification as it would 'know' what it was looking for.

The response to trouble was thus to produce a contained effect by contriving a set of laboratory conditions for showing research funders that the system 'worked' – at least within a narrowly bounded set of associations that could endure within one part of the airport and for one moment. The research funders were made aware that an item of luggage was unlikely to be abandoned in the airport at precisely the time they were present for a demonstration and elements of the 'abandonment' would be staged. However, this abandonment might still appear somewhat genuine – or less contained – if the algorithmic system itself did not 'know' of the artifice of laboratory conditions. If the system was sifting through the streams of digital video data, using maps to perform background subtraction, object classification through parameterisation and tracking to issue an alarm for abandoned luggage – in other words, if the system was operating within the set of associations initially marked out by the computer scientists – and while operating within these associations, the system could pick up the item of abandoned luggage, this might still demonstrate the system's effectiveness in demarcating relevance from irrelevance. But knowledge of the abandonment was not so unevenly distributed between funders, human project participants and the algorithmic system.

Producing an 'idealised' contained space by filtering out confounding variables as far as possible, provided a prior basis for background subtraction, parameterisation and then subsequent classification and tracking. In this way, the associations of the algorithmic system could endure as the system 'knew' what it was looking for, where and when and what it ought to do next. As a result, the 'abandoned' luggage was identified by the system during the demonstration; this success was the result of the contained effect. The project team did not need to reproduce the contained effect in the world beyond the 'idealised' conditions of this particular space and time of the airport. Instead, the possibilities of achieving the same effect in the world outside the laboratory-airport only needed to be pointed towards.[2] Much of this pointing, involved critiquing existing airport terminals for their inadequate architecture (such as low ceilings), poor lighting and flooring, and outdated cameras systems. In other words, any future failure of the algorithmic system's associations to endure beyond the contained effect achieved in this particular airport, could be explained through the absence of conditions which favoured the algorithmic system. For the algorithm to demonstrate and consistently produce an asymmetrical effect by distinguishing relevant and irrelevant video data, required the rest of the world (or at least those locations where the system would be used) to match those conditions of the contained effect that would enable its associations to endure.

Conclusion

We have argued in this paper for the importance of moving away from considering algorithms as having social power in the sense that the algorithm itself would be noted as the agential character in the drama, able to cause an effect on society. Instead, we have suggested it is necessary to recognise the situated character of algorithmic systems. That is, algorithmic systems come to make sense through their situated-ness, wherein distinct components are designed and re-worked and come together with rules, people, processes and specific kinds of relationships. We have sought to extend existing work on algorithmic systems (Gillespie, 2011; Neyland, 2015) by drawing on the work of Latour (2005), to explore a particular approach to power. This has enabled us to move from a treatment of algorithms as having power derived from an asymmetrical distribution of the ability to act independently on others, to exploring asymmetries as an achievement and 'power' as constituted through associations. Such an approach does not entail abandoning the idea that consequences follow from the introduction of algorithmic systems. But it does mean, that, if we want to understand how algorithms are implicated in power relationships, we need to cast a wider net and explore the heterogeneous practices and materials, going everywhere from labs, companies and control rooms to transportation hubs.

We were particularly interested in the ways in which algorithmic systems became involved in the production of an asymmetric effect, distinguishing relevant from irrelevant data. Our purpose in turning to associations is to direct attention to the ways in which such effects as the production of asymmetries are made and endure (or collapse) through the composition (and decomposition) of associations.

[2] Future research could explore the questions raised by the on-going operation of algorithmic systems beyond these moments of demonstration. For example, what form does trouble take in systems that are accepted as 'working'?

The algorithmic IF…THEN rules of the projects analysed in this paper provided a basis for considering particular types of associations and the effects that emerged. In particular we explored the IF-conditions of classification and mapping and the THEN-consequences and trouble that ensued. We suggested that both classification and mapping were on-going, quite precarious achievements that resulted from the associating of a number of different kinds of entities and distributions of roles and responsibilities which followed from the articulation of conditions that also either confirmed or led to a re-writing of those conditions. The trouble that occurred for both projects in trying to achieve consequences – namely the issue of an alert to surveillance system operators as a product of successfully producing an asymmetrical distribution between relevant and irrelevant data – was instructive for shedding further light on the nature of algorithmic associations. Both projects were characterised by an on-going failure of association which led to similar responses.

First, project members responded to trouble through a kind of on-going bricolage. In [Author 2]'s project, failure to recognise expected features of the world (the theft of a painting and [Author2]'s age and gender), led to bricolage in the form of searches in software libraries for pre-existing solutions, efforts to stop cameras vibrating, dredging up lines of code from within the system, and everything from external confounding variables to hyphens in lines of code were made available for re-inspection. Similar tinkering took place in [Author 1]'s project in trying to repair classifications. Second, project members also responded to trouble – for example, a failure to detect abandoned luggage or dangerous behaviour – by producing a contained effect. In [Author1]'s project, a small section of the airport where the demonstration would take place was transformed into a controlled, laboratory-like space in which confounding variables could be manipulated away from the algorithmic system as a basis for demonstrating that effects could be achieved (albeit under very purified conditions). In [Author2]'s project the same was true for the University hall in which they demonstrated the technology a second time.

What we would like to suggest is that this focus on association, while de-centring the algorithm as *the* cause of effects, also points up what we might term the associational dependencies required for the algorithmic system to operate. The need to continually re-work the bases for doing classification and mapping, the purification of conditions required to demonstrate the contained effect and the bricolage necessary for re-working failed lines of code, all suggest particular kinds of associational dependence. Classification could not happen without maps or stable cameras or pre-defined parameters, maps could not happen without a stable location or co-ordinated effort to continually work out the permanent attributes of a setting, contained effects could not be demonstrated without purification and the exclusion of confounding variables and bricolage could not take place without computer scientists, their training and open source libraries. Our suggestion is that future studies of algorithms pay further attention to these associational dependencies – it is through making and holding together these dependencies that algorithmic effects are achieved.

References

Aneesh, A. (2009) Global Labor: Algocratic Modes of Organization. *Sociological Theory* 27(4): 347-370.

Barocas, S and Selbst, AD (forthcoming): Big Data's disparate impact. *California Law Review*, available from: http://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID2615343_code1328346.pdf?abstractid=2477899&mirid=1 (last accessed: 18th July, 2015).

Beer, D. (2009) Power through the algorithm? Participatory web cultures and the technological unconscious. *New Media & Society* 11(6): 985-1002.

Bowker, G. and Star, SL (2000) *Sorting Things Out*. Cambridge, MA, USA: MIT Press.

Bucher, M. (2012) Want to be on top? Algorithmic power and the threat of invisibility on Facebook. *New Media and Society* 14(7): 1164-80.

Diakopoulos, N. (2013) Algorithmic Accountability Reporting: On the investigation of black boxes, available from: http://towcenter.org/wp-content/uploads/2014/02/78524_Tow-Center-Report-WEB-1.pdf (accessed: 21st October, 2014)

Drucker, J. (2013) 'Performative Materiality and Theoretical Approaches to Interface.' *Digital Humanities Quarterly* 7(1): n,p.

Gillespie, T. (2014) The relevance of algorithms. In Tarleton Gillespie, Pablo J. Boczkowski, Kirsten A. Foot (Eds.): *Media technologies. Essays on communication, materiality, and society*. Cambridge: The MIT Press, pp. 167–193.

Gillespie, T. (2011) 'Can an algorithm be wrong? Twitter Trends, the specter of censorship, and our faith in the algorithms around us,' *Culture Digitally*, available from: http://culturedigitally.org/2011/10/can-an-algorithm-be-wrong/ (last accessed 17th December, 2015)

Goodwin, C. (1994) Professional Vision. *American Anthropologist*, 96(3): 606-33.

Hallinan, B. and Striphas, T. (2014) Recommended for you: The Netflix prize and the production of algorithmic culture. *New Media and Society*, pp: 1-21.

Kitchin, R. (forthcoming)

Kushner, S. (2013) The freelance translation machine: Algorithmic culture and the invisible industry. *New Media and Society* 15(8): 1241-58.

Lash, S. (2007) Power After Hegemony. *Theory, Culture and Society* 24(3): 55-78.

Latour, B. (2005) *Reassembling the Social*. Oxford: Oxford University Press.

Latour, Bruno (1986): The Powers of Association. In John Law (Ed.): *Power, action and belief. A new sociology of knowledge?* London: Routledge, pp. 264–280.

Lyon, David (2003). Surveillance as Social Sorting: Computer Codes and Mobile Bodies, in David Lyon (ed.), *Surveillance as Social Sorting: Privacy, Risk, and Digital Discrimination*. New York: Routledge.

MacKenzie, D. and Pardo-Guerra, JP (2014) Insurgent Capitalism: Island, Bricolage and the Re-Making of Finance. *Economy and Society*, 43(2):153-182.

Muniesa, F. and Callon, M. (2007) Economic Experiments and the Construction of Markets, in D, MacKenzie, F. Muniesa and L. Siu (eds) *Do Economists Make Markets?* Princeton University Press, Oxford, pp. 163-88.

Neyland, D. (2015) Bearing Account-able Witness to the Ethical Algorithmic System, *Science, Technology and Human Values*, 41(1): 50-76.

Sacks, H. (1972) Notes on police assessment of moral character, in Sudnow, D. (ed) *Studies In Social Interaction* (Free Press, New York, USA), pp. 280–93.

Seaver, N. (2014) Knowing algorithms. Paper presented at: Media in Translation 8, Cambridge, MA, USA, April, 2013.

Slavin, Kevin. 2011. How algorithms shape our world. available from: http://www.ted.com/talks/kevin_slavin_how_algorithms_shape_our_world.html (last accessed: 14th February, 2014)

Spring, T. (2011) How Google, Facebook and Amazon run the internet. Available from: http://www.pcadvisor.co.uk/features/internet/3304956/how- google-facebook-andamazonrun-the-internet/ (last accessed: 14th February, 2014)

Suchman, L., Randall, T. and Blomberg, J. (2002) Working Artefacts: Ethnomethods of the prototype. *British Journal of Sociology* 53(2): 164-79.