A Machine Learning View on Profiling

Martijn van Otterlo (*m.vanotterlo@donders.ru.nl*)
Cognitive Artificial Intelligence
Radboud University Nijmegen, The Netherlands

Draft – appeared at CPDT11 - do not cite

1. Introduction

"To the right and to the left as in mirrors, to the right and to the left through the glass walls I see others like myself, other rooms like my own, other clothes like my own, movements like mine, duplicated thousands of times. This invigorates me; I see myself as a part of an enormous, vigorous, united body; and what precise beauty!" **(Zamyatin, 1924, pp.31-32)**

Our current information age is shifting the concept of *privacy* day by day. Privacy may eventually be a futile notion, drowned in an ocean of *big data* (Bollier and Firestone, 2010). The main eroding forces on privacy include the amount of information that is available, the data hunger of public and private institutions, the (perceived) threat of terrorist attacks and the corresponding business opportunities of (security) companies, and the rise of *smart algorithms* (Nilsson, 2010) to automatically gather, store, analyze, and utilize enormous amounts of data [1].

When it comes to big data and privacy, we can find examples in every newspaper on an almost daily basis. In the Netherlands, current debates and plans are about i) a central database for medical records, ii) a GPS-based tracking system that will be required in every car to make differentiated use and payment possible, iii) a chip card system which will record and track every individual making use of public transportation, and iv) *smart energy meters* that will monitor our demand for energy. These systems can deliver useful information to optimize various services but at the same time they disclose a lot of information concerning habits, preferences and whereabouts of individuals. Other concrete systems include v) surveillance cameras extended with microphones in Amsterdam, vi) automatic face recognition in the Rotterdam metro system, vii) body scanners at Schiphol airport, and viii) the addition of our biometric data (e.g. fingerprints) to our passports. All these systems are tacitly introduced into our societies and generate huge amounts of private information of individuals.

When people talk about negative aspects of privacy they often employ terms as *"Big Brother"* and *"Orwellian"*, to connect to the idea of an all-seeing eye overseeing every bit of information we (have to) reveal about ourselves. Sometimes we are aware of this modern electronic version of Bentham's panopticum and consciously think about the fact that we are being watched continuously, and sometimes act accordingly. In most occasions, however, privacy is seen as attached only to the information we reveal, or have to give up, about ourselves to governments, companies and other parties. Our concerns are often solely about what personal information, i.e. attached to a specific identity, is out there, who gets to see it, who owns it, and how that information is used to control, track, monitor

[1] There has been considerable debate about the characteristics of our information age, in which smart algorithms and large amounts of data are ubiquitous. Various concepts have come up, and their properties and consequences for interaction, privacy and human factors have been discussed. These include *ubiquitous computing, pervasive computing, ambient intelligence, smart environments* and so. In this article it suffices to think of any environment where large amounts of information about user's behaviors are being measured, stored, analyzed and exploited to build general models about users and their behavior. The crucial factor is that all these operations on this data are *automated* through the use of intelligent computer algorithms. See also (Brey, 2005)

or even manipulate us. Many studies concerning the legal aspects of such information have been conducted, and countermeasures exist in the form of *privacy preserving* algorithms.

Attaching data and information to specific individuals is related to *identification* and *control*, and it is here where the connection to Orwell's 1984 [2] and its *Telescreens* is appropriate. However, in the context of big data, the concept of *profiling* makes it possible to go beyond the individual level (i.e. identity, cf. de Vries, 2010), and track, monitor, measure and manipulate various groups of individuals. Profiling amounts to building (statistical) models from large amounts of data from many individuals, after which the profiles themselves can be exploited to derive novel information about particular individuals. For example, even though I may not know the writer Murakami, a quick look at my book purchases on the internet may reveal that it is very likely that I will appreciate his writing. This aspect of privacy comes much closer to the dystopian novel *We* by Zamyatin (see the opening quote in this article) than to Orwell's book. Indeed, one can imagine walking through one of the streets Zamyatin describes, filled with glass houses, and observing many citizens at the same time, seeing correlated behavioral patterns, in other words: building profiles. Extending the metaphor; even if I would – as an individual – replace all glass in my own house by wood (i.e. protect my data) it would still be possible to build profiles of all my neighbors and derive information about me for exploitation, manipulation or control by third parties.

Profiling is enabled by progress in faster computers, the availability of data, the Internet, and especially smart *algorithms* to process, understand and employ the data. These algorithms stem from the field of *artificial intelligence* (AI), and include *machine learning* (ML) and *datamining* (DM). Many people do have experience with some of these techniques, for example when Amazon or online music stores make recommendations based on previous purchases or click-data from different users. Online stores can use their customer data to "guess" that you may like a particular product you have never seen before. Privacy as in *who-can-see-what-information* has always been an issue[3], yet the broader possibilities and implications of smart algorithms and profiling have not yet been well studied, especially not from the viewpoint of privacy (but see Hildebrandt and Gutwirth, 2008c). In order to facilitate such studies into such *ambient law* (Hildebrandt and Koops, 2010), and encourage a debate on the implications of profiling, we will describe some of the main characteristics of data science and smart algorithms, with a light emphasis on *probabilistic* approaches.

The **outline** of this article is as follows. In the next section we first explain the characteristics of profiling and how it's related to identification. The main section starts after that and deals basically with "all one can do with large amounts of data". We discuss three main approaches in data science and how they interconnect. The last part of this article deals with several selected topics and trends in data science which are particularly interesting in developments concerning privacy.

2. Data, Behavior, Persons and Profiles

We first have to consider several important notions. Here we look upon possible sources and forms of data about users and their behavior, and we discuss what profiles are and how these relate to (virtual) persons and identity.

2.1 Elements of Profiling

In order to understand ML approaches to profiling, we need to distinguish *data* and *behavior* on the one hand, and *persons* and *profiles* on the other.

[2] Even the ownership of electronically purchased books can be unclear. An ironic example was recently given by Amazon, who removed the electronic version of Orwell's 1984 from customer's e-readers without their permission.

[3] See for example the concerns about the so-called *database society* (Warner and Stone, 1970) forty years ago.

2.1.1 DATA AND BEHAVIORS

Nowadays there is much data available, and it comes from various sources. Put very simply, governments want lots of data for security and control, companies want to make money, and scientists want interesting data. Furthermore, people like to share lots of information on social networks, list sites, blogs, etc. Computers, and especially the internet, enables the generation and storage of massive amounts of data *in electronic form* which makes size and location relatively meaningless. In contrast, decades ago scientists, companies and governments were relying on manual labor[4] and annotation to obtain useful data. A notorious example is the East-German security agency *Stasi* with a large part of the population physically gathering physical information about citizens[5] Especially companies like Google and Facebook have greatly expanded their automated data gathering efforts[6]. A lot of data comes from people leaving electronic traces when they perform ordinary daily activities, because they either use services provided by companies or they make use of government-provided services such as utilizing a passport, or traveling through electronic transportation. Companies harvest data mainly because they can use it to approach individuals with targeted advertisements[7] and governments are much concerned with surveillance and terrorist's threats[8].

In science, data has always been important for experiments. But also many other efforts to obtain additional data to aid in the process of profiling and understanding. For example, in the 80's people like Douglas Lenat[9] started to formalize *commonsense knowledge* and create huge databases with useful knowledge to reason about ordinary things, like *a dog has four legs*, *a cat too*, and *both cats and dogs are mammals*. Recent ontologies, databases such as OpenMind[10] and the well-known Wikipedia are all efforts in this direction, and of which knowledge can be drawn for ML purposes. Other recent efforts use entertaining *games* to collect data. For example, Luis Von Ahn created a series of games in which users had to interact with each other and where the playing behavior provided the necessary labels for pictures[11]. Yet another example is the *Mechanical Turk* [12] initiative where scientists can post data to be annotated by humans for profit. All these initiatives generate *meaningful* data through human labor. The generation and exchange of such knowledge by robots only is explored in various projects, for example the RoboEarth project[13]. But above all, much data is supplied and maintained by people themselves, in the form of personal pages on social networks, life blogging and making lists, Flickr photo accounts, general blogs, chat rooms and Skyping and many more.

With **data** we consider all pieces of information lying around in electronic databases, on the internet and so on. However, in the context of profiling, we want talk more specifically about **user data** and **user behavior**. User data is any set of information pieces that can be attributed to the *same en-*

[4] See the hilarious movie *Kitchen stories* (http://www.imdb.com/title/tt0323872/) in which researchers physically sit many hours in real kitchens to observe how people move around it and use it.

[5] See the interesting movie *Das Leben der Anderen* (http://www.imdb.com/title/tt0405094/).

[6] Services on the internet are most often not as free as they seem; in the end you pay with information, which can be transformed into cash through profiling (Sobiesk and Conti, 2007). See also Microsoft giving up users' privacy for profit, in Wall Street Journal August 2nd 2010
(http://online.wsj.com/article/SB10001424052748703467304575383530439838568.html)

[7] See for an extrapolated example in the future the movie *Minority report* (http://www.imdb.com/title/tt0181689/)

[8] Many movies play with these concepts, see for example *EyeBorgs* (http://www.imdb.com/title/tt1043844/), 1984 (http://www.imdb.com/title/tt0087803/) and *Enemy of the state* (http://www.imdb.com/title/tt0120660/)

[9] See the work on Cyc (Lenat *et al.*, 1990), with recent versions for commercial and academic use.

[10] http://www.openmind.org/

[11] In the *Peekaboom* game, the goal is to obtain labeled images, i.e. labeled with the concept appearing on it. One player is *peek* and the other is *boom*. Boom starts with an image and a word, and can reveal parts of the image, and peek can guess the word (boom can indicate whether cold or warm). There is an incentive to get more points by only revealing the necessary part of the image, see (von Ahn and Blum, 2006)

[12] https://www.mturk.com/mturk/welcome

[13] RoboEarth is a World Wide Web for robots: a giant network and database repository where robots can share information and learn from each other about their behavior and their environment (http://www.roboearth.org/)

tity. This can be, for example, the properties of a network account, or the combined facts filled in at some webpage. User behavior is defined similarly, but now the pieces of information are *actions*, for example the path walked in front of a surveillance camera, or the click behavior on some webpage. In many cases it is not necessary to distinguish between actions and facts, and we can combine all information into one **record**. de Vries (2010) calls these informational *shibboleths*, a term coined by Derrida[14]. Crucial here is that the information *belongs to each other* (i.e. to one (virtual) identity.

2.1.2 Persons and Profiles

The main dichotomy that is important in this article, and which will be defined more precisely in a latter section, is that between data at an *individual level* and that at an *aggregated level*. The first deals with personal data (of any kind) of a specific individual, consisting of individual pieces of information. The second level is the profiling level, consisting of (statistical) models which model correlations between pieces of information appearing in the individual's data, causal patterns and general rules that apply to a subset of the individuals; for example, that people *who have bought X and Y are more likely to buy product Z if they have already looked at product W*. Profiles provide means to infer knowledge about an individual that is not actually observed[15].

The crucial difference between the two levels is that the information at the individual level is actually observed, i.e. it is factual knowledge. Knowledge at the profiling level is not usually available to the individual user, and often not observed. Instead, the profile is *applied to* the individual user to infer additional facts, preferences, or assumed intentions (e.g. to buy some product). Thus, we aim mostly at what Hildebrandt (2008) calls *non-distributive* profiles, for example a checklist for psychopaths in which the profile lists properties which will not hold for all people who will most *likely* belong to that profile. An interesting fact is that in order to apply a profile to an individual, one generally does not need to identify the individual with a real person; just some observed facts may be enough to predict what is needed (see Neudecker, 2010, for interesting examples). On the other hand, identification is a useful way to *link* different sets of facts to the same individual, thereby extending the information about a certain virtual person[16]

Hiding information yourself will often not work, since the provider of the service (e.g. Facebook) owns all data anyway, and there are many ways additional information about you gets revealed by other users, applications and so on. Even social security numbers provide the means to infer additional information through statistical analysis (see Acquisti and Gross, 2009). In one famous case, the internet provider AOL released (in 2006) data of twenty million search keywords for over 650,000 users, and this data was anonymized. However, users often provide much information about themselves in their search queries. *The New York Times* successfully discovered the identity of several searchers, and exposed user number 4417749 as Thelma Arnold, a 62-year-old Georgian widow [17]. In some cases, privacy-preserving techniques can, and have been, applied in the context of automated datamining, but this is no safeguard against profiling.

[14] Her *algorithmically constructed* shibboleths will get a more technical meaning in Section 3.

[15] *"Profiles discovered by these techniques may not be anticipated, desired or politically correct but modern algorithms do no (yet) have the power to care"* (Anrig *et al.*, 2008).

[16] This is why lately many services at social networks are connected to many other sites; to increase the knowledge about each virtual person. In fact, quite interestingly, the more power people think they get about the information they disclose, the eager they are in sharing even more information (see the *illusion of control* principle) (Brandimarte *et al.*, 2009).

[17] See (Barbaro and Zeller Jr., 2006) and http://en.wikipedia.org/wiki/AOL_search_data_scandal

2.2 Automated Profiling: Activity Recognition

Profiling[18] is the formation of general models on the basis of data from a number of individual users. The *automated way* version of this amounts to *machine learning* and *datamining*. In addition to information available about people in databases and on the internet, a great amount of additional *sensors* is available for increased data gathering and behavior monitoring. These sensors include RFID tags, video cameras (e.g. CCTV), GPS signals and mobile phones. A common feature of these devices is that they are widely available in our daily life, as opposed to specialized or expensive sensors such as ultrasonic sensors or brain scanners. A subfield of AI particularly interested in learning profiles from such rich (sensor) data is **activity recognition** (Yang, 2009). Here we view activity recognition as the main research area subsuming profiling. Its aims are to *interpret the sensor readings of humans (or moving things) and tell us in high-level, human understandable terms what is going on.*

The last years, there is a growing interest in privacy aspects of such automated procedures for profiling and activity recognition[20]. Quoting one of the main investigators in that area: *"Profiling is a matter of pattern recognition, which is comparable to categorisation, generalisation and stereotyping. To understand what is new about profiling we must differentiate between classification (ex ante categorisation) and clustering (ex post categorisation) ..."* (Hildebrandt and Gutwirth, 2008a, chap.17). Classification approaches (and related techniques) usually take the stance of a classical ML setting in which some function is learned from data. The *ex post categorization* approaches are more aimed at finding corresponding (behavioral) patterns among users, finding interesting new knowledge, and clusters of people who share certain features. In the latter, the feedback (or guidance) is less related to *what* we want to learn (e.g. a classification), but more related to *for what purpose* (e.g. what are the interesting characteristics of people who are likely to buy my product). Some pointers to the field are about activity *learning* (Turaga *et al.*, 2008), activity *mining* (Cao, 2006), *tracking* (Xiang and Gong, 2006), and *databases* (Choudhury *et al.*, 2006)

[18] Modeling and predicting behavior solely based on externally observable features resembles the *behaviorism* movement in psychology in the first half of the twentieth century with famous names such as Watson and Skinner. Quoting the latter; *"A much simpler solution [simpler than equating the mind with the brain] is to identify the mind with the [physical] person. Human thought is human behaviour. The history of human thought is what people have said and done."* (Skinner, 1976)[19]. Behaviorists felt that because it was absolutely impossible to look inside someone's brain to find out *why* some behavior comes about, they argued that psychology should be only concerned with *externally observable features of behavior*. Stipulating higher-order motivations, reasons, goals, intentions and things like that are merely speculation about the reasons for certain behavior. Whereas humans could possibly *talk about* their motivations, and perform *introspection* on what was on their mind (as *cognitive* psychologists would do) is not the right way to do repeatable, falsifiable science, especially with animals which are not capable of verbal explanations or introspection. The problems Hartz Søraker and Brey (2007) see are only partially valid, since the success of search engines such as Google and many other successful applications in activity recognition, show that much can be done by "just" correlation patterns. These successes are described in the famous Wired issue of 2008 on *"The Petabyte age"* (*Wired* July 2008)., the 2010 report on *"Big Data"* (Bollier and Firestone, 2010) and several special issues of magazines such as *The Economist* (e.g. *The Data Deluge, (and how to handle it; a 14-page special report)*. The Economist Feb27-Mar05 2010.

The behaviorist movement in psychology is mirrored in AI by movements such as *behavior-based* approaches (Brooks, 1991), *new* AI and *embodied* AI Pfeifer and Scheier (1999), and also *reinforcement learning* (Sutton and Barto, 1998). All these directions place a particular emphasis on *not representing* or *reasoning* about *intentions*, *goals*, *motivations*, or *inner states*, but instead focus on how predictions and behaviors directly follow from data. This is unlike many other trends in AI where the goal is – in contrary – to model *cognitive belief states, intentions, rich internal structures* and so on (cf. Langley, 2006; Langley *et al.*, 2009). It is good to keep in mind that many of the current techniques in data science which we describe here can in principle be applied in both contexts, although we lean slightly towards the behaviorist approaches. The advent of *semantic* techniques is a recent trend, and many researchers focus on such issues since they promise transparency and transfer of knowledge across domains, and some examples in the context of the web can be found in (Torre, 2009).

[20] See more on this in the book by Hildebrandt and Gutwirth (2008c), more specifically the chapters by Hildebrandt (2008), (Hildebrandt and Gutwirth, 2008b) and (Anrig *et al.*, 2008).

2.2.1 Applications and Domain Examples

There are many examples where automated learning has been applied to learn models of users and their behavior and we can only scratch the surface of what is going on. Applications of behavior modeling, recognition and prediction can be found in video surveillance, robot control, social networks, physical movement tracking, online shopping, computer games and many more[21] Profiling behaviors was studied early on in AI. For example, *opponent modeling* approaches for games have been very successful in Chess[22]. Recent progress in games like Poker demonstrates that such models can be learnt from databases of games played. Furthermore, they can also be used to categorize players into *cautious* or *aggressive* bidders.

Well-known profiling applications are webshops and search engines. Most people will have seen the recommendations of books or music, or web advertisements based on search behavior. People are often quite surprised how accurate they can be, and the occasional blunder will be rationalized afterwards[23]. Many systems are web-based (Perkowitz *et al.*, 2004; Torre, 2009) and especially social networks are a gold mine for activity recognition efforts. Here, an interesting approach based on photo collections is described by Singla *et al.* (2008). They insert many kinds of general *rules* into a system that tries to classify each person appearing in the photos into *spouse*, *friend*, etc. Examples of (commonsense) rules are i) *parents are older than their children* and ii) *spouses have opposite gender*. So-called *soft* rules are rules that will only sometimes be true, e.g. iii) *children and their friends are of similar age* and iv) *friends and relatives are clustered*. By training a system with annotated data, the authors found out that rules such as *if person A and B are friends, they are about the same age* are found to be highly likely by the system.

Many approaches nowadays are based on *video* as input. The increasing number of cameras in public and private spaces create a huge demand, and current *computer vision* technology is increasingly capable of recognizing, tracking and monitoring the video streams (Le *et al.*, 2010). In addition to monitoring CCTV cameras, for example in metro stations, other applications can be found. For example, Francois Bremond recently investigated such techniques in elderly care in the *GerHome* project (Zouba *et al.*, 2009). A replica of a home for the elderly was built, full of cameras and other sensors. The goal is to build automated systems that can learn from inhabitant's behaviors, and detect, for example, when things go wrong. For example, *usually when people move from the kitchen to the living room, they will sit on a chair, but if mister X is staying in between – not in a standing position – then it is likely that something is wrong.* Another vision-based approach was reported by Damen and Hogg (2009) who learned typical patterns of when a bicycle was being stolen versus it being picked off by the right owner. Vision-based techniques are obviously useful, but technically very difficult due to the complexity of recognizing things and people on real-world vision data.

Other kinds of approaches that will become increasingly important are based on *location-based sensors* such as GPS. Applications such as Google Maps can already make use of your location to recommend a good pizza place in your immediate vicinity. These features can also be used to learn typical behavioral patterns. Liao *et al.* (2005) learned to extract and label a person's activities and significant places from GPS traces. Behavior profiles were learned from complex relations between readings, activities and significant places. The system was capable of highly accurately labeling certain areas on the map as being related to *work*, *sleep*, *leisure*, *visiting picking-up car* etc.

[21] See (Krueger *et al.*, 2007; Turaga *et al.*, 2008) for surveys.

[22] The famous defeat of Gary Kasparov by the IBM Chess Computer *Deep Blue* was largely based on tuning the playing style against him (Schaeffer and Plaat, 1997)

[23] For example, searching for a new style file for the text editing system LaTeX with "latex style" can trigger advertisements for stylish rubber clothes.

3. Automated Computational Methods for Profiling

In this section we describe some of the main conceptual and technical issues involved in profiling. We distinguish first between *representations* of data and profiles and afterwards highlight several characteristics of how these can be employed by *algorithms* for inference, learning and explanations.

3.1 The Field of Machine Learning

Machine learning is a *"branch of AI that seeks to develop computer systems that improve their performance automatically with experience"*, taking inspiration from diverse fields such as computer science, mathematics, statistics, optimization and psychology and many others. The terms *pattern recognition*, *machine learning*, *data mining* and *optimization* are often used for techniques in this area (such as classification, clustering, regression and probabilistic modeling), but very often not consistently. Here we use the general term *machine learning*, and take it to stand for any methodology and set of techniques that can employ data to come up with novel patterns and knowledge, and generate *models* (e.g. profiles) that can be used for effective predictions about the data.

Much of modern ML was developed in the last two decades. A classic textbook is (Mitchell, 1997) and a more recent book with the same scope is (Alpaydin, 2004). The more recent *relational* setting (see Section X) is described in (Džeroski and Lavrac, 2001; De Raedt, 2008). Current approaches are firmly grounded in *statistics* (Hastie *et al.*, 2001) and *probabilistic models* (Pearl, 1988), more generally *probabilistic graphical models* (Koller and Friedman, 2009). *Reinforcement learning* (see later sections) is discussed in detail by van Otterlo (2009) and Sutton and Barto (1998). *Datamining* as a separate technique is treated in (Witten and Frank, 2005).

3.2 Representational Aspects

"You can only learn what you can represent": this means that every piece of information employed by a computer, or generated by it as a pattern or a model, should be *represented* in a formal *language*[24]. In ML, we distinguish between an *example language* and a *hypothesis language*. The first is used to represent the observations of the data, and the second is used to represent hypotheses about the data, or more generally, models (profiles).

3.2.1 Example Language

An example language consists of a set of **features**. Data about a single person consists of a set of such features with their **values**. Examples of such features are that the person with ID=246876 is *vegetarian*, *married*, is sitting on chair number 23D, and that he (another feature) is from the Netherlands. Each feature consists of the name of the item plus the value for this individual. The feature *vegetarian* is called *boolean* because it can only have one of the values *true* and *false*. A set of feature-value pairs can be seen as a *record* in some database, or equivalently, as a row or column in a spreadsheet program and is called an *example* in ML. This type of representation is called *propositional*, or *attribute-value*, and a *dataset* corresponds to a complete table in a database.

3.2.2 Hypothesis language

Whereas the example language describes individuals, and usually consist of factual (or, *observed*) data, the hypothesis language describes general patterns that hold for a group of individuals. In this article, we consider a profile as a construction in the hypothesis language, i.e. **a model**. There is a large variety of possible languages and later we treat some in more detail. A very generic and simple hypothesis language consists of the example language itself, extended with simple *logical connec-*

[24] See (Sowa, 1999) for a general introduction to knowledge representation.

tives. For example, we could have the following expression in the hypothesis language: (**A**):(*vegetarian=yes* AND *gender=female* AND *nationality=Dutch*). This expression models all female Dutch vegetarians.

More complex models could, for example, state that (**B**): *people from the Middle East "usually" do not eat pork meat*. This type of model requires more sophisticated expressions, in which we can also express *probabilistic* information. In such models, if we assume we only know about some individual X that he is married and from the Middle East, we could infer with some confidence that he will not eat pork meat – even though this is *not* factual knowledge (i.e. we have no explicit record stating that). In general we can say that the profile enables to make (probabilistic) **predictions** about an individual, backed by knowledge about the distribution in the population. Profiles can consist of *rules* or *correlations* about the relations between features, but they can also divide the group of individuals up into a number of subgroups the members of which share certain properties, or they can be complex functions that compute a value or class label based on some features.

The relation between models and examples is intuitively that *a model holds for an example*, e.g. model **A** holds for all female Dutch vegetarians. In technical terms this is called a *coverage relation*, and it entails that the model is *more general* than a specific example (e.g. it holds for other examples too). In the same way, models can be compared too in terms of *how general* they are. For example, if we drop *vegetarian=yes* from model **A** we get a more general model. Rules **A** and **B** above can be seen as *local* models, because presumably they will be part of larger models. In addition to generality, the *model complexity* is important; models can be complex (complicated interactions between features) or simple, and in general, models which are only *as complex as needed to model the data*[25] are preferred (see later: algorithmics).

3.2.3 Background knowledge

An additional aspect of ML is what we will call **background knowledge**. This type of knowledge includes "everything" that is available about a certain domain. This includes commonsense databases we have mentioned earlier. This type of knowledge can be used to answer more complex queries about a domain, or build more complex models. For example, a very simple piece of knowledge expressing the set of countries belonging to the Middle East (e.g. Egypt, Lebanon etc.) can help to build models in which the more general concept Middle-East (i.e. a new feature) can be used to group sets of observed features expressing only countries. Such forms of background knowledge form the basis of so-called **ontologies**, which relate pieces of knowledge to each other. For some models, the use of background knowledge is less supported (e.g. in neural networks), but for others, for example *relational* learning techniques, they are naturally incorporated. Lately considerable interest is developing in *probabilistic ontologies* (Poole *et al.*, 2009) which can be used in probabilistic (relational) learning techniques (De Raedt, 2008).

3.3 Models

One way to distinguish the many models appearing in ML is by looking at *what* they model. We come back here to the difference between the *ex ante* and *ex post* categorization (see page 5). Many ML models are based on ex ante categorization, for example classification and regression techniques. They are focused on mapping examples (e.g. individuals) onto an *a priori* defined set of *class labels*. Examples include *neural networks*, *support vector machines* and *decision trees*. These models are

[25] AI has developed very sophisticated hypothesis languages for modeling. However, research in subfields such as *artificial life* and related economic approaches show that extremely simple models *per individual* can create enormously complex global behavior of the population at large. This has been shown in synthetic psychology (Braitenberg, 1984), flocks of birds and sheep (Resnick, 1994), traffic patterns and computer generated art (Papert, 1980; Resnick, 1994) and various economic models (Epstein and Axtell, 1996). A fundamental difference with the profiles in this article is that we focus on how to model individual behaviors by learning from a large population of individuals, whereas the other models try to explain global phenomena by looking at the individual level.

often called *discriminative* models and they are used to learn a desired input-output mapping; for example to map people into distinct categories[26]. In contrast, as Hildebrandt (2008) describes, for profiling other types of models are more interesting. These fall into the *ex post* category. In ML terms, these include clustering techniques, but more generally models that *capture the data* in some way, and which highlight interesting patterns or correlation in that data. These include (association) rules and especially probabilistic models, such as Bayesian networks, which can represent causal and probabilistic patterns in the data. These models generally fall into the category of *descriptive* or *generative* models.

An important thing about models is that they usually have **qualitative** and **quantitative** parts. A qualitative part may express some (causal) relations between features, for example that a person's *length* feature depends on the *gender* feature. The quantitative aspect expresses *how strong* this dependency is. Most models have some form of *weights*, *parameters* or *probabilities*, and these quantities provide the flexibility of the (qualitative) model class.

3.3.1 Rules and Patterns

A simple form of models consists of *local patterns*, or *association rules*. In this case the hypothesis language gives rise to rules of the form: IF feature(1)=value(1) AND feature(2)=value(2) AND ...feature(n)=value(n) THEN feature(n+1)=value(n+1). This *rule* says that if certain features (1 to n) have specific values, then we can *infer* that another feature (n+1) has some specific value. In probabilistic models, we would say that feature(n+1) has value (n+1) with a probability p. Many rule learning approaches and datamining approaches are based on this type of local models. A similar device is the *frequent itemset* often employed in datamining contexts. A frequent itemset is a set of features with their value that occurs "often enough". For example, in shopping scenarios, it might be useful to know that people who buy *chips* will often also buy *beer* and *wine*. Frequent itemsets are dependent on a treshold (e.g. what exactly is "frequent"?) and can provide useful information about *common* patterns in examples. Both kinds of association rules and frequent itemsets are *local* models which are usually combined into larger structures (e.g. they serve in classification machines) or they provide the components of joint models such as Bayesian networks which will be discussed next.

3.3.2 Probabilistic Graphical Models: Bayesian Networks

Since profiling deals with uncertain data, unreliable sources of that data, and also uncertainty in what it can predict about users and their behavior, models that can explicitly deal with *probabilistic* aspects are desired. For example, in a webshop, lots of knowledge about customers may be available, yet most of it is not very certain, and one can only make predictions about *how likely* it is for someone to buy a product. One of the core models in contemporary AI and ML is the class of the *probabilistic graphical models*, (Koller and Friedman, 2009), which includes *Bayesian networks* (Pearl, 1988).

For profiling purposes, we are not so much interested in learning a specific classification function, but we are more interested in a general model of the data itself. The problem, however, is that a set of features gives rise to an enormous amount of different combinations of these features because of all the possible interdependencies among them. Estimating distinct probabilities for all these possibilities, so-called *joint probabilities* is hardly possible because it leads to huge models and long computations. In order to illustrate this, let us assume we have the following (boolean) features about an automobile engine[27]:

$p1$: The starter motor is ok.

[26] For more on these types of models, see (Anrig *et al.*, 2008) or any of the mentioned textbooks in this section.

[27] Many toy examples in graphical models are aimed at *diagnostic reasoning* about machines. The example here stems from (Nilsson, 2010). Another well-known example is the *burglary alarm* case by Pearl (1988).

Figure 1: *(left) Bayesian network, (right) dynamic Bayesian network (with CPTs)*

$p2$: The starter motor cranks the engine when the starter switch is turned on.

$p3$: The fuel system is ok.

$p4$: The car starts when the starter switch is turned on

These four are obviously related. We can assume that $p4$ is dependent on the other three and if we would know that it is false, it would tell us something about the other three. Now, the number of different feature-value combinations is $2^4 = 16$, because we can have 2 different values for each feature (e.g. if we take $\neg p1$ to stand for $p1$=*false*, then the combinations are $p1,p2,p3,p4$ or $\neg,p1,p2,p3,p4$ or $\neg p1,\neg p2,p3,p4$, etc). A full probability model would have to represent (and learn) these 16 values. If we expand the number of features to only 20, we already arrive at more than a million (2^{20}) different combinations, and this number grows very quickly[28]. In some small domains, such as this one, a domain expert might be able to specify all these 16 numbers, from which (with a little help from probability theory) one could compute probabilities for situations such as "the car starts and we only know that the fuel system is ok for sure". If, instead, we would assume that all features are *independent*, we would only have to remember the 4 probabilities $P(p1)$, $P(p2)$, $P(p3)$ and $P(p3)$, from which we could calculate the probability of e.g. $\neg p1,p2,p3,p4$ as $(1-P(p1)) \times P(p2) \times P(p3) \times P(p4)$. Now, a natural assumption is that in most common domains, the features will neither be completely independent, nor completely dependent on all other features, and we can assume the number of relevant probabilities to represent falls in between 4 and 16 in our domain. *Bayesian networks* are capable of making these dependencies (and independencies) explicit.

A **Bayesian network** consists of **nodes** which represent the features in our domain, and **arrows** which represent *direct influences among features*, but also indicate certain probabilistic *independencies* among them. The only restriction is that there can be no cycles in the model (i.e. no endless loops of arrows). The Bayesian network for our automobile domain is depicted in Figure 1(left). For example, the probability of $p4$ (car starts) does not depend on the probability of $p1$ (starter motor ok) *if* we already know (or we are given) $p2$ (car cranks) and $p3$ (fuel ok). In the model we can see this because there is no direct link between $p1$ and $p4$. Furthermore, if we know $p1$ then we do not possess more information about $p4$ if we already know $p2$ and $p3$. Technically, the probability of $p4$ is **conditionally independent** of $p1$, *given* $p4$'s parents[29], which are $p2$ and $p3$.

Now, instead of the full joint distribution of 16 probabilities, the network only contains 8 probabilities, and they are kept in local probability distributions at each node, called **conditional probability tables** (CPT). For example, since the probability of $p4$ depends on the features $p2$ and $p3$, the CPT in $p4$ consists of the values (where $P(X|Y)$ stands for *the probability of X given that we know that Y holds*) $P(p4|p2,p3)$, $P(p4|\neg p2,p3)$, $P(p4|p2,\neg p3)$ and $P(p4|\neg p2,\neg p3)$. For the node $p2$ we only need 2 values (the different states of $p1$) and for the nodes $p1$ and $p3$ we need only one value each

[28] And this becomes even worse for non-boolean features, for example *number of children*. If the number of values of the feature is n and the number of features is k then the number of combinations is n^k.

[29] A parent of a node N is a node which has an arrow pointing towards N, and the children of node N are all the nodes that have N as parent.

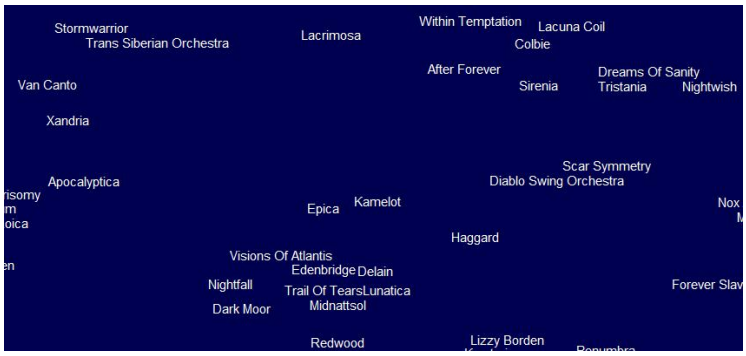

Figure 2: *A clustering of music artists.*

(the *unconditional* probability of being true, also called the *prior*). Using probability theory we can compute all 16 joint probabilities from just these 8 values. How these CPTs come about in the presence of data is discussed in the next sections. Coming back to the previous; for Bayesian networks, the *topology* of the network (nodes and connections) is the *qualitative* part, and the CPTs form the *quantitative* part. Bayesian networks are expressive enough to model *any* probability distribution over a given set of variables, which indicates their applicability and popularity. Another main reason for their widespread use is that efficient algorithms for inference and learning exist.

Bayesian networks form only the base model for the case where all features can – in principle – be observed and allow for various sorts of inference about the data. With the right algorithms, networks of considerable size (more than hundreds of nodes) can be used. Several extensions were developed in the literature, for example in the form of *Markov networks*, where the influences are modeled through *undirected* arrows, and which therefore allow cycles. An important class of extensions adds a **temporal** dimension. In many domains we can have features that are measured at different points in time. For example, we can have a feature visited(t) denoting that a user has visited our webpage at time t. Now, presumably, a visit at a next time (visited(t+1)) will depend on visited(t). Figure 1(right) shows an example of a *dynamic Bayesian network* in which the probability of raining is dependent on yesterday's weather, and the probability of bringing my umbrella today will depend only on today's weather. A third type of extensions add **unobserved** features to the network. For example, *hidden Markov Models* are well-known in *speech recognition*, where the observed features are the *phonemes* (i.e. sounds in the speech signal) and the unobserved features are the words which are spoken. Such networks model the probability distribution over a sequence of phonemes, *conditioned on* the word (which is not observed, but can only be inferred).

3.3.3 Distances, Kernels and Prototypes

Another type of model we want to discuss briefly are *instance based* and *distance based* methods. Instead of building an explicit model of the underlying probability distribution of the data, as with probabilistic graphical models, we can also look at the *similarity* between examples. The simplest way to do this, is to consider each set of features (an example) as a *vector* in n dimensions, where n is the number of features. Now, one can compute the standard *Euclidean distance* between two vectors, in the same way as one would compute a distance between points in a 3D space (i.e. between two three-dimensional feature vectors (X, Y, Z)). Such distances are used in any standard *clustering* technique. Using distance measures, the data itself is actually the model.

As an example, see the *Music Map*[30] in Figure 2, in which the band *Epica* is centered among bands that are similar in terms of the music they make. Based on this clustering of bands, I can find out about bands I possibly like too, even though I may not know them. Instance based learning (or, memory-based) learning uses the complete dataset and assigns a new example to the closest (or the n closest) example(s) that is (are) already in the dataset. In terms of profiling, I could find the closest individual in the dataset and most properties of that closest one will also apply to the new individual. Generated clusterings can also be used to decrease the dataset by only keeping a couple of *prototypes*, the cluster centers which are in some way typical representatives of some kind. Nowadays, more sophisticated approaches under the name of *kernels* exist (Schölkopf and Smola, 2002). Kernels are in essence distance functions, but go beyond that and are capable of modeling, and learning, similarities for very complex structures such as graphs, web pages, databases, images, sequences and many more.

3.3.4 A Note on Relational Knowledge Representation

So far, we have dealt with *propositional* knowledge representations and models. However, for many domains it is desired to have more powerful representations that can deal with explicit *relations between individuals* in models. This is useful for modeling social network relations, but also for example for pedigrees concerning blood types. In the latter, the blood type of one person is probabilistically dependent on the blood type of his or her parents. This constitutes a *relation* between features of one individual and those of others. *Relational representations* of knowledge for machine learning can make these aspects explicit. Hypothesis languages for these systems are based on *first-order logic* in which one can state things such as $\forall X \exists Y : \text{parent}(Y, X)$, meaning *for all persons X, there is a person Y who is the parent of X*, or *everybody has a parent*. These representations also enable new things like *collective classification*, in which, for example, one can classify people in a social network *in the context of how his or her friends are classified*, because of the explicit relations between this person and friends in the social network.

These representations are beyond the scope of this article, and we refer to (De Raedt, 2008) for pointers to the literature. They are important to mention because i) they are becoming widespread in AI (all discussed techniques such as patterns and kernels are applicable), ii) many ML systems use them (e.g. the examples on the photo categorization and the bike theft we mentioned earlier) and iii) most importantly, many of them are based on the exact same graphical models we discuss here.

3.4 Algorithmic Aspects

Representation is about which phenomena one can model; algorithms are all about how they come about. *Inductive methods* are the most typical ML setting, in which a dataset of examples is used to learn a model, i.e. to generate new knowledge. However, here we will discuss two additional types which together with inductive techniques form an integrated whole[31] for learning and profiling. Let us assume we have a dataset D of examples (each consisting of available features and their values) and a model class M of all possible models (where we assume that M extends to models where we employ any available background knowledge).

- **deduction** assumes the existence of a specific model m (from M) and computes statements (or their probability) that are entailed by m. These statements are about the general characteristics of the data D modeled by m. In addition, one can infer statements about a particular example d. Deduction is about *employing* profiles to derive information.

[30] This website features clusterings music artists that are "close" to one you search for (http://www.music-map.com/).

[31] One of the first scientists writing about the three together, and laying a foundation for many contemporary abductive approaches was C. S. Peirce (http://en.wikipedia.org/wiki/Charles_Sanders_Peirce).

- **induction** takes a dataset D and finds a specific model m in M which *best fits the data*. Induction is the core mechanism for *profiling*, and is about *obtaining* models from data.
- **abduction** takes a specific example e (from E) and a specific model m (from M) and computes an *explanation* (e.g. a set of *assumed* facts) which would enable to infer the example from the theory (with some certainty). Abduction supports *diagnostic reasoning* about individuals.

3.4.1 Deduction and Inference

Deductive inference assumes a model is present and amounts to *reasoning from causes to consequences*. The models in the above are like theories about a domain supporting these inferences. From *if A then B* and *A* one can *infer* that *B*. Deduction can generate "new" knowledge that is entailed by the model but which is maybe not not immediately present. For example, complex mathematical theorems are often a consequence of basic math, but it takes deductive inference to make that explicit.

Let us take a look at probabilistic inference. In the Bayesian network in Figure 1(right), if we know that it rains at time t, we can infer that the probability that it will rain at time $t+1$ is 0.7 and based on that, we can infer that the probability I will take my umbrella is $0.7 * .09 = 0.63$. Bayesian networks are typically used to compute a *posterior probability* of some features, *given* that one knows about the values of some other features. If in the same network, we know only that it rains at time $t+1$, we can infer that for my umbrella the probability of bringing is now 0.9 that it rained at time t is 0.49. Inference needs a *query* (e.g. a question) and a model, and it will determine whether the query is entailed by the model (possibly with a probability attached). In case we have several (local) models, their predictions have to be combined[32].

Since *exact* inference for large networks or theories can take long to compute, many efficient *approximations* exist that can guarantee a *good enough* answer in less time. Typically they do this by *sampling*. Instead of computing the probabilities using exact rules, they "try out" a number of values for features (in a smart way) and see what the probability of the feature you are interested in is in that sample set. More samples means more accurate answers. Approximation is also needed for efficient inference with huge (probabilistic) ontologies and commonsense knowledge databases.

Another example of probabilistic inference on a very simple model (profile) of users in a social network was given recently in Van den Broeck *et al.* (2010). The context here is marketing and the goal of the system was to find the best selection of people to send direct mail to. Rules were very simple and stated that i) *if I send an advertisement to person X he will buy my product with some probability*, ii) *if person X buys my product and person X and Y are friends, then Y will buy my product too with a certain probability*. Sending mails costs money, but selling products will result in receiving money. The system was capable of doing probabilistic inference on what would happen[33] (for thousands of people in the network) if we would send a certain selection of people a mailing. Thus, the query was about which people would (probably) buy the product, *given* that I send a mailing to certain people.

3.4.2 Induction, Learning and Mining

Learning (and mining) is essentially a **search process**. Given a space of models (generated by a choice for hypothesis language and background knowledge) the goal of learning is to find the "best" model for a given dataset. A **general-to-specific ordering** on the space of models can help to search starting from simple models to more complex models. A crucial component is a *goodness measure* that tells us how well a model fits the data. For example in the case of Bayesian networks, this comes down to comparing the distribution specified by the model, and that of the data. In order to make sure the

[32] See the article on a large-scale application with a computer playing *Jeopardy* in the *New York Times* ("What Is I.B.M.'s Watson?", June 14, 2010) where more than a dozen probabilistic algorithms are combined to make predictions.

[33] An additional, approximate, optimization step was added to actually find a best selection.

model can *generalize* to new data, usually the data is split into a **training set** and a **test set**. The first is used to find a model, and then the second (which is not used for learning the model) is used to *validate* how good the model actually is on data it has never seen before.

Learning can focus on the *parameters* (e.g. probabilities) of a *known* model structure (the quantitative part) or on the model structure itself (the qualitative part). We will discuss both in turn. First of all, learning the parameters is basically a matter of counting on "sample statistics". This means that, e.g. for $p4$'s CPT in the automobile example of Section 3.3.2, to estimate $P(p4|p2,p3)$ we can count in the data in how many examples $p4$ is true when $p2$ and $p3$ are both true. The network structure gives us the knowledge of which probabilities we actually have to compute from the data. For temporal models and especially models with hidden features things become more complicated, since some of the features are not really observed and we have to guess whether they were true or not. Still, relatively efficient techniques exist to compute these numbers from data.

Learning the qualitative part of models is more complex and deals with learning which feature interactions exist in the data. Common ML techniques go through the following steps:

1. **Start**. First an initial model is chosen. Based on a generality order, we would like to start with the simplest one, which for Bayesian networks is a topology with no arcs (so, just the nodes). Here this amounts to assuming all features are independent. For rule learners, or pattern miners, one might start with rules with an empty condition, or the pattern *true* (which applies to any example).

2. **Find out how good the current model is.** Based on a goodness measure, we can *score* the current model. For Bayesian networks, we can estimate (read: do parameter learning) how well it models the distribution in the data, and for rules or patterns, we can e.g. calculate to how many examples they apply.

3. **Find a possibly better model by changing the current one.** Given the current model, we can change small things to find an improved model. The model class determines what we can do. In Bayesian networks, we can add an arc between two features, delete one that is already there, or even swap nodes. For rules or patterns, we can add conditions to a rule, or delete them. Since there are several ways to modify the model, we try them all (or a selection) and estimate their goodness. The best modification is chosen and we continue from this model with the previous step and so on. This type of search is called hill-climbing, because it tries to improve the model until no further improvements are possible from that model.

Given any space of models, the search for the right model is much influenced by a *search bias* (how do we search? do we keep multiple candidate models?, by a *language bias* (which models do we "visit"? which modifications are allowed? can we add hidden features?) and the goodness measure. The last one relies on the deductive inferences we can make with our model. For Bayesian networks, there are many applications in which the network was successfully learned from data (see Koller and Friedman, 2009, for pointers).

3.4.3 Abduction, Assumptions and Explanations

Abduction (Josephson and Josephson, 1994) is something like the opposite of deduction, in that it *reasons from consequences to causes*. If we assume that *if A then B* and we observe *B* then we could *assume that A*. But if we also assume that *if C then B* then also *C* could be an *explanation* for the observation *B*. Abductive inference is *unsound*, which means that we do not know for sure whether *A* or *C* are true, but we can only *assume* that they are. Both *A* and *C* are *hypotheses* that we could take on to explain the observation *B*. Each of these two could be considered new knowledge the moment we accept them as being true (i.e. assume them). The problem remains: how to choose between them?

In some cases, we could choose the "simplest" one, in the same way as we can compare models using a generality order.

In probabilistic models, choosing the best one can be stated as finding the hypothesis with the highest probability of having *caused* the observation. For example, in speech recognition we are looking for the *word* with the highest probability of having caused a certain speech signal. Or, in activity recognition, we are looking for that *activity* for which the probability of causing the observed user behavior is the highest. In our automobile application, one possible hypothesis for $\neg p4$ (car does not start) is that $\neg p3$ (fuel system down). But another possibility is that $\neg p1$ (starter motor broken), and there are several other possibilities (e.g. they could be both broken). Finding the *most likely values* of $p1$, $p2$ and $p3$ given that $\neg p4$, is called the **maximum a posteriori** (or, MAP) hypothesis. In essence it is very similar to induction, for it searches for the best set of assumptions (i.e. modifications to the example) that will maximize a goodness measure (i.e. the likelihood of those assumptions causing the observation).

3.5 Summary

As we have seen, deduction is about deriving knowledge that is (probabilistically) entailed by the model, and both inductive and abductive techniques can deliver new knowledge; the former in the form of rules (or patterns) and the latter in the form of facts (features). Here we stress again that for most applications of ML, it is natural, and it makes sense, to integrate the three styles of reasoning to make the best out of the data.

As an example of a very successful application of such a combination, Ross King has worked for many years on the *robot scientist*, a physical robot capable of fully automatically doing real scientific experiments, by combining the deductive, abductive and inductive methods (King *et al.*, 2004, 2009). The domain is *bioinformatics* and the goal is to find out the function of yeast genes. The system is given knowledge about biological theories, and based on that, it applies abduction to come up with hypotheses to be tested. Deductive consequences of these can help in selecting them, based on the predicted costs of conducting them. Based on actual physical experiments (conducted by the robot *Adam* himself) and their outcomes, Adam learns new rules by applying inductive learning techniques, and has even made some new discoveries.

4. Computer Says No

Using computers for profiling, and controlling people in what they can or cannot do based on those profiles, has the potential to create circumstances in which the computer simply says "No"[34]. Below we highlight some caveats and recent directions in machine learning which are relevant.

4.1 Caveats of Statistical Models

"A man who travels a lot was concerned about the possibility of a bomb on board his plane. He determined the probability of this, found it to be low but not low enough for him, so now he always travels with a bomb in his suitcase. He reasons that the probability of two bombs being on board would be infinitesimal".(Paulos, 1988, p.21)
Humans are notoriously bad[35] with numbers and statistics but there is no guarantee if we use com-

[34]This is used in the British comedy series *Little Britain* in series of sketches involving Carol Beer. Carol's sketches revolve around her bored and unhelpful attitude to her customers. Whether in the bank, travel agency or hospital, she listlessly inputs customers' requests into her computer only to flatly reply with her catchphrase, "Computer says no..." This is usually followed by a single cough, directed at the customer, with no attempt to cover her mouth (http://en.wikipedia.org/wiki/Carol_Beer).

[35]A large number of books have been written about the topic. See the following texts for entertaining discussions: (Reichmann, 1961), (Tversky and Kahneman, 1981), (Paulos, 1988), (Gonick and Smith, 1993), (Best, 2001) and (Whyte, 2004).

	A	$\neg A$
B	0.5	0.3
$\neg B$	0.2	0.1

$P(A) = 0.7, P(B|A) = \frac{5}{7}, P(B|\neg A) = \frac{2}{7}$

$P(B) = 0.8, P(A|B) = \frac{5}{8}, P(A|\neg B) = \frac{3}{8}$

Figure 3: *Causal versus correlation: two ways of modeling the same distribution in a Bayesian network.*

puters to do profiling, things are settled. Since current ML is basically doing statistics with powerful knowledge representation, most of the general issues with statistics carry over to ML. This includes *sample size* (do I have enough data to make predictions?), *data set distribution* (does my target population have the "same" characteristics as the population I have trained my model on?) and *completeness of the model* (do storks really bring babies or is there an unmodeled factor explaining a possible correlation between births and stork populations?).

In the context of ML, due to the rich set of hypothesis languages and models, important additional aspects are the search and language biases, the goodness measure, but also whether inference is done exactly or approximately. All these factors determine what exactly one can expect from the final model. The rich class of models that ML supports also creates a *bias-variance trade-off* which intuitively says that one can either i) choose for very general models, but then the variance in the results of that model is large, or ii) one can choose more specific models (biases by preferences or prior knowledge) and get a lower variance, but maybe the model does not fit the underlying characteristics of data anymore. The search process for good models itself influences the final model, since usually there is no guarantee we get the best model possible, but only good in the "vicinity" of models we have visited during the search. For example, take a look at Figure 3. Both small networks model the same distribution that is depicted on the left, but differ in the connection structure. Since this can happen in such small models, one can imagine that small differences in how the search progresses can generate different networks in the end.

A last thing to keep in mind is that models are trained using a *global* goodness measure, i.e. the model is judged on how well it predicts *on average*. This means that predictions for any individual may be wrong; something to consider when it comes to profiling (what do we do if the model predicts the individual to have some property with only probability 0.6; do we act upon this?)

4.2 Comprehensibility, Transparency, Visualization, Automation

Instead of *privacy enhancing technologies*, Hildebrandt and Koops (2010) advocate *transparency enhancing technologies* (TET) to aid in developing ambient law. In the context of ML several developments are relevant for enabling TETs.

Comprehensibility of what is being learned has always been somewhat neglected in ML and only sometimes it is discussed explicitly (e.g. Džeroski and Lavrac, 2001, Sec 15.6.). In general it is accepted that e.g. neural networks are less comprehensible, and Bayesian networks and relational representations support *understanding* what has been learned in human understandable terms. But one has to be careful with such models too. For humans, huge sets of hundreds of rules are very hard to inspect visually, especially when their predictions are combined probabilistically in complex ways. Furthermore, humans have a built-in bias to interpret models in a *causal way*. Look again at the Figure 3. An arrow from A to B immediately triggers a causal interpretation that *A causes B*, but actually this arc only indicates a dependency, and we can also place the arrow the other way around. On the other hand, *if we stick to a causal model, we end up having to specify fewer numbers, and the numbers will often be easier to come up with*[36].

There many developments that make *data visualization* easier, and many new creative ways are

[36] (Russell and Norvig, 2003, p.498)

being developed to visualize data, and it will be useful to employ such techniques in the context of profiling too. In the ML community there have been efforts to create *benchmarks datasets*[37] to be able to compare algorithms with at least the same data. A very relevant development in the direction of TETs may come from *experiment databases* (see the recent PhD thesis by Vanschoren (2010)). The goal of such databases is to store, in addition to data itself, the used *algorithms, settings and parameters* and *results* of experiments in ML. The main goal is to be able to *compare* algorithms more carefully, and to not repeat experiments that have been done over and over again by many scientists. Recent efforts go into developing *query languages* to ask any question about experiments, or find *relevant* experiments and results in case we want to relate to our own experiments. A similar goal is reported in the work on the robot scientist we discussed earlier: by making *everything* explicit (data, hypotheses, algorithms, parameters and results) in formal languages, one can make the whole process of learning from data *transparent* and *reusable*. Similar efforts are reported in the robotics world (e.g. the mentioned roboearth project) where the goal is to make robot's knowledge learned on a task *transparent, understandable for both humans and robots*, and *reusable*.

4.3 Feedback, Active Learning, and Adaptive Behavior

So far, we have mainly dealt with the situation where there is *some* dataset, and we want to build a model out of it. However, if models function as profiles, presumably something is going to be *done* with the information drawn from the model (see also Hildebrandt, 2008, Sec. 2.4.4.). For example, if the model represents information about customers in a webshop, the model will be used to target specific users to receive advertisements, discounts, or even get rejected from accessing the store (e.g. for not paying on time). This, in turn, can trigger the population of customers to change – either customers coming and leaving, or by customers changing their behaviors, and this, in turn, may require to rebuild the model because it is outdated. In other words, profiling generates models, and actions based on these models will change the behavior of individuals, and because of that models will (have to) change, and this *feedback loop* will continue.

With respect to *acting* upon the data, the ML subfield of *reinforcement learning* (RL) (Sutton and Barto, 1998; van Otterlo, 2009) studies learning techniques with such feedback loops. A typical toy domain in RL is a *maze*. A *policy* represents the behavior of a robot moving through the maze, and based on this policy the robot will visit parts of the maze. Once it modifies its behavior (its policy), the parts of the maze the robot visits will change, and this again drives a new modification of the behavior and so on. An important characteristic of RL is a *reward function* which drives behavior modifications. Our maze robot will receive rewards for reaching the exit, and for using fewer steps. The feedback the robot gets about its "progress" in learning how to behave is scarce (i.e. it does not get specific instructions on how to get out, but only some feedback of how it is doing) and delayed (good initial steps will only be rewarded at the end when it exits the maze). Translating this to a profiling case in a webshop; the rewards can come from customers ordering products, and the actions are advertisements and so on. An *adaptive system* could learn a model of its customers (as discussed throughout this article) and based on that it could act[38], get some feedback and possibly modify its behavior based on that model. Many techniques in RL exist for learning behavior policies with or without explicitly learning models, and they could provide another interesting direction for ambient intelligence (cf. Hartz Søraker and Brey, 2007).

In the same direction the advent of *probabilistic programming languages* is very relevant. At the crossing of approaches in probabilistic learning, RL, and relational learning and even robotics,

[37] With the *UCI Machine Learning Repository* being one of the oldest and most well-known (http://www.ics.uci.edu/ mlearn/MLRepository.html).

[38] One cycle of this was discussed earlier when we mentioned the system by (Van den Broeck *et al.*, 2010); only here the model was given beforehand and not learned.

programming languages are developing that support probabilistic ML and RL[39]. As an example, take the following (made-up) program fragment:

```
p := P(customer X buys product Y | X browsed through webpages Z and W)
IF p > 0.70 THEN place_advertisement_discount(page_top, product_X)
r := profit_product_X - advertisement_cost
...
```

Both p and r are *parameters* of the program that are not available when writing the program. The program itself expresses the knowledge available at that time, and the parameters are *autonomously learned* from data when the program becomes active. A parameter such as p can come from e.g. a Bayesian network as the one we have discussed in this article, since most of these languages are based on the same probabilistic graphical models we have discussed. A program can also contain several *choice points*, i.e. places where the program can select one of several actions, which can be learned through RL (cf. Simpkins *et al.*, 2008; van Otterlo, 2009). Several[40] probabilistic relational learning systems are being expanded into such fully adaptive, probabilistic programming languages (Poole, 2010). Profiling *programs*, programmed in such languages will eventually have full autonomy[41] in gathering, analyzing and exploiting data, and it is especially here where new TETs are required[42].

5. Conclusions

In this article we report on a machine learning view on profiling. We have discussed representations and algorithms for learning models of user behaviors, which we have framed as activity recognition, an active branch of machine learning and artificial intelligence. We have discussed some descriptive models in more detail, to highlight vital issues in the process of learning about, and prediction of, users and their behaviors.

The previous section has presented two directions that are, in our opinion, interesting to look at in the future when it comes to privacy issues in profiling. The *transparency* issue is important, and it is interesting to see that recent directions in machine learning seem to be focused on transparency, albeit often for different reasons. In many other cases, transparency seems to be a paradoxical issue: on the one hand, transparency in experiments gives the opportunity to *relate*, *reuse* and *explain*, yet on the other hand, the success of e.g. Google makes comprehensibility and transparency less relevant since "big statistics" can solve many problems in searching for the right webpage and translating texts. Transparency is a key issue though, and especially in the context of adaptive, probabilistic programming languages, which constitutes the second direction.

References

Acquisti, A. and Gross, R. (2009), Predicting Social Security Numbers from Public Data, *in: Proceedings of the National Academy of Science*, volume 106, pp. 10975–10980.

Alpaydin, E. (2004), *Introduction to Machine Learning*, The MIT Press, Cambridge, Massachusetts.

Anrig, B., Browne, W. and Gasson, M. (2008), The Role of Algorithms in Profiling, *in:* Hildebrandt, M. and Gutwirth, S. (eds.), *Profiling the European Citizen: Cross-Disciplinary Perspectives*, chapter 4, Springer, pp. 65–87.

[39] One one the main persons in current ML lists this as one of the strategic directions in ML (in 2006), see (Mitchell, 2006).

[40] But see also efforts such as Microsoft's *infer.net* (http://research.microsoft.com/en-us/um/cambridge/projects/infernet/).

[41] As always, film makers are quite creative when it comes to computers running out of control, for example see *Wargames* (http://www.imdb.com/title/tt0086567/), *2001: A Space Odyssey* (http://www.imdb.com/title/tt0062622/), *Eye-Borgs* (http://www.imdb.com/title/tt1043844/) and *Eagle Eye* (http://www.imdb.com/title/tt1059786/).

[42] Apparently the game industry has long been very sceptical about game products that would – using AI and ML – adapt to the owner, for reasons of unexpected or unwanted developments into product modifying themselves in the wrong direction.

Barbaro, M. and Zeller Jr., T. (2006), A Face is Exposed for AOL Searcher No. 4417749, New York Times August 9.

Best, J. (2001), *Damned Lies and Statistics*, University of California Press, Berkeley and California, California.

Bollier, D. and Firestone, C. M. (2010), The Promise and Peril of Big Data, the Aspen Institute, ISBN 0-89843-516-1.

Braitenberg, V. (1984), *Vehicles: Experiments in Synthetic Psychology*, The MIT Press, Cambridge, Massachusetts.

Brandimarte, L., Acquisti, A. and Loewenstein, G. (2009), Privacy Concerns and Information Disclosure: An Illusion of Control Hypothesis, *in: INFORMS Annual Meeting*.

Brey, P. (2005), Freedom and Privacy in Ambient Intelligence, *Ethics and Information Technology*, volume 7, pp. 157–166.

Brooks, R. A. (1991), Intelligence without Representation, *Artificial Intelligence*, volume 47, pp. 139–159.

Cao, L. (2006), Activity Mining: Challenges and Prospects, *in: Proceedings of ADMA*, volume 4093 of *LNAI*, pp. 582–593.

Choudhury, T., Philipose, M., Wyatt, D. and Lester, J. (2006), Towards Activity Databases: Using Sensors and Statistical Models to Summarize People's Lives, *IEEE Data Engineering Bulletin*, volume 29(1), pp. 49–58.

Damen, D. and Hogg, D. C. (2009), Attribute Multiset Grammars for Global Explanations of Activities, *in: BMVC*, pp. xx–yy. URL `http://www.bmva.org/bmvc/2009/Papers/Paper111/Paper111.pdf`

De Raedt, L. (2008), *Logical and Relational Learning*, Springer.

de Vries, K. (2010), Identity, profiling algorithms and a world of ambient intelligence, *Ethics and Inf. Technol.*, volume 12(1), pp. 71–85.

Džeroski, S. and Lavrac, N. (eds.) (2001), *Relational Data Mining*, Springer-Verlag, Berlin.

Epstein, J. and Axtell, R. (1996), *Growing Artificial Societies: Social Science From The Bottom Up.*, MIT Press.

Gonick, L. and Smith, W. (1993), *The cartoon guide to statistics*, HarperCollins, New York.

Hartz Søraker, J. and Brey, P. (2007), Ambient Intelligence and Problems with Inferring Desires from Behaviour, *International Review of Information Ethics*, volume 8(8), Special issue on: Ethical Challenges of Ubiquitous Computing.

Hastie, T., Tibshirani, R. and Friedman, J. H. (2001), *The Elements of Statistical Learning: DataMining, Inference, and Prediction*, Springer Verlag, New York.

Hildebrandt, M. (2008), Defining Profiling: A New Type of Knowledge, *in:* Hildebrandt, M. and Gutwirth, S. (eds.), *Profiling the European Citizen: Cross-Disciplinary Perspectives*, chapter 2, Springer, pp. 17–45.

Hildebrandt, M. and Gutwirth, S. (2008a), Concise Conclusions: Citizens out of Control, *in:* Hildebrandt, M. and Gutwirth, S. (eds.), *Profiling the European Citizen: Cross-Disciplinary Perspectives*, chapter 17, Springer, pp. 365–368.

——— (2008b), General Introduction and Overview, *in:* Hildebrandt, M. and Gutwirth, S. (eds.), *Profiling the European Citizen: Cross-Disciplinary Perspectives*, chapter 1, Springer, pp. 1–13.

Hildebrandt, M. and Gutwirth, S. (eds.) (2008c), *Profiling the European Citizen: Cross-Disciplinary Perspectives*, Springer.

Hildebrandt, M. and Koops, B. J. (2010), The Challenges of Ambient Law and Legal Protection in the Profiling Era, *The Modern Law Review*, volume 73, pp. 428–460.

Josephson, J. and Josephson, S. (1994), *Abductive Inference: Computation, Philosophy, Technology*, Cambridge University Press, Cambridge, England.

King, R. D., Oliver, J. R. S. G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P., Soldatova, L. N., Sparkes, A., Whelan, K. E. and Clare, A. (2009), The Automation of Science, *Science*, volume 324(5923), pp. 85–89. URL `http://www.sciencemag.org/cgi/content/abstract/324/5923/85`

King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G. K., Bryant, C. H., Muggleton, S. H., Kell, D. B. and Oliver, S. G. (2004), Functional genomic hypothesis generation and experimentation by a robot scientist, *Nature*, volume 427, pp. 247–252.

Koller, D. and Friedman, N. (2009), *Probabilistic Graphical Models: Principles and Techniques*, MIT Press.

Krueger, V., Kragic, D., Ude, A. and Geib, C. (2007), The Meaning of Action: A Review on action recognition and mapping, *Advanced Robotics*, volume 21(13), pp. 1473–1501.

Langley, P. (2006), Cognitive Architectures and General Intelligent Systems, *AI Magazine*, volume 27, pp. 33–44.

Langley, P., Laird, J. E. and Rogers, S. (2009), Cognitive Architectures: Research issues and challenges, *Cognitive Systems Research*, volume 10(2), pp. 141 – 160.

Le, T. L., Boucher, A., Thonnat, M. and Bremond, F. (2010), Surveillance video retrieval: what we have already done?, *in: The Third International Conference on Communications and Electronics (ICCE)*.

Lenat, D. *et al.* (1990), CYC: towards programs with common sense, *Communications of the ACM*, volume 33(8).

Liao, L., Fox, D. and Kautz, H. (2005), Location-based Activity Recognition, *in: Proceedings of the Neural Information Processing Conference (NIPS)*.

Mitchell, T. M. (1997), *Machine Learning*, McGraw-Hill, New York.

——— (2006), The Discipline of Machine Learning, *Technical Report CMU-ML-06-108*, U.S.A.

Neudecker, S. (2010), Die Tricks der Anderen, Zeit Wissen Juni 2010.

Nilsson, N. J. (2010), *The Quest for Artificial Intelligence*, Cambridge University Press.

O'Donohue, W. and Ferguson, K. E. (2001), *The Psychology of B. F. Skinner*, Sage Publications.

Papert, S. (1980), *Mindstorms*, Basic Books.

Paulos, J. A. (1988), *Innumeracy: Mathematical illiteracy and its consequences*, Vintage Books/Random House, New York.

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufman.

Perkowitz, M., Philipose, M., Fishkin, K. P. and Patterson, D. J. (2004), Mining models of human activities from the web, *in: Proceedings of the 13th international conference on World Wide Web, WWW*, pp. 573–582.

Pfeifer, R. and Scheier, C. (1999), *Understanding Intelligence*, The MIT Press, Cambridge, Massachusetts.

Poole, D. (2010), Probabilistic Programming Languages: Independent Choices and Deterministic Systems, *in:* Dechter, R., Geffner, H. and Halpern, J. (eds.), *Heuristics, Probability and Causality: A Tribute to Judea Pearl*, College Publications, pp. 253–269.

Poole, D., Smyth, C. and Sharma, R. (2009), Ontology Design for Scientific Theories That Make Probabilistic Predictions, *IEEE Intelligent Systems*, pp. 27–36, special Issue on Semantic Scientific Knowledge Integration.

Reichmann, W. J. (1961), *Use and Abuse of Statistics*, Methuen London.

Resnick, M. (1994), *Turtles, Termites, and Traffic Jams*, MIT Press.

Russell, S. J. and Norvig, P. (2003), *Artificial Intelligence: a Modern Approach*, Prentice Hall, New Jersey, 2nd edition.

Schaeffer, J. and Plaat, A. (1997), Kasparov versus Deep Blue: The Re-Match, *International Computer Chess Association Journal*, volume 20(2), pp. 95–101.

Schölkopf, B. and Smola, A. J. (2002), *Learning with Kernels*, MIT Press, Cambridge, Massachusetts.

Simpkins, C., Bhat, S., Isbell, C. L. and Mateas, M. (2008), Adaptive Programming: Integrating Reinforcement Learning into a Programming Language, *in: Proceedings of the Twenty-Third ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, pp. 603–614.

Singla, P., Kautz, H., Luo, J. B. and Gallagher, A. C. (2008), Discovery of social relationships in consumer photo collections using Markov Logic, *in: CVPR Workshop on Semantic Learning and Applications in Multimedia*, pp. 1–7.

Skinner, B. F. (1976), *About Behaviorism*, Vintage Books, New York.

Sobiesk, E. and Conti, G. J. (2007), The Cost of Free Web Tools, *IEEE Security & Privacy*, volume 5(3), pp. 66–68. URL `http://dx.doi.org/10.1109/MSP.2007.74`

Sowa, J. F. (1999), *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Thomson Learning, Stamford, Connecticut.

Sutton, R. S. and Barto, A. G. (1998), *Reinforcement Learning: an Introduction*, The MIT Press, Cambridge, Massachusetts.

Torre, I. (2009), Adaptive systems in the era of the semantic and social web, a survey, *User Modeling and User-Adapted Interaction*, volume 19(5), pp. 433–486.

Turaga, P., Chellappa, R., Subrahmanian, V. S. and Udrea, O. (2008), Machine Recognition of Human Activities: A Survey, *IEEE Transactions on Circuits, Systems and Video Technology*, volume 18(11), special issue on Event Analysis.

Tversky, A. and Kahneman, D. (1981), The Framing of Decisions and the Psychology of Choice, *Science*, volume 211(4481), pp. 453–458.

Van den Broeck, G., Thon, I., van Otterlo, M. and De Raedt, L. (2010), DTProbLog: A Decision-Theoretic Probabilistic Prolog, *in: Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

van Otterlo, M. (2009), *The Logic of Adaptive Behavior*, IOS Press, Amsterdam, The Netherlands.

Vanschoren, J. (2010), *Understanding Machine Learning Performance with Experiment Databases*, Ph.D. thesis, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium.

von Ahn, L. and Blum, M. (2006), Peekaboom: A Game for Locating Objects in Images, *in: ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 55–64.

Warner, M. and Stone, M. (1970), *The Data Bank Society; Organizations, Computers and Social Freedom*, Allen and Unwin, London.

Whyte, J. (2004), *Crimes Against Logic*, McGraw-Hill.

Witten, I. and Frank, E. (2005), *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2nd edition.

Xiang, T. and Gong, S. (2006), Beyond Tracking: Modelling Activity and Understanding Behaviour, *International Journal of Computer Vision*, volume 67(1), pp. 21–51.

Yang, Q. (2009), Activity Recognition: Linking Low-level Sensors to High-level Intelligence, *in: IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*.

Zamyatin, Y. (1924), *We*, Penguin Classics (1993).

Zouba, N., Bremond, F., Thonnat, M., Anfonso, A., Pascual, E., Mallea, P., Mailland, V. and Guerin, O. (2009), A Computer system to monitor older adults at home: preliminary results, *Gerontechnology*, volume 8(3).