Chapter 2

Research Challenges in Ubiquitous Knowledge Discovery

Michael May, Bettina Berendt, Antoine Cornuéjols, Jõao Gama, Fosca Giannotti, Andreas Hotho, Donato Malerba, Ernestina Menesalvas, Katharina Morik, Rasmus Pedersen, Lorenza Saitta, Yücel Saygin, Assaf Schuster, Koen Vanhoof

Fraunhofer IAIS, KU Leuven, LRI, University of Porto, ISTI-CNR, University of Kassel, University Bari, University Madrid, University of Dortmund, Copenhagen Business School, University of Piemonte Orientale, Sabanci University, Technion, University of Hasselt

Abstract Ubiquitous Knowledge Discovery is a new research area at the intersection of machine learning and data mining with mobile and distributed systems. In this paper the main characteristics of the objects of study are defined. Next, a number of examples from a broad range of application areas are reviewed and analyzed. Based on this material, important characteristics of this field are identified and a number of research challenges are discussed. The purpose of this chapter is to chart the territory, to identify landmarks and challenges ahead.

2.1 Ubiquitous Knowledge Discovery

2.1.1 Introduction

Knowledge Discovery in ubiquitous environments (KDubiq) is an emerging area of research at the intersection of the two major challenges of highly distributed and mobile systems and advanced knowledge discovery systems.

Today, in many subfields of computer science and engineering, being intelligent and adaptive marks the difference between a system that works in a complex and changing environment and a system that does not work. Hence, projects across many areas, ranging from Web 2.0 to ubiquitous computing and robotics, aim to create systems which are “smart”, “intelligent” , “adaptive” etc., allowing to solve problems that could not be solved before. A central assumption of ubiquitous knowledge discovery is that what seems to be a bewildering array of different methodologies and approaches for building smart, adaptive, intelligent systems, can be cast into a coherent, integrated set of key ideas centered on the notion of learning from experience.

Focusing on these key ideas, ubiquitous knowledge discovery aims to provide a unifying framework for systematically investigating the mutual dependencies of otherwise quite unrelated technologies employed in building next-generation intelligent systems: machine learning, data mining, sensor networks, grids, P2P, data stream mining, activity recognition, Web 2.0, privacy, user modeling and others. Machine learning and data mining emerge as basic methodologies and indispensable building blocks for some of the most difficult computer science and engineering challenges of the next decade.

The first task is to characterize the *objects of study* for ubiquitous knowledge discovery more clearly. The objects of study

1. exist in time and space in a dynamically changing environment,

2. can change location and might appear or disappear,

3. have information processing capabilities,

4. know only their local spatio-temporal environment,

5. act under real-time constraints,

6. are able to exchange information with other objects.

Objects to which these characteristics apply are humans, animals, and, increasingly, various kinds of computing devices. It is the latter, that form the objects of study for ubiquitous knowledge discovery.

2.1.2 Dimensions of Ubiquitous Knowledge Discovery Systems

Mainstream data mining and machine learning is focused centrally on the learning algorithm. Algorithms are typically treated as largely independent from the application domain and the system architecture in which the algorithm is latter embedded. Thus the same implementation of a support vector machine can be applied to texts, gene expression data, or credit card transactions; the difference is in the feature extraction during pre-processing.

Design Space. Ubiquitous knowledge discovery challenges these independence assumptions in several ways. In the further sections it will be argued, that often the learning algorithms have to be tailored for a specific network topology characterized by communication constraints, reliability, or resource availability. Thus when designing a ubiquitous knowledge discovery system, major design decisions in various dimensions have to be taken. These choices are mutually constraining each other. Dependencies among them have to be carefully analyzed. For analyzing the different possible architectures of ubiquitous knowledge the *design space* of ubiquitous knowledge discovery systems is factored into six dimensions:

- *Application Area.* What is the real-world problem being addressed?
- *Ubiquitous Technologies.* What types of sensors are used? What type of distributed technology is used?
- *Resource Aware Algorithms.*Which machine learning or data mining algorithms are used? What are the resource constraints imposed by the ubiquitous technologies? How does the algorithm adapt to a dynamic environment?
- *Ubiquitous Data Collection.* What issues arise from information integration of the sensors? Are the issues from collaborative data generation?
- *Privacy and Security.* Does the application create privacy risks?
- *Human Computer Interaction (HCI) and User-Modeling.* What is the role of the user in the system? How does he interact with the devices?

Ubiquity of Data and Computing. Two important aspects of ubiquity have to be distinguished, namely the *ubiquity of data*, and the *ubiquity of computing*. In a prototypical application the ubiquity of computing corresponds naturally to the ubiquity of the data: the data is analyzed when and where it is generated – the knowledge discovery takes place *in situ*, inside the interacting, often collaborating, distributed devices.

There exist however borderline cases that are ubiquitous in one way but not in the other, e.g. clusters or grids for speeding up data analysis by distributing files and computations to various computers, or track mining from GPS data where the data are analyzed on a central server in an offline batch setting.

While research on this kind of systems is in several respects highly relevant for ubiquitous knowledge discovery, for the purpose of this chapter a more narrow point of view is adopted, and the following characterization is assumed: *Ubiquitous knowledge discovery is that part of machine learning and data mining that investigates learning* in situ*, inside a dynamic distributed infrastructure of interacting artificial devices.*

2.2 Example 1: Autonomous driving vehicles

To provide a more specific description of the content of ubiquitous knowledge discovery, in the next sections a number of examples is analyzed. The following selection criteria have been used: (1) each example focuses on a different domain; (2) it presents a challenging real-life problem; (3) there is a body of prior technical work addressing at least some of the six dimensions of ubiquitous knowledge discovery, while other dimensions are not covered.

Contributions from various fields are analyzed – robotics, ubiquitous computing, machine learning and data mining. Work is not necessarily done under the label of "ubiquitous knowledge discovery", since the subject is new and draws inspiration from work scattered around many communities. These examples provide material for discussing the general features of ubiquitous knowledge discovery in the next sections.

Application. Modern vehicles are a good starting point to discuss ubiquitous knowledge discovery systems, since they exist in a dynamic environment, move in time and space, and are equipped with a number of sensors. There are various directions to add to the intelligence of modern cars. An ambitious attempt is to construct *autonomous driving vehicles*. In the DARPA 2005 grand challenge, the goal was to develop an autonomous robot capable of traversing unrehearsed road-terrain: to navigate a 228 km long course through the Mojave desert in no more than 10 hours. The challenge was won in 2005 by the robot Stanley. What sets robots such as Stanley apart from traditional cars on the hardware side is the large number of additional sensors, computational power and actuators.

Learning Component. Machine learning is a key component of Stanley, being used for a number of learning tasks, both offline and online [28]. An offline classification task solved with machine learning is obstacle detection, where a first order Markov model is used. The use of machine learning is motivated by the fact that it would be impossible to train the system offline for all possible situations the car might encounter.

More importantly for our discussion, a second online task is road finding: classifying images into drivable and non-drivable areas. Drivable terrain is globally represented by a mixture of n Gaussians defined in the RGB color space of pixels. A new image is mapped into a small number of k local Gaussians (where $k << n$); they are used to update the global model. This way, a distribution that changes over time can be modeled. During learning the mean and variance of the global Gaussian and a pixel count can be updated, new Gaussians can be added and old ones discarded. The decision whether to adapt or to add or forget is taken by calculating the Mahalanobis distance $d(i,j) = (\mu_i - \mu_j)^T(\Sigma_i + \Sigma_j)^{-1}(\mu_i - \mu_j)$ between local and global Gaussians. Additionally, exponential decay is used for the counters in memory. An area is classified as drivable, if its pixel values are close to the learned Gaussians that characterize drivable terrain. Adapting the parameters helps to model slow changes in the terrain, while adding and forgetting can accommodate for abrupt changes.

The significance of Stanley for our present discussion is that it provides a very pictorial example how acting in a dynamic environment combined with real-time constraints demands learning algorithms that have have been a niche topic in machine learning so far: *algorithms that can adapt to concept drift, i.e. to distributions that can change slowly or abruptly over time.* In the next examples we see that this is an almost universal feature of ubiquitous knowledge discovery, and in sec. 2.6.2 a general discussion can be found.

Communication. Autonomous robotics puts strong emphasis on planning and control to achieve the vision of autonomy, where for ubiquitous knowledge discovery full autonomy is normally not the goal. Instead, collaboration and interaction among humans and devices is stressed. The desert driving scenarios is very limited in this respect if compared to a normal traffic scenario. The robot has no knowledge about the existence of other objects similar to itself (treating them as obstacle at best), not matching characteristic (6) above. The DARPA 2007 urban challenge was a step in that direction, since vehicles were required to navigate their way under normal traffic conditions, with other cars present, turns etc. Thus the vehicles needed modules for tracking other cars [27]. Yet cars were not able to communicate [7] or learn from each other. A distributed protocol of learning among cars was outside the scope of the challenge.

2.3 Example 2: Activity recognition – inferring transportation routines from GPS-data

Application. The widespread use of GPS devices has led to an explosive interest in spatial data. Classical applications are car navigation and location tracking. Intensive activity, notably in the ubiquitous computing community, is underway to explore additional application scenarios, e.g., in assistive technologies or in building models of the mobile behavior of citizens, useful for various areas, including social research, planning purposes, and market research. We discuss an application from assistive technologies, analyze its strength and shortcomings and identify research challenges from a ubiquitous knowledge discovery perspective.

The *OpportunityKnocks* prototype [21] consists of a mobile phone equipped with GPS and connected to a server in a mobile client/server setup. The mobile phone can connect to a server via GPRS and transmit the GPS signals, thus tracking the person's behavior. The server analyzes the data, utilizing additional information about the street network or bus schedules from the Internet. Using this information the person is located and the system makes inferences about his current behavior and gives suggestions what to do next. This information is sent back to the client and communicated to the user with the help of an audio/visual interface.

The system is able to give advice to persons, e.g., which route to take or where to get off a bus, and it can warn the user in case he commits errors, e.g., takes the wrong bus line. The purpose of the system is to assist cognitively impaired persons in finding their way through city traffic.

This application meets the main criteria for ubiquitous systems: the device is an object moving in space and time in a changing and unknown environment; it has computing power, and has a local view of its environment only; it reacts in real-time and it is equipped with GPS-sensors and exchanges information with other objects (e.g., satellites, the server). Compared to the last example, the current one does not aim for an autonomous device but is designed for interaction with a human.

Learning components. Since both the environment and the behavioral patterns are not known in advance, it is impossible to solve this task without the system being able to learn from a user's past behavior. Thus, machine learning algorithms are used to infer likely routes, activities, transportation destinations and deviations from a normal route. The basic knowledge representation mechanism is a *hierarchical dynamic Bayesian network*. The topology of the network is manually build and creates a hierarchy, with the upper level devoted to novelty detection, the middle layer responsible for estimating user's goals and trip segments, the lowest level representing mode of transportation, speed, and location. Time is represented by arcs t and $t-1$ connecting time slices. While the generic network design is specified in advance and is the same for every user, the specific parameters of the distribution are learned from the data in an unsupervised manner. Data comes in streams, but apparently the full information is stored in a database. For efficient online inference of the hidden variables given the GPS data, a combination of particle and Kalman filtering is employed [21].

Although innovative, the architecture of this prototype will face a number of practical problems. Thus in absence of a phone signal, communication with the server is impossible, and the person may get lost. Similarly, when there is no reliable GPS signal, e.g., in urban canyons, or indoors, guidance is impossible. A further problem is that communicating via a radio network with a server consumes a lot of battery power, so that the system works only 4 hrs under continuous operation. Finally, continuously tracking of a person and centrally collecting the data creates strong privacy threats.

An implicit assumption of the prototype seems that sensing is always possible, that communication between client and server is generally reliable, and that power consumption does not play an important role. In other words, it assumes a setting as is appropriate in a local network. But these assumptions are invalid to a degree that would prevent a real-world deployment of the system. It should be noted that cognitive assistance is more demanding here than e.g. usual car navigation, because the the people may be helpless without the device.

Moving the learning to the device. The significance of this example is that on the one hand it describes a highly interesting scenario for ubiquitous knowledge discovery and advanced machine learning methods, but on the other hand the design of the overall system does not match the constraints of a ubiquitous environment. Ubiquitous knowledge discovery starts with the observation *a learning algorithm cannot be designed in abstraction from the characteristics of the systems on which it is deployed* (see sec. 2.6.3).

The ubiquitous knowledge discovery paradigm asks for distributed, intelligent and communicating devices integrating data from various sources. A "KDubiq Upgrade" would result in a much more satisfactory design for the prototype. It would be guided by the imperative to move the machine learning to the mobile device. If the major part of the learning is done on the mobile device – especially that part that refers to localization on the street map – there is no need for constant server communication, and assistance becomes more reliable.

Splitting the computation into an energy and computationally efficient onboard part yielding highly compressed models, transmitting only this compressed information and performing computationally intensive parts on the server would be a favorable solution. Section 2.5 discusses this in more detail.

Industrial application. A GPS device that can do data mining *in situ* and on-the-fly has important industrial applications. Here we describe one such scenario. The "Arbeitsgemeinschaft Media-Analyse" (ag.ma) – a joint industry committee of around 250 principal companies of the advertising and media industry in Germany – commissioned in 2007 a nationwide survey about mobile behavior of the German population. This is the basis for determining performance characteristics in outdoor advertising – e.g. the percentage of the population that has had contact with a poster network within one week. The basic input are mobility data in form of GPS trajectories. Nationwide, the daily movements of about 30.000 people have been surveyed for up to seven days.

In order to model the behavior of the *overall German population* from this sample, a number of data mining and modeling tasks had to be solved by Fraunhofer. Using techniques based on survival analysis and simulation techniques the mobile behavior for all German cities is estimated [1]. A second data source is a sample of approx. 100.000 video measurements on traffic frequencies in German cities. A k-NN-based spatial data mining algorithm has been developed that derives traffic frequency predictions for 4 Mio. street segments in German cities ([23], sec.9.6). Track data and frequency estimates are combined in a data fusion step. Other tasks are related to spatial clustering. The application described so far is about analyzing data collected with ubiquitous mobile devices. Collectively it took several month and several project partners to complete the data preparation. This project has a high impact because the pricing of posters is based these models and a whole branch of German industry is based on the data mining predictions.

A future scenario is to do all the track related data preparation online. We are currently working on a scenario where data mining is done in the GPS-device and annotated tracks are inferred on the fly. For applications where the user has agreed to make his data available, not the raw GPS data, but annotated diaries of activities are send to a server via a radio network and processed using a Grid infrastructure [30]. This would not only shorten development time dramatically, but allows the possibility to derive a snapshot of a population's mobility with very short delay.

2.4 Example 3: Ubiquitous Intelligent Media Organization

While the first example did not match the collaborative aspect of ubiquitous knowledge discovery and the second did not investigate learning *in situ*, the next two examples, while very different from each other, match all criteria of ubiquitous knowledge discovery.

Application. With the advent of Web 2.0, collaborative structuring of large collections of multi-media data based on meta-data and media features has become a significant task. Nemoz (NEtworked Media Organizer) [9] is a Web 2.0-inspired collaborative platform for playing music, browsing, searching and sharing music collections. It works in a loosely-coupled distributed scenario, using P2P technology. Nemoz combines Web 2.0-style tagging, with automatic audio classification using machine learning techniques. This application is a representative of a innovative subclass of applications in a Web 2.0 environment. Whereas most Web 2.0 tagging applications use a central server where all media data and tags are consolidated, the current application is fully distributed.

The application differs from the preceding ones in that the (geo)spatio-temporal position of the computing devices does not play an important role; the devices and the media file collections they contain are stored somewhere on some node in the network. Yet it is a defining characteristic of the application that two collections $\mathcal{C}_i$ and $\mathcal{C}_j$ are stored at different places, and it is important whether or not two collections are connected via a neighborhood graph.

Also in contrast to the other examples the *fully distributed nature* of the problem is a defining characteristic of the application. In a P2P environment computing devices might be connected to a network only temporarily, communication is unreliable and the collections are evolving dynamically; items are added and deleted, and also classifications can change. In many distributed data mining applications, originally centralized data are distributed for improving the efficiency of the analysis. The current application is different because, firstly, the data are inherently distributed, and secondly, there is no intention to come up with a global model. Thus it is, as discussed in the introduction, a system where the ubiquity of computing naturally corresponds to the ubiquity of data.

Learning components. Nemoz is motivated by the observation that a globally correct classification for audio files does not exist, since each user has its own way of structuring the files, reflecting his own preferences and needs. Still, a user can exploit labels provided by other peers as features for his own classification: the fact that Mary, who structures here collection along mood, classifies a song as melancholic might indicate to Bob, who classifies along genre, that it is not a Techno song. To support this, Nemoz nodes are able to exchange information about their individual classifications. These added labels are used in a predictive machine learning task. Thus the application is characterized by evolving collections of large amounts of data, scattered across different computing devices that maintain a local view of a collection, exchanging information with other nodes. It is a crucial aspect of this application that the nodes maintain a local view, incorporating information from other nodes. We are not aware of other solutions that are able to automatically *learn from other user's classifications while maintaining a local or subjective point of view.*

The significance of this example is that Nemoz introduces *a new class of learning problems*: the collaborative representation problem and localized alternative cluster ensembles for collaborative structuring (LACE) [33]. From the perspective of ubiquitous knowledge discovery this is important, since in a non-distributed environment, these new learning scenarios would be very hard to motivate. This new class is relevant for both examples discussed above. If a mobile device is used in city traffic, it could be very helpful if the device would be able to exchange information with other devices. In this case each device would maintain a local model of the surrounding traffic partially by exchanging information with other devices. It can also be imagined that a future competition on autonomous vehicle driving that includes communication among cars would include learning of local, subjective views in a collaborative setting.

This potential transfer of learning scenarios from seemingly very unrelated areas mobile assistive technology, autonomous robots and music mining is made possible by analyzing the applications in a common framework.

2.5 Example 4: Real-Time Vehicle Monitoring

Application. The Vehicle Data Stream Mining System VEDAS [18] is a mobile and distributed data stream mining application. It analyzes and monitors the continuous data stream generated by a vehicle. It is able to identify emerging patterns and reports them back to a remote control center over a low-bandwidth wireless network connection. Applications are real-time on-board health monitoring, drunk-driving detection, driver characterizations, and security related applications for commercial fleet management.

VEDAS uses a PDA or other light weight mobile device installed in a vehicle. It is connected to the On Board Diagnostic System (OBD-II); other sensory input comes from a GPS device. Significant mining tasks are carried out on board, monitoring the state of transmission, engine and fuel systems. Only aggregated information is transmitted to a central server via a wireless connection. The data-mining has to be performed on-board using a streaming approach, since the amount of data that would have to be transmitted to the central server is too huge.

Learning components. The basic idea of the VEDAS data mining module is to provide distributed mining of multiple mobile data sources with little centralization. The data mining algorithms are designed around the following ideas: minimize data communication; minimize power-consumption; minimize on-board storage; minimize computing usage; respect privacy constraints.

VEDAS implements incremental PCA, incremental Fourier transform, online linear segmentation, incremental k-means clustering and several lightweight statistical techniques. The basic versions of these algorithms are of course well-known and precede the data mining age. The innovation lies here in adapting to a resource-constrained environment, resulting in new approximate solutions.

A comparison of this example with the activity recognition scenario reveals resources to improve the latter scenario. It offers a solution that locally computes and pre-aggregates results and communicates only few data via the radio network, splitting the computation into an energy and computationally efficient on-board part yielding highly compressed models, transmitting only this compressed information and performing computationally intensive parts on the server.

However, an additional price is paid: for the on-board part new algorithms are necessary that trade accuracy against efficiency. The specific trade off is dictated by the application context, and the choice made in the vehicle monitoring application would be hard to motivate in an offline-context (or even for the current application).

The significance of this example is as a template how to design *resource aware* mobile data mining solutions and it avoids some of the pitfalls of the activity recognition scenario.

2.6 Research Challenges

The ubiquitous knowledge discovery paradigm asks for distributed, intelligent and communicating devices integrating data from various sensor sources. The learning takes place *in situ*. Privacy has to be addressed.

The examples discussed match that paradigm to various degrees. Example 1 showed the importance of algorithms that are able doing inference in real-time, inside the device, paying attention to concept drift. To turn this into a realistic scenario, communication among cars would be needed. Example 2 used a mobile client server scenario. However for a realistic deployment it would be necessary to move the algorithms to the device. Additionally, privacy constraints have to be addressed. Example 3 was an example for a fully distributed scenario in which nodes learn from each other and build a local subjective model. Example 4 finally provided a template for building resource-aware approximate algorithms for monitoring the state of a system.

In the following section the general characteristics and challenges that emerge for research in ubiquitous knowledge discovery are discussed.

2.6.1 Resource Constraints

In applications comprising mobile and/or small devices limitations in storage, processing and communication capabilities, energy supply and bandwidth, combined with a dynamic and unreliable network connectivity are a major constraining factor. Optimizing a learning algorithm for such systems often leads to a coupling of application semantics and the system layers.

In many cases the best available algorithm might be too demanding in terms of computational or memory resources to be run on the designated device. Thus an approximation has to be designed. The approximation might depend on the exact configuration of the system and on specifics about the application. E.g. in example 4, the state of the vehicle is monitored in real-time. To monitor a set of variables, a principal component analysis is performed. Changes in driving characteristics result in changes in the eigenvectors of the covariance matrix. Because constant recomputation would be too costly, the system determines only the upper bounds on changes in eigenvalues and eigenvectors over time, and initiates a recomputation only if necessary [18]. Other examples for the use of approximation in order to save resources are [26],

Table 2.1: Violating the iid assumption gives rise to different areas of machine learning and statistics.

	independent	not independent
identical	Statistical Learning Theory, PAC, Mainstream DM	Simple Kriging, Stationary Time Series, Statistical Relational Learning, Markov Chains
not identical, slowly changing	PAC online learning	ARIMA, State Space Models, Kalman Filter
not identical abrupt changes	Concept Drift CUSUM	Piecewise ARIMA

where a trade-off between accuracy and communication load for monitoring threshold functions in various kinds of distributed environments is discussed. For a general overview on resource-aware computing in sensor networks, see Zhao & Guibas [34].

2.6.2 Beyond identically distributed data

An important challenge is that ubiquitous knowledge discovery focuses on learning beyond identically distributed data. Although work in this area exists, this implies a significant shift in focus from the current mainstream in data mining and machine learning. At the core of the problem is the following observation: *In a ubiquitous setting, we cannot assume anymore to have an independent and identically distributed sample of the underlying distribution.* The reason is that inference takes place under real-time constraints in a dynamically changing environment, as has been describe in example 1.

A collection of random variables $X_1, ..., X_n$ is said to be independent and identically distributed (iid) if each X_i has the same distribution function f and the X_i are mutually independent. Each of the two conditions – independence and identical distribution – may be violated separately, or both may be violated at the same time. This gives rise to distinct areas of machine learning and statistics (table 2.1).

Most practical and theoretical results in machine learning depend on the iid assumption (top left corner of the table). The main body of PAC learning [15] and statistical learning theory [29] are crucially based on it. The reason is that if we sample independently from a distribution that is supposed to be fixed and invariant with respect to time (i.e. if the process is iid), all necessary information about the future behavior of a signal can in principle be obtained from a (sufficiently large) sample of past or present values. This justifies to attempt forecasting. Moreover, techniques such as cross-validation can be used to assess the prediction error on future instances.

Dependent data. If the independence assumption is invalid but the distribution is fixed, sampling instances become autocorrelated (top right corner of table). Traditionally, the theory of stationary time series [5] and spatial statistics (e.g. simple Kriging) [6] deal with temporally and spatially autocorrelated variables, respectively. Markov chains are widely employed to model sequential data, and some extensions of PAC learning for this setting exist [11]. More recently, statistical relational learning starts investigating scenarios that violate the independence assumption [13, 22].

Slowly changing distributions. Some extensions of learning theory cover slowly changing sequences of concepts [16], where the instances of a concept are drawn at random (middle left cell of table). It derives bounds on how fast a concept can change so that learning is still possible.

Very common in econometrics and engineering are approaches that combine both autocorrelation and slowly changing distributions, especially ARIMA (Auto-Regressive Integrated Moving Average) and state space models [5] (middle right cell). In an ARIMA model, although the input signal can have a trend or cycles, it is assumed that after taking differences finitely many times (usually just one or two), the signal becomes stationary. So there is at least *some* component of the original signal, that gives information useful for forecasting. The trick is here that we can recover stationarity in some way, so that stationary time series analysis becomes applicable again.

Abrupt changes. The least explored but for ubiquitous knowledge discovery most relevant and interesting part of the table is where we face abrupt changes in the distribution (bottom row of table).

An important distinction concerns the available information: are only the past values available (*forecasting*), a measurement of the current value (*filtering*) or past, future and present values (*smoothing*)? As we move down the table, prediction becomes increasingly difficult, and for the case of abrupt changes, the typical setting is that of monitoring or filtering, instead of prediction. Thus examples 1, 2, 4 are all cases for monitoring or filtering. Breaks are detected but not predicted. All examples use *online algorithms*.

A body of work in machine learning on concept drift addresses this scenario. It assumes non-stationarities – breaks – but assumes the data in between to be independent [19, 31]. The predictive accuracy is monitored and once it drops a new model is learned. The main objective is to automatically keep an up-to-date model over time, e.g. in a spam filtering scenario. A streaming scenario is explicitly addressed in [12, 17]. Although designed for scalability in terms of data size, the various approaches assume sufficient compute power and are not designed e.g. for a mobile solution.

Control charts and the CUSUM method (that calculates the CUMulative SUM of differences between the current and average value) are more traditional approaches to this problem for univariate data [2], often used to monitor industrial processes. Example 4 [18] discusses vehicle monitoring and break detection in a multivariate setting.

Severo and Gama [25] combine machine learning and traditional approaches in a generic scenario where a regression algorithm, e.g. a regression tree, is used as a basic learner, its residuals are monitored and thresholds adapted using a Kalman filter; CUSUM is used for deciding whether a break occurred. If the performance degrades, a new tree is learned.

The most general setting allows data to be dependent and distributions to be non-stationary, with both slow and abrupt changes. There is recent work that combines elements from statistics and machine learning on piecewise ARIMA models that can model both slow and abrupt changes [8]. The breaks are identified using genetic algorithms and the Minimum Description Length principle. However, it presupposes that the full series is available, so breaks can be modeled only *a posteriori*; thus is not applicable in a typical ubiquitous knowledge discovery scenario.

In contrast, the drivable terrain detection algorithm that has been described in section 2.2 allows for an evolving stream of data, since it detects both slow and abrupt changes in an *online setting.* Also the activity recognition would fall under this scenario, since a person might abruptly change her behavior (e.g. her daily routine after changing job), but also more gradually (e.g. undertaking longer walks when it becomes summer).

This shows how common and important this last scenario is. *A more systematic and unified approach is needed in the Machine Learning and Data Mining community to develop methods for detecting slow and abrupt changes in possibly dependent data.* It would be important to address this in a typical data mining scenario where a large number of variables can be included, and no *a priori* knowledge about their relevance is available.

2.6.3 Locality

A third central feature are various forms of locality. Locality and the stationarity assumption are in fact closely linked. The assumption that a process is spatio-temporally stationary implies that we can make a translation on the time scale or in the spatial coordinates and the autocorrelation structure remains invariant. Thus we can take a sample at some place on earth at the beginning of the 21st century and can draw valid inferences about the process in some distant galaxy, from stone age to the end of days. While this proved to be a powerful assumption for fundamental physical processes, it leads into trouble for areas related to human activities. Assuming iid data is a way of removing spatio-temporal boundaries from our inference capabilities. Embracing dependent data and non-stationary distributions with breaks leads us to inference in temporally and/or spatially local environments. The associated challenges are discussed in this section.

Temporal Locality. A first aspect is *temporal locality.* In a typical scenario for ubiquitous knowledge discovery the learner has access to past and maybe present data from a time-varying distribution (see last section), and has to make inferences about the present (filtering) or future (forecasting). In a memory constrained environment typically a data stream setting is assumed. This setting leads to incremental and online learners, time window approaches, weighting etc. (for an overview on data streams see [10]), as already discussed in various examples.

Spatial Locality. There is a second aspect of locality that derives from *spatial locality and distributedness.* For example, in a sensor network the reach of a sensor is restricted so that it can sense only the local environment. In the case of terrain finding example 1, the vehicle is moving, having just a local snapshot of the terrain. The nodes in Nemoz have full access only to their own collection.

If the task is to have a global model about the whole terrain, locality is related to the iid assumption. None of the nodes has access to the full distribution. Instead each node has (often highly autocorrelated) measurements coming from a small range of the full distribution. In P2P networks, neighboring nodes will in some case share other relevant features as well (e.g. the kind of music they like in the Nemoz application) so that sampling values from neighbor nodes gives a biased sample.

To share this local view with others and to come up finally with a global model, a node has to communicate. But for small devices dependent on battery power communication is costly. It has been shown that in many scenarios a fully centralized solution involves too much communication overhead to be feasible. Along the same lines fully reliable, globally synchronized networking is is not attainable in many P2P scenarios [32].

Distributed learning algorithms. Under this condition, the task is to find a near optimal solution to the inference problem that takes account of specific constraints in communication and reliability. Solutions may be either exact or approximate. Local communication lead in some cases to totally different algorithms than their counter parts in centralized scenarios. As a result, algorithms appear that would make no sense in a globally centralized, static environment.

An example is the large-scale distributed association-rule mining algorithm by Wolff and Schuster [32]. It relies on a distributed *local majority voting* protocol to decide whether some item is frequent. It is applicable to networks of unlimited size. Distributed monitoring of arbitrary threshold functions based on geometric considerations is described in Sharfman et al. [26]. The case is considered where $X_1, X_2, ..., X_d$ are frequency counts for d items and where we are interested to detect whether a non-linear function $f(X_1, X_2, ..., X_d)$ rises or falls below a threshold. It is pointed out that for a non-linear function in general it is not possible to deduce if e.g. the average of two counts at X_1 and X_2, if they are passed through the function f, by just looking at the local values. This observation is e.g. relevant for a distributed spam-filter, where a system of agents is installed on a number of distributed mail servers. The task is to set up the learning systems in such a way that by monitoring the threshold at local nodes, we can be sure that if the constraints are met at all local nodes, no global violation has occurred, and thus no communication across nodes is necessary. The solution is based on *a geometric approach* to monitor threshold violations. Further examples for local inference algorithms that depend on network characteristics and involve either geometric or graph theoretic considerations are described in [34].

It is claimed that in sensor networks, the topology of the network can not be decoupled from application semantics [34]. This statement takes over to ubiquitous knowledge discovery: *the network topology, the information processing, and the core learning algorithms become mutually dependent.* This fact is a deeper reason why we cannot design ubiquitous learning algorithms independently from considerations about the underlying distributed technology, the specific data types, privacy constraints and user modeling issues.

2.6.4 Further challenges

In this section, further challenges are shortly summarized. From the discussion it emerges that ubiquitous knowledge discovery requires new approaches in spatio-temporal data mining, privacy-preserving data mining, sensor data integration, collaborative data generation, distributed data mining, and user modeling. The most successful approaches will be those, that combine several aspects. Thus ubiquitous knowledge discovery holds the potential as an integrated scenario for various otherwise fragmented directions of current research.

Spatio-Temporal Mining Case studies 1 and 2 highlighted the central role of spatio-temporal data mining, especially GPS track data. For an overview on recent developments in this area, see [23] [20].

Data collection. On the data collection side two major issues arise. The first one is collection and integration of data collected from heterogeneous sensors as in case studies 1 and 2, and 4. Example 3 highlights data collection issues in a collaborative Web 2.0 environment.

Privacy. All case studies proved to be privacy sensitive, since ubiquitous devices reveal highly sensitive information about the persons that carry them. Privacy-preserving data mining in a distributed, spatio-temporal environment poses many challenges [24] [4]. Privacy issues will become even more pressing once the application migrate from a research prototype status to real products.

User Modeling. Finally, user modeling, and HCI are particularly challenging, for examples since not only experts but technically non-skilled end users will be confronted with those systems [3]. For example, once autonomous cars would go into production, HCI and user modeling (as well as privacy) will play a central role: There are many questions starting from user acceptance (the autonomy of the car diminishes the autonomy of the user!) to liability and legal issues.

2.7 Summary

In this section common lessons from the case studies are drawn and research challenges identified.

We reviewed examples from data mining, machine learning, probabilistic robotics, ubiquitous computing, and Web 2.0. Collectively, these applications span a broad range of ubiquitous knowledge discovery applications from vehicle driving, assistive technologies, transportation, and leisure. This showed that across a large sector of challenging application domains, further progress depends on advances in the fields of machine learning and data mining; increasing the ubiquity sets the directions for further research and improved applications.

Ubiquitous knowledge discovery investigates learning *in situ*, inside distributed interacting devices and under real-time constraints. From this characterization the major challenges follow:

- *Devices are resource constrained.* This leads to a streaming setting and to algorithms that may have to trade-off accuracy and efficiency by using sampling, windowing, approximate inference etc.

- *Data is non-stationary, non-independent.* The distribution may be both temporally and spatially varying, and it may change both slowly or abruptly.

- *Locality.* Temporal locality combined with real-time properties leads to online algorithms and to a shift from prediction to monitoring, change detection, filtering or short-term forecasts. Spatial locality (combined with resource constraints) leads to distributed algorithms that are tailored for specific network topologies and that make use of graph theoretic or geometric properties.

At the heart of the algorithmic challenges for KDubiq is thus local inference beyond iid data in resource constrained environments. While inference on iid data occupies only one sixth of the cells in table 2.1, by far the most amount of work in data mining and machine learning is devoted to this topic so far. There are large areas of unexplored terrain that wait for research from the data mining and machine learning community.

From a systems perspective, the challenge consists in building learning algorithms for distributed, multi-device, multi-sensor environments. While partial suggestions exist on how to implement privacy-preserving, distributed, collaborative algorithms, respectively, there is hardly any existing work that properly addresses all the dimensions at the same time in an integrated manner. *Yet as long as one of these dimensions is left unaddressed, the ubiquitous knowledge discovery prototype will not be fully operational in a real-world environment.* We need both new algorithms – including analysis and proof about their complexity and accuracy – and an engineering approach for integrating the various partial solutions – algorithms, software and hardware – in working prototypes.

Acknowledgements. This work has been funded by the European Commission under IST-FP6-021321 KDubiq Coordination Action. The KDubiq research network (www.kdubiq.org) was launched in 2006 to stimulate research in this area. Currently it has more than 50 members. This chapter is based on the discussions in the network, and contributions from project partners are gratefully acknowledged.

References

[1] ag.ma. Kontakt und Reichweitenmodell Plakat. `http://www.agma-mmc.de/03_forschung/plakat/erhebung_methode/abfragemodell.asp?topnav=10&subnav=199`, 2008.

[2] Michele Basseville and Igor Nikiforov. *Detection of Abrupt Changes: Theory and Application.* Prentice Hall, 1993.

[3] B. Berendt, A. Kröner, E. Menasalvas, and S. Weibelzahl, editors. *Proc. Knowledge Discovery for Ubiquitous User Modeling '07*, 2007. `http://vasarely.wiwi.hu-berlin.de/K-DUUM07/`.

[4] F. Bonchi, Y. Saygin, V.S. Verykios, M. Atzori, A.Gkoulalas-Divanis, S. Volkan Kaya, and E. Savas. Privacy in spatio-temporal data mining. In Giannotti and Pedreschi [14], pages 253–276.

[5] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting.* Springer, 2nd edition, 2002.

[6] Noel Cressie. *Statistics for Spatial Data.* Wiley, 1993.

[7] DARPA. Darpa urban challenge rules. `http://www.darpa.mil/grandchallenge/rules.asp`, 2007.

[8] R.A. Davis, T. Lee, and G. Rodriguez-Yam. Structural break estimation for nonstationary time series models. *J. American Statist. Assoc.*, 11:229–239, 2006.

[9] O. Flasch, A. Kaspari, K. Morik, and M. Wurst. Aspect-based tagging for collaborative media organization. In Berendt et al. [3]. `http://vasarely.wiwi.hu-berlin.de/K-DUUM07/`.

[10] Joao Gama and Mohammed Garber, editors. *Learning from Data Streams: Processing Techniques in Sensor Networks*. Springer, 2007.

[11] David Gamarnik. Extension of the PAC framework to finite and countable markov chains. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 1999.

[12] Jing Gao, Wei Fan, Jiawei Han, and Philip S. Yu. A general framework for mining concept-drifting data streams with skewed distributions. In *SDM*, 2007.

[13] Lise Getoor and Ben Taskar, editors. *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[14] F. Giannotti and D. Pedreschi, editors. *Geography, mobility, and privacy: a knowledge discovery vision*. Springer, 2008.

[15] David Haussler. Probably approximately correct learning. In *National Conference on Artificial Intelligence*, pages 1101–1108, 1990.

[16] David P. Helmbold and Philip M. Long. Tracking drifting concepts by minimizing disagreements. Technical Report UCSC-CRL-91-26, University of California, 1991.

[17] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 97–106, San Francisco, CA, 2001. ACM Press.

[18] H. Kargupta, R. Bhargava, K. Liu, M. Powers, P. Blair, S.Bushra, J. Dull, K. Sarkar, M. Klein, M. Vasa, and D. Handy. Vedas: A mobile and distributed data stream mining system for real-time vehicle monitoring. In *Proceedings of the SIAM International Data Mining Conference*, Orlando, 2004.

[19] Ralf Klinkenberg and Thorsten Joachims. Detecting concept drift with support vector machines. In Pat Langley, editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 487–494, Stanford, US, 2000. Morgan Kaufmann Publishers, San Francisco, US.

[20] B. Kuijpers, M. Nanni, C. Körner, M. May, and D. Pedreschi. Spatio-temporal data mining. In Giannotti and Pedreschi [14], pages 277–306.

[21] L. Liao, D.J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 2007.

[22] Jennifer Neville and David Jensen. Relational dependency networks. In Getoor and Taskar [13], pages 239–268.

[23] S. Rinzivillo, S. Turini, V. Bogorny, C. Körner, B. Kuijpers, and M. May. Knowledge discovery from geographical data. In Giannotti and Pedreschi [14], pages 253–276.

[24] A. Schuster, R. Wolff, and B. Gilburd. Privacy-preserving association rule mining in large-scale distributed systems. In *Proceedings of CC-GRID'04, Chicago, Illinois*, 2004.

[25] Milton Severo and João Gama. Change detection with Kalman Filter and CUSUM. In *Discovery Science*, pages 243–254, 2006.

[26] Izchak Sharfman, Assaf Schuster, and Daniel Keren. A geometric approach to monitoring threshold functions over distributed data streams. *ACM Trans. Database Syst.*, 32(4), 2007.

[27] Stanford Racing Team. Stanford's robotic vehicle "junior": Interim report. `www.darpa.mil/GRANDCHALLENGE/TechPapers/Stanford.pdf`, 2007.

[28] Sebastian Thrun, Mike Montemerlo, and et al. Stanley, "The Robot that Won the DARPA Grand Challenge". *Journal of Field Robotics*, 23(9):661–622, 2006.

[29] Valdimir N. Vapnik. *Statistical Learning Theory.* Springer, 1998.

[30] Dennis Wegner, Dirk Hecker, Christine Koerner, Michael May, and Michael Mock. Parallel grid-applications with R: an industrial case study. submitted, 2008.

[31] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101, 1996.

[32] R. Wolff and A. Schuster. Association rule mining in peer-to-peer systems. In *Proc. of the IEEE Conference on Data Mining ICDM*, Melbourne, Florida, 2003.

[33] M. Wurst, K. Morik, and I. Mierswa. Localized alternative cluster ensembles for collaborative structuring. *ECML/PKDD 2006*, 2006.

[34] Feng Zhao and Leonidas Guibas. *Wireless Sensor Networks. An Information Processing Approach.* Morgan Kaufman, 2004.