DAGger: Clustering Correlated Uncertain Data

(to predict asset failure in energy networks)

Dan Olteanu
Dept. of Computer Science, University of Oxford
Oxford OX1 3QD, United Kingdom
Dan.Olteanu@cs.ox.ac.uk

Sebastiaan J. van Schaik
Oxford e-Research Centre
Oxford OX1 3QG, United Kingdom
Sebastiaan.vanSchaik@oerc.ox.ac.uk

ABSTRACT

DAGger is a clustering algorithm for uncertain data. In contrast to prior work, DAGger can work on arbitrarily correlated data and can compute both exact and approximate clusterings with error guarantees.

We demonstrate DAGger using a real-world scenario in which partial discharge data from UK Power Networks is clustered to predict asset failure in the energy network.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*

Keywords

Clustering, classification, uncertain data, probabilistic data, correlations, partial discharge, dagger.

1. CLUSTERING UNCERTAIN DATA

Recent years have witnessed a surge in the amount of digitally-born data. In many scenarios, this data is inherently uncertain or probabilistic, such as in automatic data extraction, image and voice detection (*e.g.*, processing handwriting, controlling mobile phones by voice), location detection, sensor networks, and measurement data [13]. Uncertain data calls for new processing approaches where uncertainty is explicitly accounted for, and it has led to a solid body of work on building probabilistic databases, such as MystiQ, Trio, and MayBMS. Albeit at a smaller scale, there is effort to adapt well-known data mining tasks to uncertain data, e.g., in discovery of frequent patterns and association rules [14], clustering [5], and classification [10]. However, to the best of our knowledge, prior work only considers limited probabilistic data models based on a simplifying independence assumption and circumvents the hardness of probability computation by the use of expected values and Monte-Carlo sampling. Expected values can lead to unintuitive results, for instance when data values and their probabilities follow skewed and non-aligned distributions. In case of correlated input events, the independence assumption can lead to results that are arbitrarily off from the ground truth.

In this paper we demonstrate DAGger, a novel approach to clustering correlated uncertain data. At its core, DAGger is a variant of the well-known k-medoids clustering algorithm adapted to accommodate uncertainty throughout the clustering process and probability computation.

DAGger has the following key features:

- The clustering outcome has a simple and intuitive meaning given by the possible worlds semantics: conceptually, it is equivalent to clustering in each possible world represented by the input data. This is in line with virtually all work in probabilistic databases [13] and thus allows for an easy integration of query processing and mining tasks.
- It supports arbitrarily correlated input data through a symbolic representation of probabilistic events. Complex events generated during the clustering process are expressible within the same representation formalism.
- At any stage in the clustering process, DAGger computes clustering events stating the membership of an object to a cluster and whether an object is a cluster medoid. The probability of such events can be computed exactly or approximately with absolute or relative error guarantees using a novel compilation technique of independent interest. This technique first represents the events of all objects and clusters at all iterations in a directed acyclic graph (DAG) where common factors are represented only once; each node in this graph thus represents an event. It then bulk-compiles all events into one decision diagram to the degree required to compute their probabilities.
- In addition to the events that are intrinsic to the clustering process, DAGger supports queries over the clustering output, e.g., to compute the probability that two given objects belong to the same cluster.

The purpose of the demonstration is to show how DAGger can be effectively used to cluster and classify sensor readings of a phenomenon in energy distribution networks, called *partial discharge*. This is used to predict asset failure in energy distribution networks. We will use real (anonymised) data from UK Power Networks consisting of known readings representing asset failures and new unclassified readings. These readings are naturally uncertain due to limited sensor sensitivity, hardware failure, and unreliable transmission channels [1, 3, 4]. By using DAGger, we can improve the quality

KDD'12, August 12–16, 2012, Beijing, China.

	(metadata) date/time	class	(uncertain) PD	load	(events) $\phi[o_i]$
...					
o_5	20/12 16:00	`OK`	5	140	$x_4 \wedge x_5 \wedge x_6$
o_6	20/12 17:00	`OK`	6	140	$x_5 \wedge x_6 \wedge x_7$
o_7	20/12 18:00	`OK`	9	150	$x_6 \wedge x_7 \wedge x_8$
o_8	20/12 19:00	`OK`	50	160	$x_7 \wedge x_8 \wedge x_9$
...					
o_{15}	10/01 03:00	`warn`	22	25	$x_{14} \wedge x_{15} \wedge x_{16}$
o_{16}	10/01 04:00	`warn`	20	25	$x_{15} \wedge x_{16} \wedge x_{17}$
o_{17}	10/01 05:00	`warn`	24	40	$x_{16} \wedge x_{17} \wedge x_{18}$
o_{18}	10/01 06:00	`warn`	25	50	$x_{17} \wedge x_{18} \wedge x_{19}$
...					
o_{25}	01/07 19:00	??	16	100	$x_{24} \wedge x_{25} \wedge x_{26}$
o_{26}	01/07 20:00	??	30	80	$x_{25} \wedge x_{26} \wedge x_{27}$

Table 1: Simplified data set. The labelled sensor readings are prior to a fault on January 11, 2011. The last two readings can be classified by clustering them with labelled data.

of the clustering for the set of new sensor readings and are able to distinguish spurious readings from readings that indicate an imminent failure of an asset (*e.g.*, a cable). The audience of this demonstration can explore the clustering outcome visually, as well as a ranked list of critical assets.

In the rest of this paper, we explain our demonstration scenario, show how `DAGger` clusters uncertain sensor readings, and detail on how the audience of our demonstration can interact with the system.

2. DEMONSTRATION SCENARIO: CLUSTERING PARTIAL DISCHARGE DATA

We demonstrate the clustering capability of `DAGger` in an application that predicts asset failure in energy networks.

2.1 Partial discharge

Partial discharge (PD) is an electrical discharge that does not fully bridge the insulation between two conducting electrodes. It has been identified as one of the major causes of long-term degradation and eventual failure of cables.

In order to minimise the customer minutes lost, energy distribution network operators (DNOs) are currently deploying sensors to monitor partial discharge activity in the distribution network, to be able to act preemptively [8, 9]. Unfortunately, monitoring partial discharge is not a straightforward task: the phenomenon is hard to detect, sensors often report spurious measurements and are prone to failure (as are the transmission channels).

The HiPerDNO project [15] aims to show the benefits of the introduction of cutting edge computational tools and techniques to improve electricity distribution network operations in partnership with UK Power Networks and other European DNOs.

2.2 Uncertain readings of PD and load

The data used to demonstrate our system is historical data on partial discharge activities in distribution networks, as well as records of network load and asset failure. It is gathered from two different types of sensors: (1) partial discharge sensors installed on switchgear and cables in substations of the distribution network, and (2) network load sensors in substations. By aggregating the number of partial discharge occurrences over the duration of an hour and subsequently pairing this value with the average network load during that hour, a data set like the one depicted in Table 1 is obtained. `DAGger` can deal with data with an arbitrary number of dimensions. For this demonstration each data point has the attributes load and partial discharge as described above. A single asset typically yields up to 24 data points per day.

`DAGger` interprets this data probabilistically. The rightmost column of Table 1 contains probabilistic events, i.e., arbitrary propositional formulas over independent Boolean random variables, that quantify the correlation and probability of readings. This probabilistic data formalism, whereby records are associated with probabilistic events, is called *probabilistic conditional tables*, or pc-tables for short, and is common in probabilistic databases [13].

We associate each sensor reading with a probabilistic event. The probability of that reading being true is thus given by the probability of the event. Each load and partial discharge reading has a probability of being accurate, which is inferred from sensor specification and measurement intervals in historical data. In the events used in `DAGger`, this is captured by independent Boolean random variables $x_0, \ldots x_{m-1}$.

Inference of probabilities and correlations can be done using many techniques, *e.g.* using inference in Bayesian networks or Markov Logic Networks [11, 6] and possibly based on the hardware specifications of the sensor manufacturer. In this specific application of `DAGger`, we construct a Markov chain in which each data point o_t at time t only depends on the data point o_{t-1} at time $t-1$. The conditional probabilities are then converted into events that can be processed by `DAGger`. As expected, consecutive sensor readings are strongly correlated [12]. In the example in Table 1, we used a sliding window of size three that yields events represented by conjunctions of three literals.

The possible worlds represented by our sample data are obtained by total assignments of the event variables. For instance, the worlds in which reading o_8 is correct are defined by assignments where x_7, x_8, and x_9 are *true*. Hence, the probability of o_8 being correct is given by the product of probabilities of these Boolean random variables being *true*. The readings o_5 and o_6 are positively correlated, since they both depend on x_5 and x_6. The readings o_8 and o_{15} are independent, since their events are independent.

2.3 Predicting asset failure

In order to predict asset failure, we perform the following procedure using `DAGger`:

1. Construct a clustering using both historical data (labelled readings), and the unclassified sensor readings.
2. For each unclassified reading, query the clustering for the probability that the reading is in the same cluster as one (or more) of the readings from the `warn`-labelled set. Depending on the type of labelled readings in the same cluster with the new readings, an expert user can also understand the type of failure.

After `DAGger` has constructed a probabilistic clustering, we can query it for the event that reading o_{25} is clustered into the same cluster as at least one reading from the set $R_{\texttt{warn}}$ with readings labelled `warn`. This query is constructed using clustering events (for clusters $C_1, \ldots, C_k$):

$$\phi[o_{25} \text{ is } \texttt{warn}] = \bigvee_{1 \le j \le k} \left(\phi\left[o_{25} \in C_j\right] \wedge \left(\bigvee_{o_w \in R_{\texttt{warn}}} \phi\left[o_w \in C_j\right] \right) \right)$$

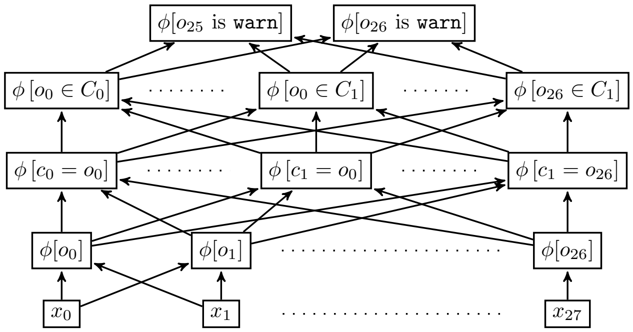


Figure 1: Partial DAG with five layers encoding clustering events for clusters C_0 (OK) and C_1 (warn) in our example.

In this expression, $\phi\,[o_i \in C_j]$ denotes the event that reading o_i belongs to cluster C_j. `DAGger` can cluster the data set from Table 1 and perform exact classification of o_{25} within seconds. The system can thus inform the user whether new readings indicate that a fault is imminent.

3. UNDER DAGGER'S HOOD

At the core of `DAGger` lies the well-known k-medoids clustering algorithm [2, 7], an unsupervised data mining technique that partitions a set of data points into k groups of similar points. It repeatedly assigns data points to clusters and re-elects cluster medoids, until convergence is reached.

In `DAGger`, the assignment of data points to clusters and selection of cluster medoids are probabilistic events. Therefore, a data point belongs to a cluster or is a cluster medoid with a certain probability. Conceptually, `DAGger`'s outcome is equivalent to applying the standard k-medoids algorithm in each possible world. However, `DAGger` cannot afford to enumerate the exponentially many possible worlds and perform a clustering in each of them. Instead, its computation is more symbolic as it traces the clustering events and uses them to compute probabilities of possible clusterings to any approximation degree. This symbolic computation can be orders of magnitude faster than the more extensional approach based on explicit enumeration of the possible worlds.

In this section, we give some details on how `DAGger` works.

Constructing events. The events $\phi[o_i]$ associated with input readings are the building blocks for events that are subsequently created by `DAGger` to express medoid selection and cluster assignment. At each clustering step, such events depend solely on events from the previous step. All events can be represented in a layered structure, where each layer corresponds to a clustering step and where we factor out common expressions. This layered factorisation, which is a directed acyclic graph (DAG), is key to the compact representation of the events, as it exploits the combinatorial nature of clustering computation. For instance, the event $\phi[o_i \in C_j]$ that reading o_i belongs to cluster C_j is expressed as a conjunction of the event $\phi[o_i]$ and of events for all cases in which a reading o_l is the medoid of cluster C_j and the distance from o_i to o_l is the smallest among all distances from o_i to the other readings. Figure 1 partially depicts such a DAG. Clustering events are expressed using conditional expressions that involve propositional formulas and distances, since the selection of new cluster medoids depends on input events and distances between data points. They are expressed in the algebraic structure of the semimodule defined by the tensor product $\mathbb{B}[\mathbf{X}] \otimes \mathbb{R}$ of the Boolean semiring $\mathbb{B}[\mathbf{X}]$ freely generated by the set $\mathbf{X}$ of input random variables and the SUM monoid of real numbers $\mathbb{R}$. For instance, the following expression represents the total distance-sum of a reading o_i to the readings $o_0, \ldots, o_{n-1}$ in cluster C_j:

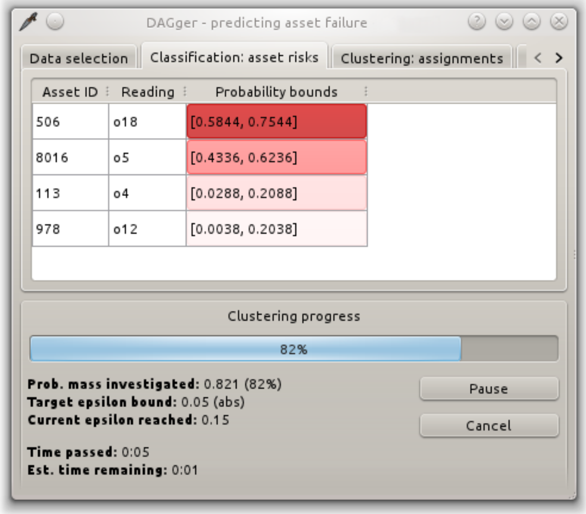


Figure 2: Screenshot of the system interface, showing an ordered list of assets of which sensor readings were classified as 'warning' with a high probability.

$$\Delta(C_j, o_i) = \sum_{0 \le a < n, a \ne i} (\phi[o_a \in C_j] \otimes d(o_i, o_a))$$

This expression represents a discrete probability distribution function over all possible distance-sums of o_i to readings in cluster C_j in a compact way. Indeed, for each possible truth assignment of random variables, this expression can yield a different distance-sum with a different probability. Such distance-sums are used inside inequalities to construct the events that describe medoid selection: the data point with the smallest distance-sum to the other points in the cluster is chosen as the new cluster medoid.

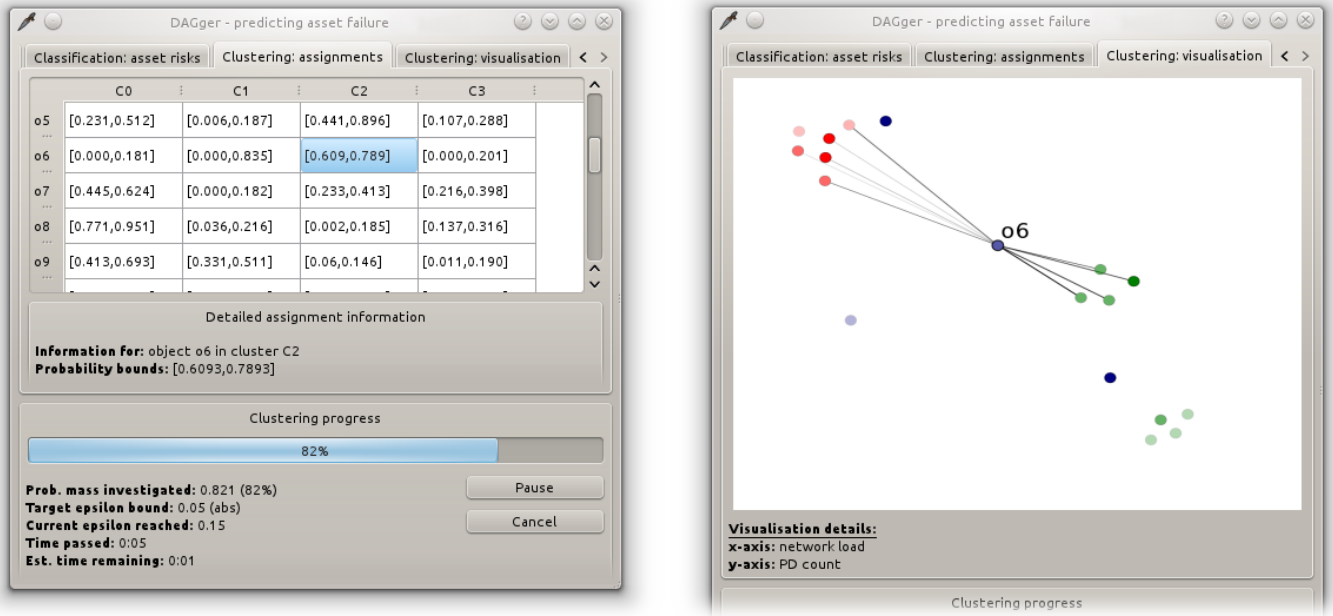
$$\phi[c_j = o_i] = \phi[o_i \in C_j] \wedge \bigwedge_{0 \le a < n, a \ne i} (\Delta(C_j, o_i) \le \Delta(C_j, o_a))$$

In the absence of the semimodule $\mathbb{B}[\mathbf{X}] \otimes \mathbb{R}$, these inequalities would only be expressible as propositional events of size exponential in the number of objects (or readings).

Once the clustering events are constructed, classification queries such as the one described in Section 2.3 are added to the DAG. The DAG in Figure 1 includes classification queries for objects o_{25} and o_{26} from Table 1 in the top layer.

Probability computation. `DAGger` uses a novel bulk compilation strategy to efficiently compute the probability of the events represented in a layered DAG structure. The core idea of this compilation technique is Shannon expansion: given a Boolean random variable x, the probability $P(\Phi)$ of an event Φ is the weighted sum of probabilities of the events $\Phi|_x$ and $\Phi|_{\neg x}$ obtained by setting x to *true* and respectively to *false* in Φ, i.e., $P(\Phi) = P(x) \cdot P(\Phi|_x) + P(\neg x) \cdot P(\Phi|_{\neg x})$.

The challenges faced by `DAGger` are to extend Shannon expansion (1) to work well on sets of events represented in a DAG structure and on semimodule expressions, and (2) to incrementally compute approximations to any degree.



(a) Probabilistic assignment of data points to clusters

(b) Visualisation of probabilistic clustering

Figure 3: Screenshots of the system interface, showing two different views of the clustering result.

4. DRIVING DAGGER

`DAGger` has a graphical user interface to present the clustering outcome, as well as the incremental probability computation of the clustering events. Screenshots of this interface are given in Figures 2 and 3.

On the first tab, the user can make a selection of the input data (both labelled and unlabelled data) which is to be analysed by `DAGger`. After the data analysis has started, the user can monitor the progress and examine the results.

On the tab "Classification: asset risks" (Figure 2), the system displays the results of the classification of the unlabelled data points. It lists the assets that were classified into the `warn` category in decreasing order of probability.

The tab "Clustering: assignments" (Figure 3a) shows the probabilistic assignment of data points to clusters.

The tab "Clustering: visualisation" (Figure 3b) presents the user with a visual representation of the uncertain clustering. By selecting a sensor reading (in this case: o_6), the interface will show the user the probability that the data point will be clustered into the same cluster as the closest neighbouring data points: the darker the line that connects o_6 to another data point, the higher the probability that the two data points end up in the same cluster.

Throughout the interface of the system, the user will see the exact lower and upper bounds of the probabilities, whilst the probabilities are being established. Unless `DAGger` is configured to compute approximate probabilities, the system will present the user with the exact probabilities once the lower and upper bounds have converged.

Acknowledgements

The authors would like to thank Peter Lang, Matthieu Michel, and Koen Tavernier (UK Power Networks, IPEC) for providing us with the data. This work is supported by EC FP7 HiPerDNO project under grant 248235 and EPSRC ADEPT project under grant EP/I000194/1.

5. REFERENCES

[1] C. C. Aggarwal. *Managing and Mining Uncertain Data*, volume 35 of *Advances in Database Systems*. Kluwer, 2009.
[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.
[3] R. Cheng and S. Prabhakar. Managing uncertainty in sensor database. *SIGMOD Rec.*, 32:41–46, 12 2003.
[4] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *VLDB*, 2004.
[5] F. Gullo, G. Ponti, and A. Tagarelli. Clustering uncertain data via k-medoids. In *SUM*, 2008.
[6] A. Jha and D. Suciu. Probabilistic databases with markoviews. In *VLDB*, 2012.
[7] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Berkeley Symp. on Mathematical Statistics and Probability*, 1967.
[8] M. Michel. Innovative asset management and targeted investments using on-line partial discharge monitoring and mapping techniques. In *CED*, 2007.
[9] M. Michel and C. Eastham. Improving the management of MV underground cable circuits using automated on-line cable partial discharge mapping. In *CEED*, 2011.
[10] B. Qin, Y. Xia, R. Sathyesh, S. Prabhakar, and Y. Tu. urule: A rule-based classification system for uncertain data. In *ICDM Workshops*, 2010.
[11] M. Richardson and P. Domingos. Markov logic networks. *Machine learning*, 62(1):107–136, 2006.
[12] S. Sathe, H. Jeung, and K. Aberer. Creating probabilistic databases from imprecise time-series data. In *ICDE*, 2011.
[13] D. Suciu, D. Olteanu, C. Ré, and C. Koch. *Probabilistic Databases*. Morgan & Claypool Publishers, 2011.
[14] L. Sun, R. Cheng, D. W. Cheung, and J. Cheng. Mining uncertain data with probabilistic guarantees. In *KDD*, 2010.
[15] G. Taylor, D. Wallom, S. Grenard, A. Yunta Huete, and C. J. Axon. Recent developments towards novel high performance computing and communications solutions for smart distribution network operation. In *ISGT*, 2011.