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...
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.

<table>
<thead>
<tr>
<th>(metadata)</th>
<th>(uncertain)</th>
<th>(events)</th>
</tr>
</thead>
<tbody>
<tr>
<td>date/time</td>
<td>PD load</td>
<td>φ[\omega]</td>
</tr>
<tr>
<td>01/07 01:00</td>
<td>??</td>
<td>x_24 \land x_25 \land x_26</td>
</tr>
<tr>
<td>01/07 02:00</td>
<td>??</td>
<td>x_25 \land x_26 \land x_27</td>
</tr>
</tbody>
</table>

...
In this expression, \( \phi [o_1 \in C_j] \) denotes the event that reading \( o_1 \) belongs to cluster \( C_j \). D\text{AGger} 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 D\text{AGger}\'S HOOD

At the core of D\text{AGger} 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 D\text{AGger}, 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, D\text{AGger}\'s outcome is equivalent to applying the standard k-medoids algorithm in each possible world. However, D\text{AGger} 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 D\text{AGger} works.

Constructing events. The events \( \phi[o_i] \) associated with input readings are the building blocks for events that are subsequently created by D\text{AGger} 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 layer 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 \( \phi[o_1 \in C_j] \) that reading \( o_1 \) belongs to cluster \( C_j \) is expressed as a conjunction of the event \( \phi[o_1] \) and of events for all cases in which a reading \( o_1 \) is the medoid of cluster \( C_j \) and the distance from \( o_1 \) to \( o_1 \) is the smallest among all distances from \( o_1 \) 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 \( \mathbb{B}[X] \otimes \mathbb{R} \) of the Boolean semiring \( \mathbb{B}[X] \) freely generated by the set \( X \) 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 \( o_1 \) to the readings \( o_0, \ldots, o_{n-1} \) in cluster \( C_j \):

\[
\Delta(C_j, o_1) = \sum_{0 \leq a < n, a \neq i} \phi[o_a \in C_j] \otimes d(o_1, o_a)
\]

This expression represents a discrete probability distribution function over all possible distance-sums of \( o_1 \) 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.

\[
\phi[o_j = o_i] = \phi[o_i \in C_j] \land \bigwedge_{0 \leq a < n, a \neq i} (\Delta(C_j, o_i) \leq \Delta(C_j, o_a))
\]

In the absence of the semimodule \( \mathbb{B}[X] \otimes \mathbb{R} \), 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. D\text{AGger} 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 \( \Phi \) is the weighted sum of probabilities of the events \( \Phi_x \) and \( \Phi_{\neg x} \) obtained by setting \( x \) to \( \text{true} \) and respectively to \( \text{false} \) in \( \Phi \), i.e.,

\[
P(\Phi) = P(x) \cdot P(\Phi|_x) + P(\neg x) \cdot P(\Phi|_{\neg x})
\]

The challenges faced by D\text{AGger} 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.
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: o6), 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 o6 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