White paper

virtana

Exploring AIOps: Cluster Analysis for Events

AIOps, i.e., artificial intelligence for IT operations, has become the latest strategy du jour in the IT operations management space to help address and better manage the growing complexity and extreme scale of modern IT environments. AIOps enables some unique and new capabilities on this front, though it is quite a bit more complicated than the panacea that it is made out to be. However, the underlying AI and machine learning (ML) concepts do help complement, supplement and, in particular cases, even supplant more traditional approaches to handling typical IT Ops scenarios at scale.

So, how does one apply and leverage AIOps to better manage I&O environments and activities in practice? Gartner, in their Market Guide for AIOps Platforms, recommends an incremental approach. Namely, start small with less critical applications, and apply the more straightforward AIOps aspects, such as categorization, correlation and anomaly detection, to start deriving value and to drive better business outcomes for the use cases under consideration.

An AIOps platform has to ingest and deal with multiple types of data to develop a comprehensive understanding of the state of the managed domain(s) and to better discern the push and pull of diverse trends in the environment, both overt and subtle, that may destabilize critical business outcomes. In this white paper, we will take a look at an AIOps approach to handling one of the fundamental data types: events.

Events

An event is a record of a notable change in state in the environment. This could be a new service or device coming online, the measured state of a component or resource breaching a threshold, an application consistently failing to connect to an external service — the possibilities are endless. Also, even though some events are one-offs, most events are repeated for the duration of the state they are recording based on the evaluation cycle of the event source. This translates to a large volume of events from even a small managed environment and potentially orders of magnitude more from data center and cloud-scale environments.

The source of the event usually provides a measure of its base-level importance (from the source's perspective) as a severity measure ranging from "info" to "critical." How truly "notable" an event is, however, is somewhat subjective, and it usually depends on the evaluation context. For example, a "ping down" event provides less useful information in most cases but is notable if you are trying to diagnose an unresponsive application on that host That is, knowledge about the problem and its domain also plays an important role in determining the noteworthiness of a particular event.

An event may have a varying degree of context associated with it, captured as metadata on the event. This may range from something minimal, such as the IP address for a ping event, to a comprehensive capture of the relevant device state at the time of the event. This context is useful in determining the relevance of an event and also in correlation exercises. In addition to the formal context, an event also has text fields, such as the "summary" and "notes" fields, that provide informal context for better interpretation and correlation.

An event has other properties, too, but these are the ones germane to the discussion in this white paper.

Efficiently handling the event volume, categorizing and prioritizing the events to look at first to minimize operational downtime, and identifying relevant events for a particular incident or problem-resolution activity are some of the main event-related tasks that need to be addressed by an AIOps platform. Traditional methods to address these areas include event deduplication to manage volume, prioritizing based on event severity to identify high-impact events, and using rules based on tribal knowledge and domain context to identify related events. These methods have worked with varying degrees of success in siloed environments with defined rules for known domains and event types, but they do not scale well to the demands of the modern enterprise. Here, the managed domains are more dynamic, varied, complex and not necessarily wellunderstood, with the resulting difficulty in predetermining rules to handle all cases. Traditional methods are also hobbled in dealing with the extreme scale of new management realms,

e.g., cloud environments, ephemeral services, IoT, global-scale data centers, et al., which translate to more complexity and ever-higher event volumes.

AI/ML enable additional approaches to dealing with this scale, dynamic change and complexity. In the event context, this translates to replacing static rule-based analysis with more adaptive ML-based methods. Specifically, an unsupervised learning approach that doesn't require training sets with predetermined outcomes but, rather, lets the system learn from the data it is exposed to might be an appropriate choice to help make headway with some of the issues with event handling at scale in complex and changing environments outlined above.

Next, we will take a look at the benefits of applying one such unsupervised approach to events: cluster analysis.

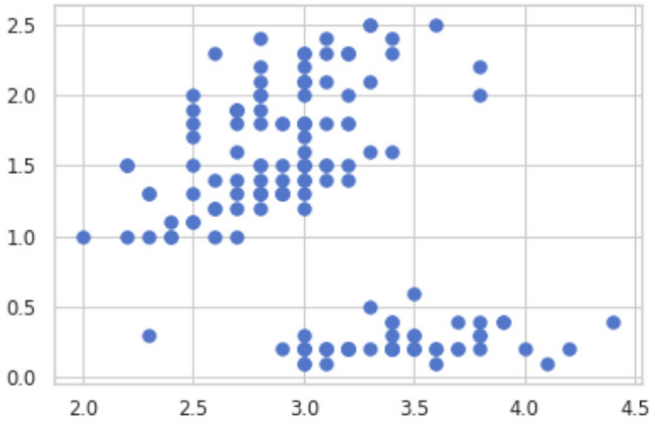
Clustering

Cluster analysis, or clustering, is a core unsupervised learning technique that attempts to group the data points in a dataset into groups, or clusters, such that data points in the same cluster are similar to one another while being sufficiently different from data points in another cluster.

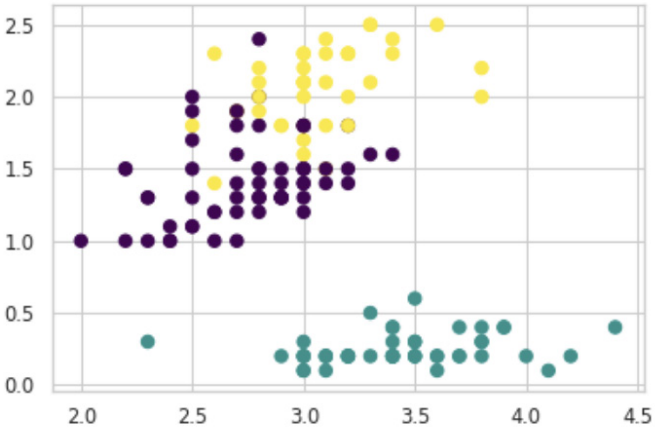
To aid in this analysis, representative features are extracted from the data points and subsequently evaluated for (dis) similarity. Numerical features are evaluated via a distance function that provides a measure of how close the data point is to the centerpoint of the cluster. Qualitative features are evaluated using a similarity function to provide an equivalent assessment vis-a-vis the centerpoint. A data point is assigned to a group based on the properties of least distance to and most similarity with the features of the representative centerpoint of the cluster. The accuracy of the clustering results can be evaluated with internal and external test data and validation functions corresponding to the algorithm employed.

To illustrate this, let's take a look at applying cluster analysis to the popular Iris dataset, available in the public domain via Kaggle and other sources. This dataset includes petal and sepal measurements for three species of the Iris flower. A plot of the raw data using the sepal width and petal width as the feature dimensions is depicted below.

A quick observation is that a group of data points is naturally separated from the rest and one could surmise that they potentially describe the same species. However, the rest of the data points are not so well delineated. The result of applying cluster analysis to this dataset is depicted below.



This identifies three clusters, representing the three different species that were incorporated by the dataset. It confirms the hypothesis of the separated group being members of the same species, and it also highlights the subtle demarcation between the other two species. This was a simple example, but it helps illustrate the benefit of applying this fundamental clustering technique to make sense of the data, especially in the context of the complex domains typically managed by AIOps solutions where the grouping may not be very apparent.



Common clustering approaches include partitioning based algorithms such as K-Means, which implements the centerpoint based clustering described earlier after K clusters are picked, and hierarchy-based algorithms such as Chameleon, which leverages relationship information from a nearest neighbor graph to group data points into small clusters and smaller clusters into larger ones. There is no dearth of clustering approaches and corresponding algorithms that can be applied to a diverse set of I&O use cases, and the choice is heavily dependent on the properties of the dataset and the desired outcomes.

An AIOps platform will typically employ one or more of these clustering approaches and algorithms, depending upon the use case(s) and the data context, to yield the optimal clustering outcome in support of the business use case(s). An advanced AIOps platform like Virtana will also bring to bear additional context such as model topology to further disambiguate and refine the clusters and to drive advanced use cases.

The net result of cluster analysis is that the input dataset is now grouped into a set of clusters. The clusters are determined via a system-learned relationship function influenced by feature-driven criteria and the clustering algorithm. These clusters don't always map directly to a domain concept, but the hope is that successful clustering will yield some interesting relationships that can, in turn, be fed into other AI/ML processing flows or used directly as warranted by the use case(s).

Let's take a look at how cluster analysis helps implement the Gartner-recommended incremental aspects of AIOps — namely, categorization, correlation and anomaly detection. This is especially with regard to helping address the aforementioned problems with efficiently handling the event volume, categorizing and prioritizing the events to look at first to minimize operational downtime, and identifying relevant events for a particular incident or problem resolution.

Applying Cluster Analysis to Events

Cluster analysis can be applied to a number of different event use cases. A simple case is an alternative to or an augmentation of the traditional event deduplication approach, wherein individual instances of the same duration-based event are grouped into the same cluster. In this case, the features used for the analysis include the event summary and other context fields that help contribute to identifying the underlying condition as well the time stamp. The clusters represent event durations, and the members represent event instances recording updates for that particular episode. A summary event for a cluster can represent the event episode referenced by all of the instances in that cluster, helping to reduce event volume.

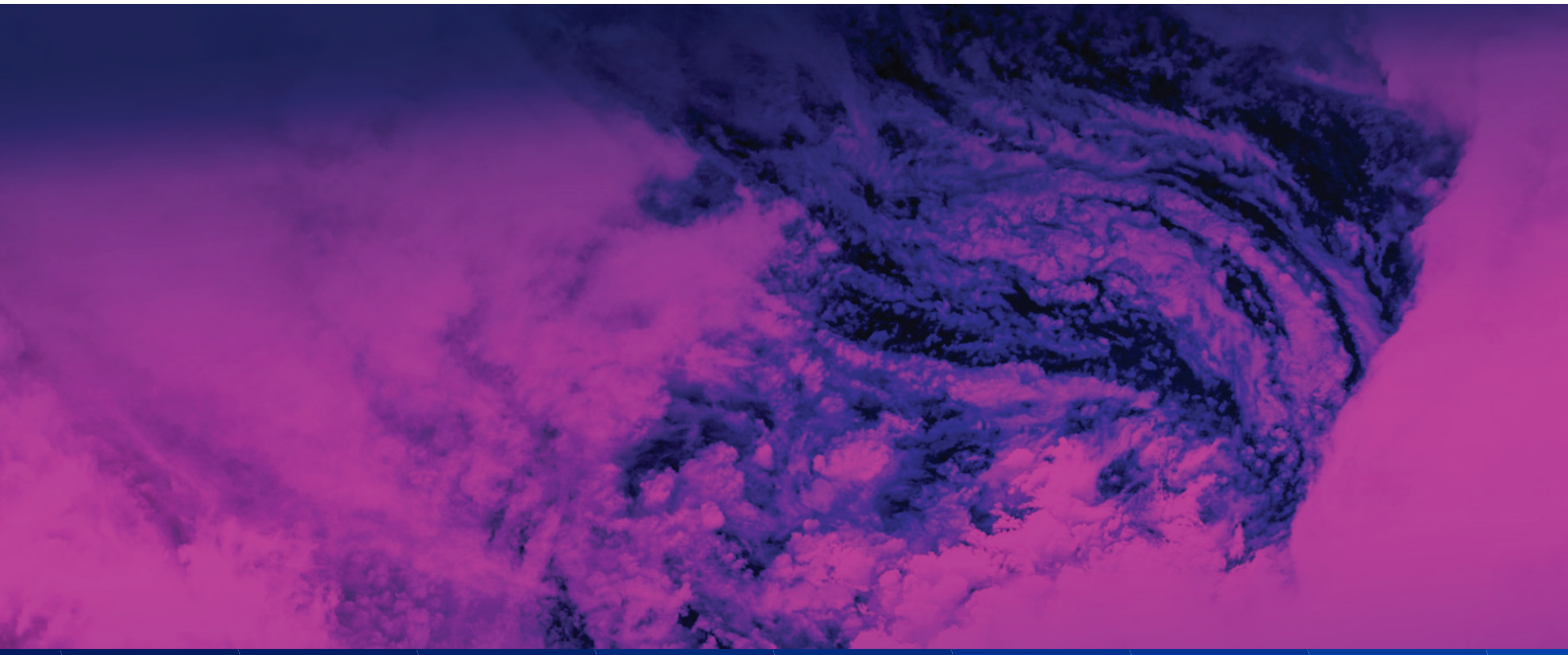
A related use case is to group similar events together. This is a more general form of the previous case — wherein the clusters represent groups of similar events, not instances of the same one — and is better classified as a correlation activity. Here, similarity is informed by the summary and metadata context rather than primarily by identification-oriented features. A cluster then represents a group of events that can be handled conceptually as a batch for further consideration as appropriate, rather than being required to look at each one individually. That is, summarization of the similar events in each cluster helps to better partition and address scale issues with large event volumes.

Cluster analysis can be applied to categorize events based on their underlying types. This goes beyond the coarse typing obtained from an "event class" field, giving more weight to contextual information to detect event types. This is very useful when receiving event data streams from different third party data sources with inconsistent levels of typing information. This use case may require an operator to assign types or labels to the generated clusters. The model can then be applied at runtime to classify an event instance to better identify automated analysis and handling and to, consequently, better handle event volume.

An advanced application of cluster analysis is to better identify relevant events for a particular incident or problemresolution activity. This is a correlation exercise, where event context is augmented with model topology and analyzed for relevance to a specific set of entities under

consideration for problem resolution. Though standard clustering, itself, can provide a rough sense of related events (like the similar events use case above), being model informed helps increase the correlation accuracy and, hence, the usefulness of the proposed grouping in narrowing down the scope of which events to look at to drive quicker time to resolution.

What about anomaly detection? Cluster analysis is applicable on this front, too. Events that are outliers, i.e., that do not belong to any cluster, are potentially "anomalous" in that they either flag new conditions in the managed environment not seen earlier or are indicative of the clustering criteria not correctly accounting for the dataset. In either case, these events would typically require operator assessment and, potentially, a new model that correctly accounts for the same.



info@virtana.com | +1-408-579-4000 | virtana.com