

# Pattern of life analysis for diverse data types

Clay D. Spence,\* Ben Southall, Alex Tozzo, Jun Hu, Thomas Kover, Joseph Ferraro, Hui Cheng  
SRI International, 201 Washington Road, Princeton, NJ USA 08540-6449

## ABSTRACT

SRI has developed a system to automatically analyze the Pattern of Life (PoL) of ports, routes and vessels from a large collection of AIS data. The PoL of these entities are characterized by a set of intuitive and easy to query semantic attributes. The prototype system provides an interface to ingest other types of information such as WAAS (Wide Area Aerial Surveillance) and GDELT (Global Database of Events, Language, and Tone) to augment knowledge of the Area of Operations. It can interact with users by answering questions and simulating what-if scenarios to keep human in the processing loop.

**Keywords:** Pattern of life, AIS

## 1. INTRODUCTION

Current technology presents us with large complex multimodal geospatial data sets. Often this data was not gathered to investigate a particular entity, and yet it reflects the activity of many entities. In geospatial data many of these entities present themselves as sites and movers. SRI has developed methods for detecting patterns of life (PoL) in such data sets, with the goal of providing users a detailed overview of the activities of the various entities in a region of interest (ROI), and the interactions between them.

SRI's PoL system currently has several analysis components. The first is data collection and exploration. Low-level description of activity can be extracted from raw data. For example, arrivals and departures at a site can be extracted from raw information of locations of movers. Counts of such events in given time intervals are then a low-level description of activity. The system then extracts higher-level descriptions and presents the information in semantic terms.

The system includes several components for characterizing sites. The interaction of movers with sites is analyzed to cluster the sites and infer site function (Section 2.1). The system also clusters objects and entities based on events time series using a Hidden Markov Model (HMM) to identify and detect different states of activity, and generate a description of the site's typical behavior (Section 2.2). It is common to have other attributes of sites from other sources, yet some attributes are missing. Or system includes a component for inferring these missing attributes (Section 2.4).

SRI's PoL analysis also includes a number of components for analyzing movers. One function is anomaly detection. This is spread over several of the components, since they model normal behaviors of both sites and movers. Multiple types of anomalies can be uncovered in an unsupervised fashion by modeling typical behaviors. There is also a component to detect anomalous behavior of movers (Section 2.5). The history of individual movers is also analyzed to characterize their typical behavior (Section 2.6), and the set of movers is clustered into different functional classes based on behavior (Section 2.7)

The system analyses larger scale behavior by examining the relationships between entities in the network to improve the reasoning of the changes in state. Correlation between events and patterns is made in an attempt to reason over changes in events. (Section 2.3).

We present the system architecture in Section 3. In Section 4 we present results on real data sets.

Further author information: (Send correspondence to C.D.S.)

\*clay.spence@sri.com

## 2. MODELS

The goal of the Pattern of Life analysis is to model the behaviors of entities, with the aim of characterizing the types of behaviors, detect anomalous behaviors, and so on. While this statement is very general, with our data we developed tools for modeling what can be called movers and sites. Movers in our data include road vehicles in WAAS data and vessels in AIS data. Sites in our data include specific store, houses, and other locations in WAAS data, and ports and possibly routes in AIS data.

We studied two broad objectives:

- Develop models of normal entity behavior and discover anomalous activities.
- Develop functional models of entities and classify them based on the functional models.

### 2.1 Queueing Models for Functional Classification of Locations

We propose a functional classification model for locations and sites. Functional classification models and classifies sites look at how frequently and at what times they are visited by movers such as vehicles and pedestrians. We have developed a Site Patronal Activity Model (SPAM) that uses the ingress and egress of movers from sites, as observed from WAAS data and later applied to AIS data, for functional classification. The parameters of the queueing models developed for sites are then used as features for clustering.

#### Site Patronal Activity Model (SPAM)

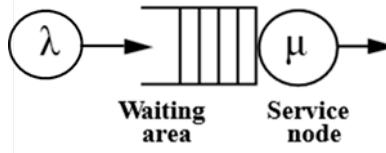


Figure 1 An M/M/1 single server queue model

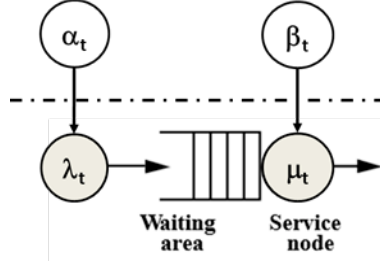


Figure 2 An  $M(\lambda_t)/M(\mu_t)/1$  queue is shown with latent variables  $\lambda_t$  and  $\mu_t$  exerting non-homogeneous temporal influences on  $\lambda_t$  and  $\mu_t$ .

We model each site as a mathematical queue (Figure 1). In the (extended) Kendall's notation, a queue is defined by a model using six factors A/S/c/K/N/D (sometimes a short notation A/S/c is used) where A denotes the arrival process; S the service time distribution, c the number of servers at the node, K the number of places (capacity) in the system, N the population from which arriving entities originate and D the queuing discipline. For details on notation and different kinds of queues, please refer to [3]. An M/M/1 queue is shown in Figure 1 where M stands for a Markovian process: the arrival process A is homogeneous Poisson process, M and the service process is homogeneous Exponential process, M with a single server in the system.

In this work, we model each site as an  $M(\lambda_t)/M(\mu_t)/1/\infty/\infty/\text{SIRO}$  queue where M stands for a Markovian process and SIRO stands for Serve In Random Order. More specifically, we model the arrival process A as a non-homogeneous Poisson process,  $M(\lambda_t)$  with a time-varying arrival rate parameter  $\lambda(t)$  and the service process as a non-homogeneous Exponential process,  $M(\mu_t)$ , with a time-varying service time  $\mu(t)$ . This model is shown in Figure 2. For this discussion, it suffices to note that the additional two nodes in the figure modeled by random variables  $\alpha_t$  and  $\beta_t$  model latent variables which systematically modify the two process node parameters. Other details are immaterial to the current discussion.

## 2.2 HMM/BPN models – PoL from Event Counts

A common type of data is counts of events occurring in equally-spaced intervals of time, e.g., days. Since these counts are non-negative integers, they differ in character from many kinds of sampled signals. Gaussians may be appropriate models at very high counts, but if the typical count is around one or less, a Gaussian model is inappropriate. The Poisson distribution may seem like a more appropriate model, but it is constrained to have variance equal to the mean. It cannot describe low-variance sequences, e.g., when there is usually one event per day, or higher variance sequences, e.g., “contagious” events. To allow for this, we used binomial distributions for low-variance behavior, Poisson distributions for equal mean and variance, and negative binomial distributions for high-variance behavior (Figure 3). These can be treated as a single type of distribution parameterized by the mean and variance, since they are mathematically related. For brevity, we’ll refer to this as the BPN distribution. An alternative to BPN distributions is the Conway-Maxwell distribution (Shmueli, Minka, Kadane, Borle, & Boatwright, 2005). This also has the property that the mean and variance are independent. We chose not to use this, since the relationship between the distribution parameters and the mean and variance appears to be not straightforward.

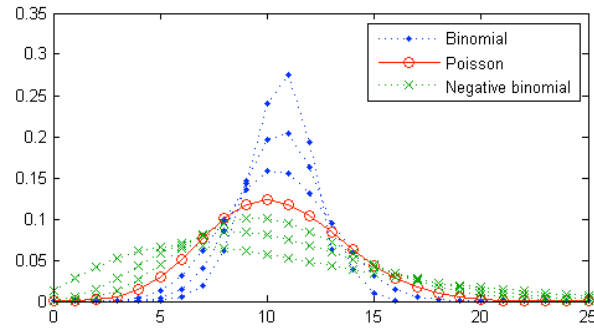


Figure 3. BPN distributions. These all have a mean of 10.5, but the variances differ. The dotted curves with larger dots at the integers are binomial distributions with variance less than the mean. The solid curve with circles at the integers is a Poisson distribution, with variance equal to the mean. The dotted curves with x’s at the integers are negative binomial distributions, with variance greater than the mean.

A potentially useful aspect of activity is how it has changed over time. Given the highly irregular nature of the data we were treating (Figure 4 and Figure 5) we developed activity descriptions in terms of periods of behavior, using hidden Markov models (HMMs) with BPN distributions for the emission probabilities. We fit the models using the Baum-Welch algorithm, and chose the number of states with the  $\chi^2$ -test to decide whether adding a state significantly improved the likelihood. We treat a time interval during which one particular state is the most likely as a period of constant behavior.

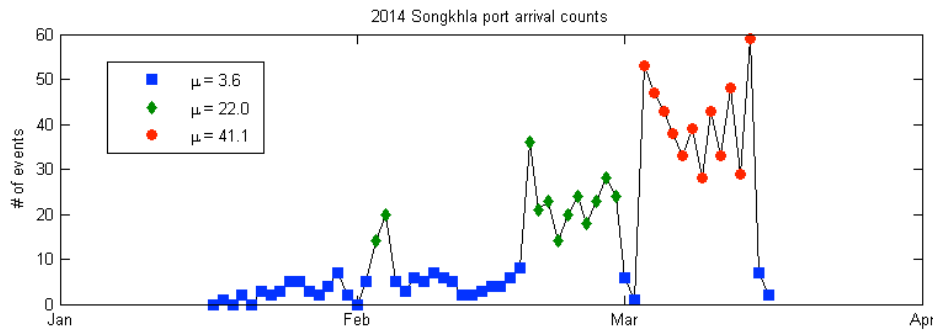


Figure 4. Plot of daily arrival counts for Songkhla (near Bo Yang). This port has the strongest trend of those in our set. The symbols indicate states assigned by the HMM-BPN model. The state means are given in the legend. For all three states the variance is around 1.5–2 times the mean.

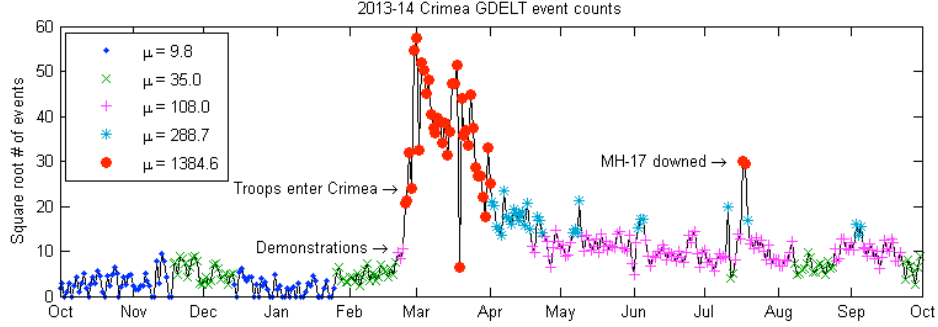


Figure 5. GDELT daily event counts in the Crimean peninsula from October 2013 to September 2014, and assignment of HMM-BPN model states. Note that the square root of the number of counts is plotted, to make the behavior on lower-count days more evident. The symbols show the most-likely state on each day, and the means of each state are given in the legend. In every state the model’s variance is much larger than the mean.

### 2.3 Network model for traffic Pattern-of-Life

The network model for traffic PoL is used to analyze the patterns in the maritime traffic flow within a region, in particular, number of vessels that arrives at and departs from each node or from node to node. The goals of this model include

- To build a simple enough model that yet can still explain and/or predict the ups and downs of traffic flow;
- To check whether any aspect of the traffic PoL in a region has changed;
- And to predict the impact of a “what-if” scenario on the regional traffic PoL.

In our network model, each port and its bounding box inside Region of Interest (RoI) is a node. The rest ports outside of RoI are considered as a single “super” node. A link is a sea route that connects either a pair of nodes or a node and the outside “super” node. Please note that links are directional, the link from A to B is not the same as the link from B to A.

The major parameters used in our network model are

- Transition matrix  $P_{ij}$ , which describe how likely a vessel will go from one node to another.
- Service time  $\Delta_i(\tau)$ , which describe the time vessels stay at each node.
- Travel time  $\Delta_{ij}(\tau)$ , which describe the time vessels spend to move from one node to another.

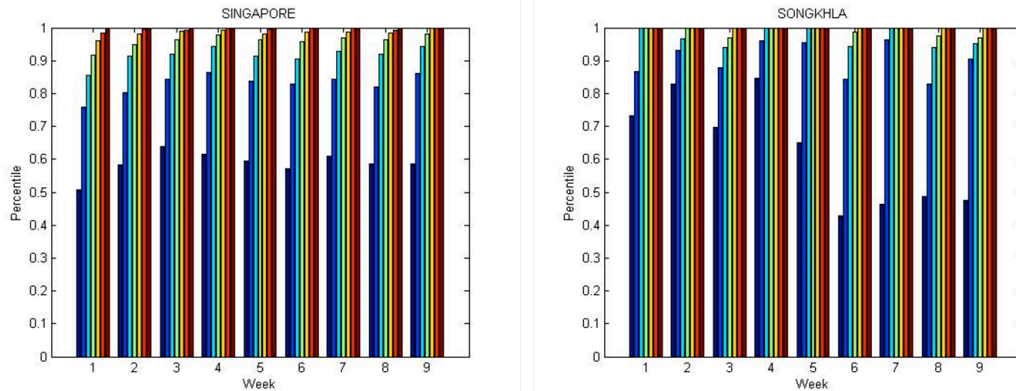
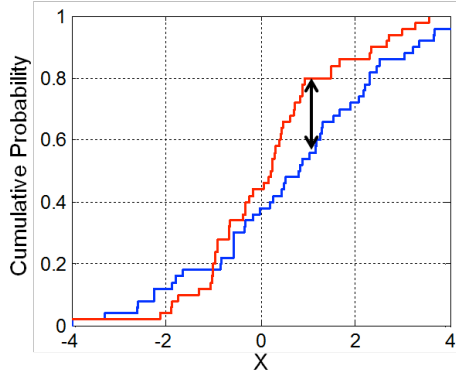


Figure 6. Examples of the service time at a node. Left: Singapore, which is quite stable from week to week. Right: Songkhla, a noticeable change between week 5 and 6.

Each parameter mentioned above is in fact a random variable with some distribution. We can use the Kolmogorov-Smirnov goodness-of-fit test [7] to compare two samples of any network parameter from different time periods to verify

if its distribution has changed. The probability  $\Pr(K \leq x)$  computed in the test is an indicator on the likelihood that the two samples are drawn from the same distribution.



K-S statistic

$$D_{m,n} = \sup_x |F_m(x) - G_n(x)|$$

*Kolmogorov distribution*

$$K = D_{m,n} \sqrt{\frac{mn}{m+n}}$$

$$\Pr(K \leq x) = 1 - 2 \sum_{k=1}^{\infty} (-1)^{k-1} e^{-2k^2 x^2}$$

Figure 7. Illustration of two-sample Kolmogorov-Smirnov test. Red and blue lines each correspond to an empirical distribution function, and the black arrow is the K-S statistic.

Assuming that every departure from a node corresponds to an earlier arrival at the same node, and further, total arrivals at the node in the past,  $A_i(t-1)$ ,  $A_i(t-2)$ , ... are all known, the expected total departures from the node can be estimated using the following equation

$$D_i(t) = \sum A_i(t-\tau) \Delta_i(\tau)$$

Now consider the null hypothesis and the alternate hypothesis:

$H_0$ : The difference between expected total departures and observed total departures is **not** statistically significant,

$H_1$ : The difference between expected total departures and observed total departures is statistically significant.

If we can accept  $H_0$  at a given confidence level, any change in observed total departures can be explained by a change in earlier observed total arrivals. Otherwise we have to reject  $H_0$ , which means something abnormal has happened at this node.

Assuming that every arrival at a node corresponds to an earlier departure from some node in the network, either the node itself or another node, and further, total departures from every node in the past,  $D_i(t-1)$ ,  $D_i(t-2)$ , ... are all known, the expected total arrivals at the node can be estimated using the following equation

$$A_i(t) = \sum \sum P_{ij} D_i(t-\tau) \Delta_{ij}(\tau)$$

Again consider the null hypothesis and the alternate hypothesis:

$H_0$ : The difference between expected total arrivals and observed total arrivals is **not** statistically significant,

$H_1$ : The difference between expected total arrivals and observed total arrivals is statistically significant.

If we can accept  $H_0$  at a given confidence level, any change in observed total arrivals can be explained by a change in earlier observed total departures, somewhere in the network. Otherwise we have to reject  $H_0$ , which means something abnormal has happened at some node(s) which is connected to this node in question.

## 2.4 Filling in incomplete data

Within a database of entities and their attributes for each, it is common to have unknown values for many of these attributes. A practical problem then is to infer the unknown attributes from the known, with some estimate of the uncertainty. The algorithm must deal with a variety of features of such attributes: They can have different types, e.g., binary, real valued, categorical, and so on. The frequency with which they are missing can be quite different for different attributes.

We could attempt to build predictors of a specific subset of attributes from another specific subset, but this is too limiting, given that variability in the presence of attributes. We chose to build a probability model of the attribute

vectors. We are not aware of distributions analogous to the multi-variate Gaussian that could handle attributes of inhomogeneous type and model statistical dependencies between them. If all attributes have a finite set of possible values one might consider enumerating all possible attribute vectors and building a categorical distribution on this set. However, it is easy to have extremely large sets of possible attribute vectors, so we did not pursue this. We could use a product of distributions, one for each of the attributes, but this assumes that the attributes are independent, and so ignores relationships between them. To reintroduce dependencies, we used a mixture of product distributions, which we'll call a MoPODD, for "Mixture of Products of One-Dimensional Distributions." The distribution is  $P(\mathbf{x}) = \sum_c P(c)P(\mathbf{x} | c)$ ,

where  $\mathbf{x}$  is the vector of attribute values and  $P(\mathbf{x} | c) = \prod_i P(x_i | c)$ .

The variety of attribute types is easily handled, since there can be a different type of factor distribution for each attribute. Missing attributes are easily handled, since they should be marginalized out. This is just a sum over their possible values, which replaces the factor for a missing attribute with a 1. So if  $\mathbf{x}_O$  is the vector of observed attributes and  $\mathbf{x}_M$  is the vector of missing attributes, then

$$P(\mathbf{x}_O) = \sum_{\mathbf{x}_M} P(\mathbf{x}) = \sum_{\mathbf{x}_M} \sum_c P(c)P(\mathbf{x} | c) = \sum_c P(c) \sum_{\mathbf{x}_M} P(\mathbf{x} | c) \quad (5)$$

$$= \sum_c P(c)P(\mathbf{x}_O | c) \quad (6)$$

The distribution  $P(\mathbf{x}_O | c)$  is just a product over the observed attributes. Expression (3) and (6) enable maximum likelihood fitting when the training data has missing attributes. In addition, these expressions let us infer missing attributes from observed, since this is just Bayes' rule:  $P(\mathbf{x}_M | \mathbf{x}_O) = P(\mathbf{x})/P(\mathbf{x}_O)$ . We use the EM algorithm to fit MoPODD models to data, and minimum description length (MDL) to choose the number of mixture components.

## 2.5 Anomaly Detection for movers

In some data types there are moving entities for which we have temporally dense records, i.e., tracks or essentially complete records of position as a function of time. We studied the detection of anomalous behavior by such movers, modeling normal behavior by constructing probability density estimates for features extracted from *pairs* of points on a track, and then detecting anomalies as point pairs with low likelihood according to the model. The negative logarithm of the density serves as a measure of anomalousness. Tracks that are unusual will have few, if any, similar tracks in the data used to fit the model, so their density should be low (high negative log likelihood).

The challenge with such an approach is coming up with a tractable model for track probability. Variations in track shape can take many forms, but also tracks vary in length, so there is not even a clear way to define an appropriate dimensionality. We have simplified the problem by extracting many feature vectors of fixed size from the track, and building a model of the density of these feature vectors, rather than a model of the density of the tracks themselves. We experimented with a few simple feature vectors, the best of which was the coordinates and velocities at two times on a track, plus the logarithm of time interval between them (Figure 8):

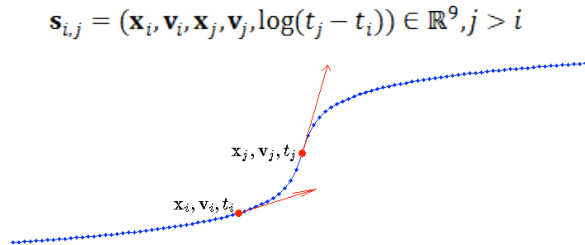


Figure 8. Sketch of features used for anomaly density estimation. The features are pairs of points and velocities at two times on a track, along with the log of the time difference between them.

This choice combines local properties of the track with long-time behavior. In fact a set of such vectors for different point pairs describes properties of the track on a variety of time scales. The log time is used to represent the *relative* time

difference, which seems more likely to be relevant than the absolute difference. For example, the difference between two seconds and four is likely to be significant, but the difference between 100 and 102 seconds is not.

From many such point pairs on many tracks we built a model of  $\Pr(s)$ . This gives many values for a single track from the many point pairs on the track, so we must choose a way to combine these to give a single score. We have used the minimum of the density (maximum of the negative log-likelihood) over the point pairs, since it has a simple interpretation as the most anomalous point pair on the track. This will label as anomalous a long track with a short anomalous section.

For the model of  $\Pr(s)$  we used a kernel density estimate (KDE). These are quite fast to build and to apply to new values of  $s$ . A disadvantage is that they take a great deal of memory because the number of point pairs is quadratic in the number of points in the track. For example, a track lasting four minutes with a sample spacing of 0.66 seconds has about 364 points, and about 66,000 point pairs. Accordingly we chose to skip very closely-spaced points. The approach is therefore insensitive to anomalies on very short time scales, but these are rare in our data.

This approach can be applied in two ways (at least): First, we can train the KDE on a training set, and evaluate it on a test set. This is analogous to building the model and then using it to search incoming new tracks for anomalies. Since this is an unsupervised method, any ground truth tracks should be placed in the test set to enable performance evaluation. Second, we could have built the KDE from all tracks, and checked their scores afterwards. This is a forensic mode, examining the set of tracks for anomalies after their collection. Note that in this mode there is an upper bound on the anomaly scores, since the tracks are in the training set. Probably a great many tracks would end up having this maximal score. We have not yet investigated how serious this problem is, but it should be straightforward.

## 2.6 Behavior model for vessel Pattern-of-Life

The behavior model for vessel PoL is used to analyze the patterns in the behavior of an individual vessel, in particular, the port calls it makes. The goals of this model include

- To predict which ports a vessel will most likely visit next;
- To derive the function/intent of a vessel based on the port calls it has made;
- And to detect unusual port calls for a vessel with known function/intent.

From our AIS data collection, we observe that the behavior of a vessel is pretty predictable once both the starting point and the destination are known. Civil vessels typically travel on fixed sea routes due to physical and economic considerations. On the other hand, predicting the next destination of a vessel is not as simple. That is why the focus of our study is not on the path finding behavior, but on the destination selection behavior of an individual vessel.

Therefore, in our analysis, the AIS track of a vessel is abstracted into a sequence of port calls. What happens between port calls does not matter. As a result, vessel PoL is equivalent to a pattern of semantic symbols. Our fundamental assumption here is that, patterns observed in the past are likely to repeat in the future.

There are a number of challenges associated with such a study.

- A large number of samples are needed to build a relatively reliable behavior model. For many vessels, there are just not enough samples collected.
- Rare scenarios are not likely to be captured in our data collection. Hence prediction made on the behavior in these scenarios is almost always wrong.
- Sometimes, an external factor which is unknown to us caused the vessel to change its behaviors during the data collection period. What we assume to be normal behaviors for that vessel are actually abnormal.

The first approach we experimented with is pattern recognition, which focuses on the recognition of patterns and/or regularities in any type of data [4]. For example, once a pattern such as  $A \rightarrow B \rightarrow C$  was observed, we would start to keep track of which symbols have appeared after this pattern. Then when this pattern is observed again, we can use statistics from the past to make a prediction on the next symbol which has not been observed yet.

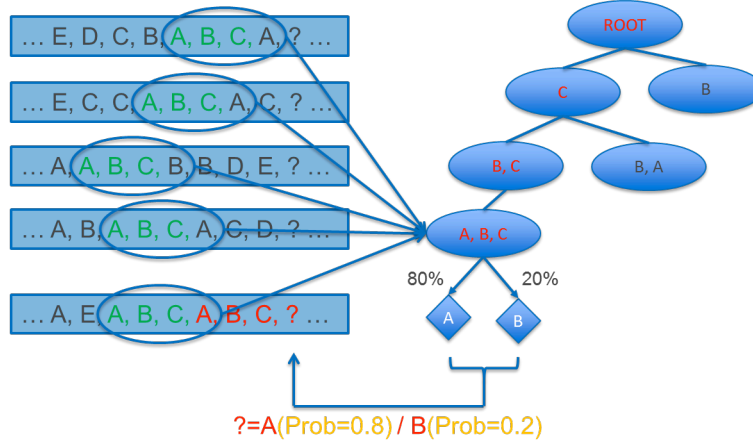


Figure 9 Predict the next port-of-call using pattern recognition. All patterns that have been observed in the past and their transition statistics are stored in a tree-type data structure for easy search and update.

The second approach we tried in our experiments is artificial neural network [5], which comes from the field of machine learning. We chose to use the simplest kind of neural network, feedforward with a single layer of hidden nodes. Here the number of nodes in the input and output layer must be equal, which is the size of the symbol alphabet. On the other hand, the number of nodes in the hidden layer is independent and controls the complexity of the model.

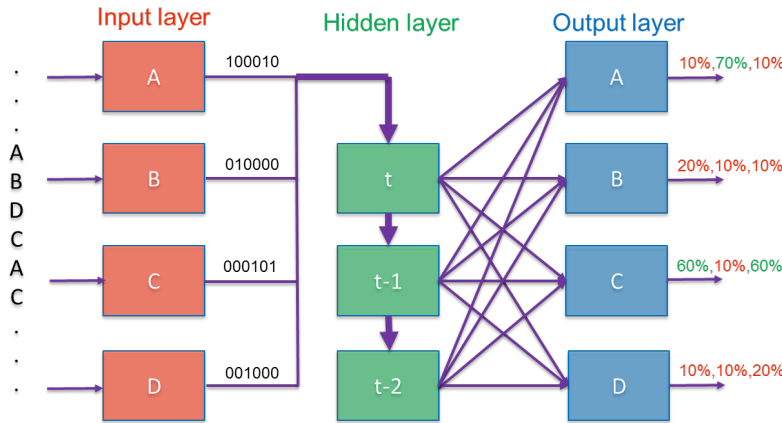


Figure 10. Predict the next port-of-call using a neural network. The neural network used here is a little different from the typical feedforward network since all nodes in the hidden layer have the same inputs but with different delays.

In both approaches, what kind of behaviors the model can recognize or learn is determined by how much information it uses to make the prediction, or more precisely, how much information from the past the model can store in memory before it starts to forget. General speaking, longer memory enables better performance at the cost of increasing time and space complexity. A comparison is only fair when both models have the same amount of memory.

When the behaviors of a vessel is to some extent random, a perfect model that provides correct prediction 100% of the time does not exist. Therefore, we asked both models to generate a short list of the top 5 most likely destinations for a given vessel which is still anchored in port. On average, a better model shall have the true destination on the list more often and ranked near the top of the list more often. Thus, the primary performance metric in our experiment is the prediction accuracy, defined as the probability that the true destination makes to the short list. And the secondary performance metric is the average rank of the true destination when it is indeed on the short list.



## 2.7 Statistical model for vessel Pattern-of-Life

The statistical model for vessel PoL is used to analyze the patterns in the activities of a group of vessels with same/similar function/intent, in particular, frequency of visits and duration of stay at each port. The goals of this model include

- To describe the activity of a vessel with a set of intelligent and human-readable attributes;
- To determine whether two vessels or one vessel in two different time periods have similar PoL;
- And to group vessels with same/similar function/intent together using the statistical model.

The dataset used in our experiments contains in excess of 20 million AIS reports from more than 17,000 unique vessels. We selected roughly 2,600 key ports from across the world based upon their location and exhibited level of activity. Since the exact boundary of each port is not available, we use a bounding box of 0.5 degree in both latitude and longitude to represent each port and its surrounding sea area.

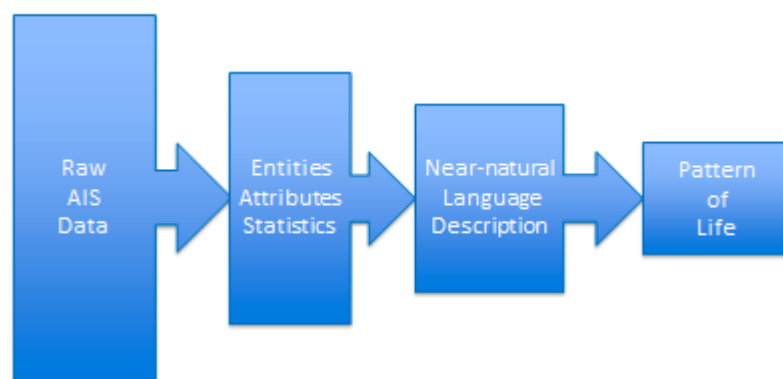


Figure 11. Data flow from AIS data to statistical PoL model. The further down the pipeline, the smaller is the data volume. Less important information is removed at each stage.

In order to boil down large volumes of individual sighting report data to a set of PoL-type activities undertaken by a vessel (e.g. ports visited, amount of time spent at sea, etc.) together with a log of key events in the vessel's history (e.g. arrivals and departures), we have to derive summary information from AIS data pertaining to individual vessels. These abstraction steps not only significantly compress the volume of data, but also provide semantic description to guide user queries. In particular, anomalous voyages by a vessel are readily captured by querying for vessels that have very rare visits to any port.

The vessel behavior description we developed can be used to characterize the activities of a vessel in near-natural language, using all the statistics on its attributes. The baseball-card-like description can answer questions such as

1. Is the activity level of this vessel very low, low, medium, high or very high?
2. Is the footprint of this vessel on the map very small, small, midsize, large or very large?
3. Is the port this vessel going to visit very unpredictable, unpredictable, somewhat predictable, predictable, or very predictable?
4. Has this vessel visited very few, few, some, many, or very many ports?
5. Has this vessel stayed at a particular port very uncommonly, uncommonly, somewhat regularly, regularly, or very regularly?
6. Has this vessel visited a particular port very rarely, rarely, sometimes, often or very often?

Such summaries can be used to rapidly filter potential matches to other data sources that may provide additional information regarding the identity of the vessel at hand. In particular, events such as changes in vessel behavior patterns,

or unusual activities (e.g. visiting a port outside of a vessel’s typical footprint) can be useful in reducing potential matches with other data sources.

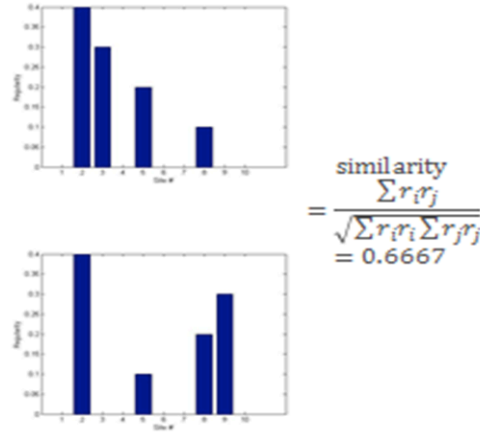


Figure 12. Similarity measurement between the activity patterns of two different vessels. Here regularity histogram is selected to compute the metric.

A vessel can change name/MMSI from time to time but keeps its usual behavior. On the other hand, a vessel with the same name/MMSI can change its behavior due a change of owner or external circumstance. The following vessel matching algorithm provides the capability to match the habit shown by two different vessels or that by the same vessel from different time periods. First, two vectors are forms based on the regularity or frequency of port visited by the vessel inside the selected time window. Then the normalized cross-correlation between these two vectors is calculated. A close-to-1 result indicates highly similar behavior. On the opposite side, a close-to-0 result indicates completely different behavior [6].

### 3. SYSTEM

SRI has designed a scalable processing system that implements our various entity analysis algorithms, and a web-based query interface for end-users to access results. For scalability, the processing system is built upon the Hadoop Map/Reduce framework, storing data in the Accumulo key/value datastore. We have made use of the Geomesa library for geospatial indexing and query support. In the following ‘movers’ are any entity that is mobile, and has at least some persistent unique ID, a location for each sighting, and a timestamp for each sighting. ‘Places’ are static locations, whose description must include a persistent unique ID, and some geometric description of global location.

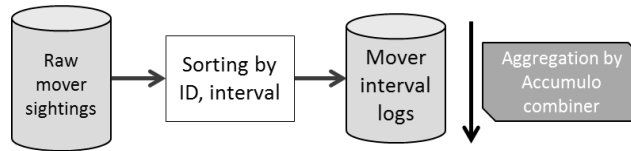


Figure 13: Initial sorting for analysis

Figure 13 illustrates the initial sorting process that is applied to newly-ingested records. Accumulo tables are shown as light gray cylindrical symbols, Map-Reduce processing patterns as white rectangles. The darker gray shape shows that a server-side Accumulo combiner is applied to the right-hand table in the figure. The first stage of sorting for analysis has the goal of reducing the number of individual records that we need to retrieve from the DB for later analysis. A large number of raw mover sightings, indexed by location, ID and timestamp, are sorted by a Map-Reduce job into quantized time intervals (day, or hour, depending upon the sampling rate of the sighting data). The sightings are written to a new Accumulo table, indexed by ID and quantized interval; an Accumulo Combiner iterator automatically aggregates these records such that all sightings for a given ID in a given interval are combined into a single row for efficient query response.

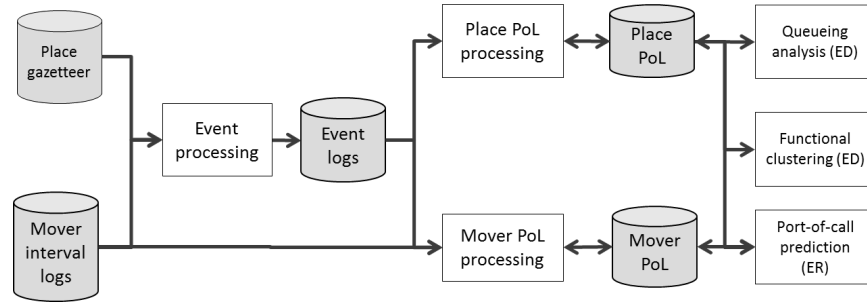


Figure 14: The PoL analysis architecture

Figure 14 shows the outline of the overall processing architecture – having built the mover-interval log, the next step is for a MapReduce job to generate events where movers interact with places; events are indexed by time, location, and the mover/place pair involved in the interaction. This event log is used (together with the mover interval data) as input to our event-based PoL analysis algorithms for places and movers, the results of which are written to results tables that are indexed by interval, entity ID and location. Higher-order algorithms such as queueing analysis, functional clustering and port-of-call prediction then add to the PoL databases.

SRI’s query interface allows users to interact with our result tables from within a web-browser. Backed by an ontology that imposes a logical structure on our database, the interface guides the user through a set of dynamically generated drop-down menus that are used to build up a query which is then sent to an instance of Geoserver that is running on top of the Geomesa-indexed tables in the Accumulo database. For example, starting from selecting a type of place, the user will be given options with regard to what property of that place type they wish to query, and then to filter the chosen attributes based upon semantically meaningful labels. Results are displayed through a number of means; a map overlay (which can also be used to restrict the geographic bounds of the query), a tabular view that allows scrolling through the returned query results, graphical views that plot the variation of the queried attributes over time, and finally a ‘baseball card’ that summarizes relevant attribute statistics and can be summoned by clicking the view of the given entity on the map.

## 4. APPLICATIONS

We primarily used two kinds of data in this work: Wide-Area Aerial Surveillance (WAAS), which is large-format video, and AIS (Automatic Identification System) data, which indicates vessel status. We used other types of data to a limited extent, and we will describe them as needed.

### 4.1 AIS Data

AIS (Automatic Identification System) provides a means for identifying and locating vessels. There are requirements that certain kinds of vessels operate AIS transceivers. These regularly broadcast the vessel’s identity, location, time, and a variety of additional information. We have obtained two sets of AIS data, one for November 2011 to February 2013, and one for January to March 2014. Both are global, though in these data sets there are regions of the world with little or no coverage, and time intervals with very sparse coverage. In addition many of the fields are missing from many reports. For example, heading and speed are missing from roughly half of the cases, navigational status is rarely present, and so on. Consequently we are currently only using identity, position and time, and occasionally speed, when it is present.

From these reports we extract vessel logs describing the ports visited by each vessel, including the time of the visit, and port logs describing arrivals and departures. The latter are extracted by searching the records for reports satisfying several conditions:

- They were sufficiently close to a port’s nominal location,
- The vessel’s previous (for arrivals) or subsequent (for departures) AIS report was not close to that port, and
- If the speed was reported, it had to be below a threshold.

We chose these criteria for several reasons. First, each port’s location is given as a single point (lat-lon pair), which does not describe the port’s boundaries. A distance threshold is a practical alternative. We chose a fairly large value for the distance threshold, 25 km, to match the sizes of some larger ports, and to match our observation of the distances at which vessels sometimes anchor from ports. Because of the large thresholds, vessels passing a port without stopping sometimes pass within the distance threshold, hence the speed criterion. In addition, if the vessel arrives at a port, then reports a speed exceeding the threshold but does not leave the port before reporting that it is near the port with either no reported speed or a speed below the threshold, we treat this as a within-port motion, and do not count it as an arrival or departure.

Given the record of arrivals and departures at a port, we accumulate statistics of this activity. We then convert the statistics to semantic attributes describing the arrival and departure activity to characterize the port’s behavior, i.e., parts of its pattern of life.

#### 4.2 Anomaly detection for movers

We have a dataset of tracks for a large number of vehicles moving on ordinary roads, often in close proximity to each other. Included in this set were some vehicles executing planned routes or “vignettes,” which we assume are unusual, compared to the typical tracks of the other vehicles. The vehicles executing vignettes were instrumented with GPS recorders, so that we have ground truth for them.

Table 1. Summary of anomaly detector performance on scripted vignettes.

<b>Vignette</b>	<b>Number of occurrences</b>	<b>Number in 5% most anomalous tracks</b>	<b>Percentage in 5% most anomalous tracks</b>
Checkpoint avoidance	13	8	62%
U-turns	15	10	67%
Safe house	8	8	100%

There were three vignettes we investigated, each of which was repeated several times (Figure 15).

- Checkpoint avoidance. Here the vehicle travels along a road, but makes a detour around one short stretch by turning left at the preceding intersection, turning right into a parking lot, parking for a time, then exiting the parking lot and continuing along the original road.
- Multiple U-turn. The vehicle begins parked on one street, begins moving and turns right onto a street, executes a U-turn at the next intersection, passes the starting street, executes another U-turn at a subsequent intersection, turns right onto the starting street, executes another U-turn and parks in the starting location.
- Safe house. The vehicle begins at a specific location, then takes a specified winding route along several streets before reaching the destination.

The anomaly detection performance for our dataset is shown in Table 1. Of the vignette tracks that were not in the most anomalous 5% of the tracks, all were missed because the tracks were broken, i.e., our tracking software did not successfully track the vehicle from the start of the vignette to the finish.

We examined non-vignette tracks that had high anomaly scores, and found that the KDE approach has difficulty with point pairs at long time intervals, since the set of possibilities for the second point given the first is very large. There are extremely few point pairs with the very longest durations, five minutes or more. A symptom of this is tracks that may appear quite normal to us that are given a high anomaly score. An example is a track that travels on one street for a long time in a normal way (Figure 16). This is a secondary street, so there are relatively few tracks on it, and most vehicles turn or stop at some point along the street, unlike this vehicle. A second point is that people actually drive along strange routes. Some anomalous non-vignette tracks involved winding through residential areas without stopping except at intersections, something like our safehouse vignette. Postal delivery routes are unlike other tracks, yet are almost always executed once per day. Further progress will probably require a more sophisticated notion of anomaly, and while the KDE algorithm is fine for moderate intervals between point pairs, something else will be useful for longer tracks.

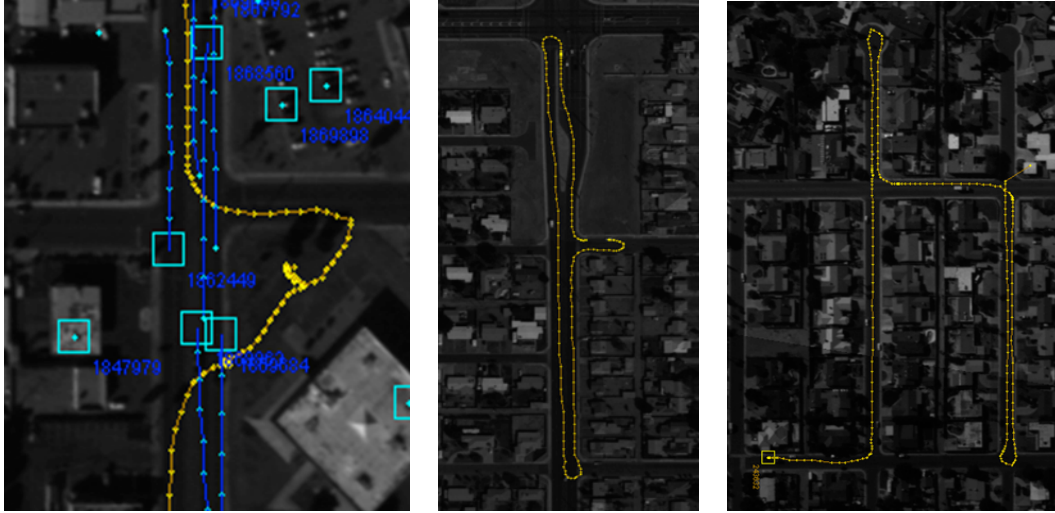


Figure 15. Tracks for vehicles executing scripted anomalous vignettes. From left to right: Checkpoint avoidance, multiple U-turn, and safehouse.



Figure 16. A straight but long track had high anomaly score because few tracks on this secondary street traveled that far before stopping or turning.

#### 4.3 Functional Classification Application to WAAS Data

The queueing models were applied to Wide Area Aerial Surveillance (WAAS) and Automatic Identification System (AIS) data. First we discuss applications to WAAS data for classifying buildings based on their patrons.

We show in Figure 17, vehicle tracks obtained from processing the data. In the figure; we show stabilized and geo-registered aerial images collected from the sensor. Parts of two parking lots are marked using red polygons. The blue dots overlaid on the two figures show sightings for each vehicle track which either starts or ends at the parking lot.

In Figure 18, we show the corresponding vehicle arrival statistics and corresponding distributions which explain the sample statistics. In left panel on the bottom row, we show vehicle arrival statistics for the left parking lot shown in Figure 17. On the right panel, we show samples for inter-arrival time along units of time while on the right we show arrivals per minute. We fit an exponential and a Poisson distribution respectively to the data using a least square algorithm. The estimated distributions are shown in the top row of Figure 18 respectively.

We empirically observed that arrival and departure rates and the service times for all the sites under consideration varied considerably with time of the day. This would also stand to reason since when people arrive, how much they stay and when they depart is strongly correlated to the function the site provides to its patrons, the sites' closing and opening hours, the needs of the patron and how those needs are balanced with the rest of the patrons' schedule. All of these strongly vary in time and motivate the need for a time-varying non-homogenous SPAM model shown in in Figure 2.



Figure 17 Two parking lots are shown above using red polygons. Vehicle tracks are shown as sightings in blue.

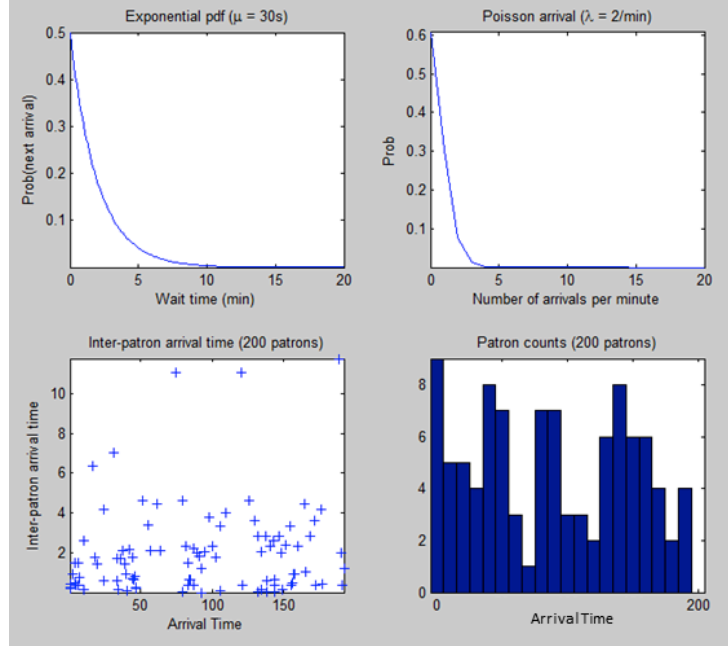


Figure 18 Bottom row shows vehicle arrival statistics for the left parking lot shown in Figure 17 – inter-arrival time is shown on left and arrival rates are shown on the right. Top row shows the corresponding exponential and Poisson distributions.

### Site Activity Signature Statistics (SASS)

The goal of this work is to achieve automatic functional classification of the sites under consideration. We considered a variety of statistics for this purpose. These are explained below.

The SPAM model provides statistics like arrival and departure rates as well as average length of stay (ALS). However, we noticed that the intra-site variability in these rates was much larger than the inter-site variability – thus the discrimination directly offered by the variables  $\lambda(t)$  (average inter-arrival time) and  $\mu(t)$  (average service time) was very low. The patronage activity related to the site shows ebb and flow through the day. In terms of Figure 2, this statistic would be provided by the latent variables  $\alpha_t$  and  $\beta_t$  respectively. The latent variables  $\alpha_t$  and  $\beta_t$  model the changes in variables  $\lambda_t$  and  $\mu_t$  respectively and we represent them as Gaussian random variables  $N(\alpha(t), \sigma_\alpha)$  and  $N(\beta(t), \sigma_\mu)$  where  $\alpha(t)$  and  $\beta(t)$  represent the estimated instantaneous rates.

In Figure 19, we depict arrival rates for two exemplar businesses. For each business, we show the time in minutes on the x-axis and the y-axis represents the inter-arrival time in minutes. The blue plus (+) signs denote the observations from the data and the black stars (\*) are the maximum likelihood fits to this local data giving us an estimate of  $\lambda(t)$ . The green line represents the regression coefficient of the linear temporal fit to the local inter-arrival rates giving us an estimate of  $\alpha(t)$ . It can be seen that  $\lambda(t)$  quite effectively brings to fore the differences between a sample of businesses and is intuitively well explained by the functionality of these businesses. If  $\lambda(t)$  is flat, then the customers arrive at constant

rate. If  $\lambda(t)$  line goes up then the inter-arrival time increases and hence there is a reduction in the rate of arrival of people (traffic to the site slows down). On the other hand, if  $\lambda(t)$  line goes down then the inter-arrival time decrease and hence there is an increase in the rate of arrival of people (traffic to the site increases).

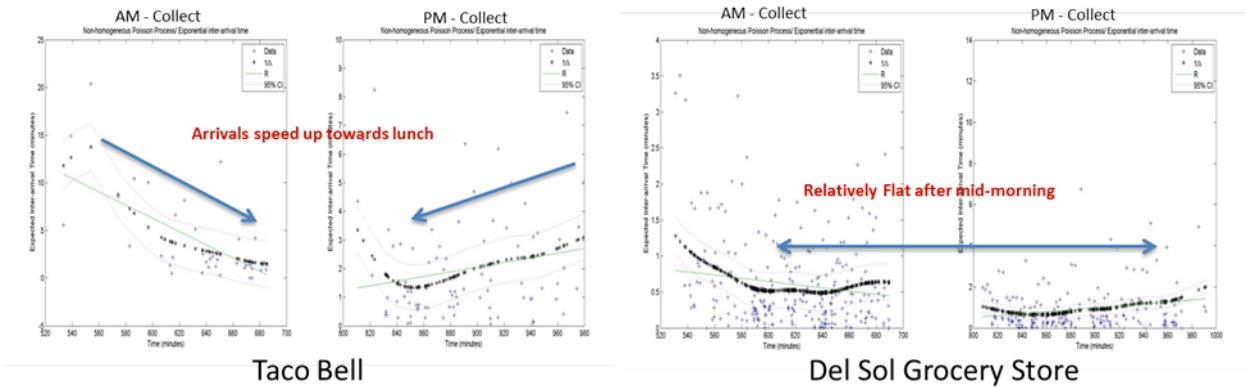


Figure 19 Inter-arrival times shown for 8 businesses. Note how rates change through the day (see text).

### Performance Evaluation: Functional Classification

We examined SPAM features for two collects. The AM collect corresponds to data collected between 9AM and 12 Noon while the PM collect corresponds to data collected between 1PM and 4PM. Fast food businesses like Taco Bell understandably see increased traffic towards lunch. The Del Sol Grocery store seems to have a steady increase in the AM period and enjoy steady patronage through the day. Thus,  $\lambda(t)$  statistic seems well correlated with the functional aspect of these business sites and supports intuitively understood patronage behavior for these sites. Further, it provides a good differentiator between them.

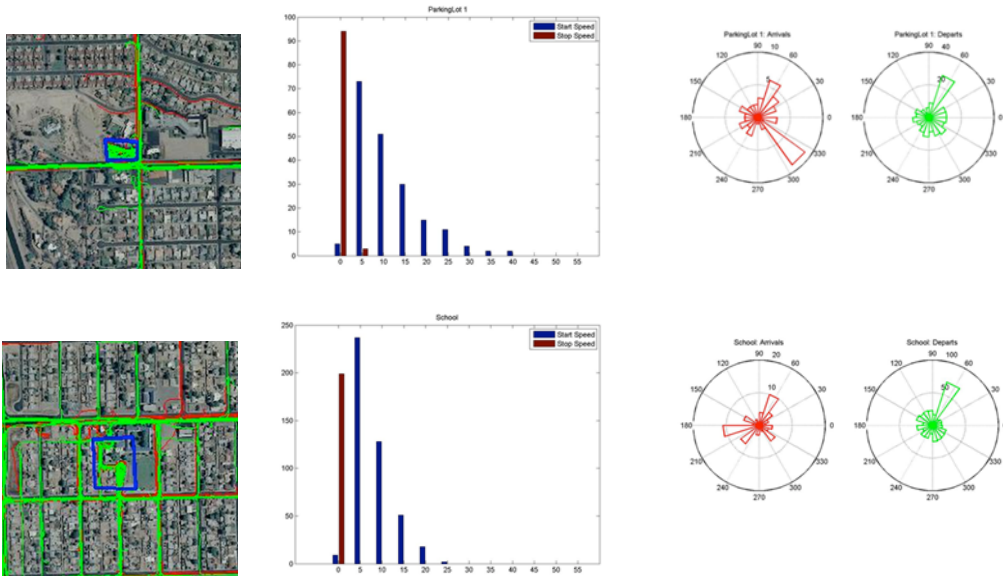


Figure 20: Two ROIs; corresponding speed, orientation histograms for arrivals & departures (see text).

Figure 20 we show two ROIs in the left panel overlaid on the image in blue. Overlaid are also vehicle tracks for vehicles entering the ROIs (in red) and for vehicles leaving the ROIs (in green). In the middle panel, we show corresponding speed histograms and in the right panel corresponding orientations (directions of arrival and departure) as rose diagrams. ROI in the top row corresponds to a business while the one in the bottom row corresponds to a school. It stands to reason that in the school zone cars would move slower than near the business location. Further, the distribution of the peaks of the rose diagram indicate the direction of the roads through the corresponding entry and exit points while their number



indicates how many entry and exist points are there. These features are very useful as they correlate with both the structure and function of the site. The number of patrons visiting the site also reveals a lot about the site. For example, the daily number of patrons of a beauty salon will be order of magnitudes smaller than the daily number of patrons of a school.

Thus, for functional classification, we consider the following SASS vector: ( $N$ ,  $\rho(t)$ ,  $\lambda_{\text{Morning}}$ ,  $\lambda_{\text{Pre}}$ ,  $\lambda_{\text{Lunch}}$ ,  $\lambda_{\text{Post}}$ ,  $\lambda_{\text{Evening}}$ , ALS-AM, ALS-PM) where  $N$  is the total number of patrons visiting the site.

**Business Classification:** We use the SASS vector define above to carry out functional classification of various sites. We did statistical analysis for 17 businesses based on the SASS features. We looked at the following features: the number of patrons, the parameter  $\rho(t)$ , the expected strength of patronage during morning, pre-lunch, during lunch, post-lunch and evening periods based on  $\rho(t)$ , and the Average Length of Stay (ALS) for AM and PM collects. The expected patronage strength is qualitatively calculated to be flat (non-changing), negative (increasing in time), positive (decreasing in time), strong (high number of patrons) and weak linkage (no statistical dependency on time). The results are shown in Table 2. The reason for not doing an automatic classification is that we do not have enough data (number of businesses of each kind) in the number of features and we expect to get deceptively good results due to over-fitting. Hence we just present a qualitative analysis.

Table 2 Statistical analysis of 16 businesses.

Business Name	Number of Patrons	Change in Inter-patron Arrival Time ( $R(\Delta t, t)$ )		Morning Patronage	Pre-lunch Flat	Lunchtime Patronage	Lunchtime Post-lunch Flat	Evening Patronage	Weak Linkage	ALS-AM (mins)	ALS-PM (mins)
		AM	PM	Pre-Lunch Positive	Pre-lunch Flat	Strong	Post-Lunch Flat	Post-Lunch Negative			
Jack in the box	238	-0.32	0.25							17.04	31.6
Ventures Beauty Salon	36	-0.7	-0.25							36.37	11.14
Mexican Restaurant	65	-0.13	-0.32							22.2	17.33
Wild Stix Asian Restaurant	42	-0.35	0.22							21.53	28.58
Taco Bell	128	-0.67	0.15							10.21	12.93
Del Sol Grocery Store	600	-0.19	0.08							23	20.98
Sunglass Store	112	0	-0.17							23.84	16.19
Nails	34	0.53	0.23							45.25	25.82
Subway Fastfood	101	-0.44	0.02							14.64	19.63
ACE Check to Cash	81	-0.1	-0.08							19.69	37.04
MIDAS Auto Services	61	-0.41	-0.11							22.54	26.38
Beeler Equipment Company	48	-0.33	0.04							13.11	21.09
Debell Music (Instruments) Store	24	-0.25	-0.54							17.91	14.98
Breakfast Burrito	46	-0.08	0.23							18.28	38.15
Pets Plus	22	-0.52	0.09							26.25	24.73
Big lots	350	-0.26	0.17							14.78	24.81
Key	Heat Map	Patronage - Activity Spike		Restaurants		Grocery stores		Heat Map		Heat Map	
		Morning + Lunchtime									
		Lunchtime + Evening									
		Evening									

- **Number of Patrons:** It can be seen that the large grocery store (Del Sol Grocery Store) stands out as being highly visited (600). Next, we find the Big Lots department store (350). The popular eatery Jack in the Box (238) comes next while everybody else is below 128.
- **Average Length of Stay (ALS):** The beauty salon Nails stands out as the place where patrons spend the most time in the mornings ( $> 45$  minutes per patron) however they seem to not indulge that much in the PM collects ( $< 26$  minutes per patron). Similarly the Ventures beauty salon: (36 mins to 11 per patron). A reverse trend is seen for Jack in the Box, Breakfast Burrito and ACE check to cash which enjoy much more patronage after noon than before. The distribution for rest of the businesses seems more equitable.
- **Trend of Patronage:** Most businesses show a stronger patronage during lunch than before or after. This stands to reason as the data was collected during week days when people have offices. All lunch businesses – Jack in the Box, Wild Stix Asian Restaurant, Taco Bell, Subway and Breakfast Burrito – show a strong peaking behavior around lunch time. Additionally, Subway continues to enjoy strong patronage in the afternoon while Breakfast Burrito seems to have a similar behavior though in the morning. The beauty salon Nails additionally enjoys a strong patronage in the morning.



From the above analysis, it is safe to conclude that the used features show strong correlation to the functions that we expect the businesses to perform and are also discriminative between different kinds of businesses. It stands to reason that with more data (more collects and businesses of each kind), we can use machine learning to develop automatic classifiers for various business classes.

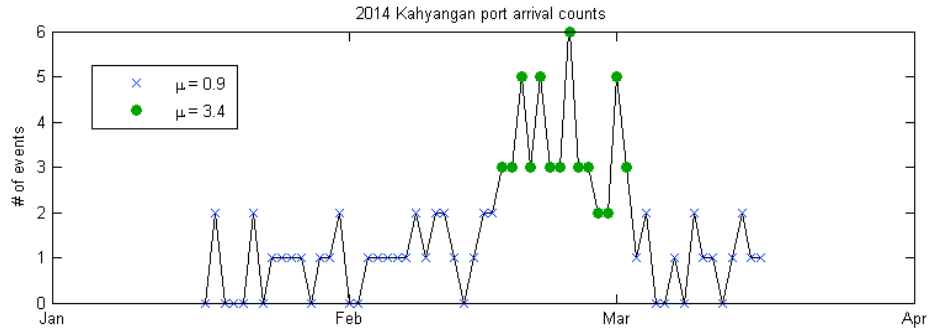


Figure 21. Daily arrival counts for the port of Kahyangan, a case whose appropriate semantic trend descriptor is not obvious. The blue x's and green circles indicate the behavior classes assigned by the HMM-BPN model fit to this data. The legend gives the model means for the two states. For both states the model variances are roughly half the model means.

#### 4.4 Semantic port attributes from HMM-BPNs

We described port arrival and departure activity with several attributes. The HMM-BPN analysis lets us extract attribute scores, i.e., numerical values, using the HMM-BPN model's means, variances, and transition probabilities. We then threshold the scores to get semantic levels. These are typically from the set very low, low, medium, high and very high. We chose the thresholds based on both theoretical considerations and the actual distribution of scores in our data. The attributes include

- **Activity level.** This is derived from the mean number of events. It is easy enough to compute directly, but with records over long periods of time the model can be used as sufficient statistics, and the mean number of events can be computed from the model's equilibrium probability of each state and the state's means. The mean number of daily arrivals and departures varies by orders of magnitude over ports, so we chose thresholds at powers of ten. The semantic levels are then essentially a logarithmic scale.
- **Variability.** We define this as the standard deviation of the daily event counts divided by the square root of the mean. A common approach would be to normalize the standard deviation by the mean, but this results in all low activity ports having a higher variability score than the high activity ports. If we think instead of the typical distribution of counts as a Poisson distribution, the standard deviation is expected to be the *square root* of the mean. Normalizing by the square root of the mean does in fact give scores that cluster around one, though there is a clear tendency for higher activity ports to have higher score. It shows that the assumptions behind the Poisson model are violated: Either the events are not independent or the arrival rate is not constant. (For ports with such low rates that most days have zero or one events, we can show that this score cannot deviate much from one.) Variability can be computed like activity level, either directly from the record of daily event counts, or using the HMM-BPN model as a compact summary of this record.
- **Predictability.** We define this by using the HMM-BPN to form a prediction of a day's event count given the counts on previous days. The HMM-BPN also gives a variance for that prediction. We then compute the predictability as the ratio of the prediction's standard deviation over its mean, averaged over the time steps. A pure Poisson model would have a value of one, while values greater than one are less predictable, and values less than one are more predictable. These are mapped to the semantic predictability levels very low through very high with appropriately chosen thresholds. The medium predictability level has ratios near one, so that such ports have behavior like that of a Poisson distribution, at least in this respect.
- **Trend.** Intuitively, we might want to be able to query for ports whose activity is trending higher or lower. Linear fits to a sequence might allow this, but the nature of these sequences is too irregular to suit that approach. Consider the arrival count sequence in Figure 21. While there are significantly more arrivals in the

second half of the interval, one could instead say that there was an episode of higher activity, and the activity has returned to normal in the past two weeks. The latter seems a more appropriate kind of description than one in terms of trend. As seen by the symbols for different HMM states in Figure 21, the HMM-BPN’s behavior classifications support this kind of description. Another example can be seen in Figure 4, whose recent activity could be described as: Activity for the past few weeks has been quite high for this port, but it switched to low in the past few days. We are working on code to extract descriptions of this sort that support queries.

#### 4.5 Functional Classification Applications to AIS Data

For AIS data, we applied a similar approach. The AIS dataset we received did not include a completed field for “at-anchor” status so we inferred arrival and departure of ports based on position over time. If a vessel remained within a small epsilon of a port known to NGA’s World Port Index, we considered it “in-service”. Its arrival and departure time was determined by the first and last time it spent in the port bounding box, respectively.

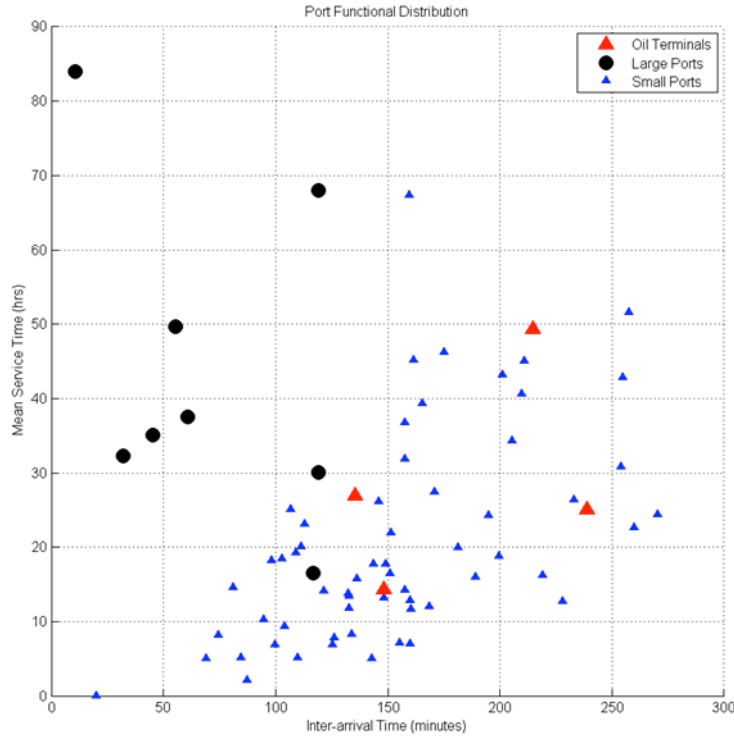


Figure 22 Distribution of Ports by Mean Service Time and Inter-Arrival Time in minutes.

The busiest ports as shown in Figure 22 cluster in the top left with less time between ship arrivals and more time to process each ship. This cluster may signify commercial shipping ports. Denoted by the red triangles, oil terminals have a different functional behavior, with less frequent ship arrivals and slightly less service time. These attributes can augment the features generated by the MoPODD approach to produce a high-dimensional representation of ports.

#### 4.6 Filling in missing values in the NGA World Port Index

We investigated this problem using data in the NGA World Port Index.<sup>2</sup> In their words, the index “contains the location and physical characteristics of, and the facilities and services offered by major ports and terminals world-wide...” Besides name, latitude and longitude, sailing directions publication and chart, the index gives sixty-four attributes for each port. Most of these attributes are binary, such as whether the port has lifts with a capacity of 100 tons or greater. However some of the attributes are categorical, such as port type, and several are ordinal, such as quality of shelter, and various depth attributes. Most ports, if not all, have attributes that are missing. Some, such as “ICE MOOR” for the “LOAD/OFFLOAD” category of attributes, are reported for very few ports, while others are reported for most or all ports. There are more than  $10^{26}$  possible attribute vectors.

We fit a MoPODD model to this data, as described in Section 2.4. For the ordinal attributes such as the depths and shelter quality, we used a distribution with an underlying log-normal distribution of positive real depth values, and equally-spaced thresholds to put the real value into one of the depth categories. The normal distribution’s mean and variance are then fit to the data.

To evaluate the model, we chose half of the ports at random, and for each port left out one attribute chosen at random. After training on this set we compared the inferred values for these left out attributes with their known values. For comparison we initially used simple prevalence, assuming independence of the attributes (hence the label “Independent” in the plots). With this method the probability  $p$  of an attribute having value  $x$  is the fraction of all ports with value  $x$  for that attribute. In some cases the MoPODD model and simple prevalence gave similar probabilities, but in some they were quite different. For example, the index does not indicate whether Portland harbor (Southern England) has degaussing facilities. Simple prevalence gives a probability of slightly less than 50%, whereas the MoPODD model gives it a probability of 99%. (The port historically had degaussing facilities, but we don’t know whether this is still present and operable.)

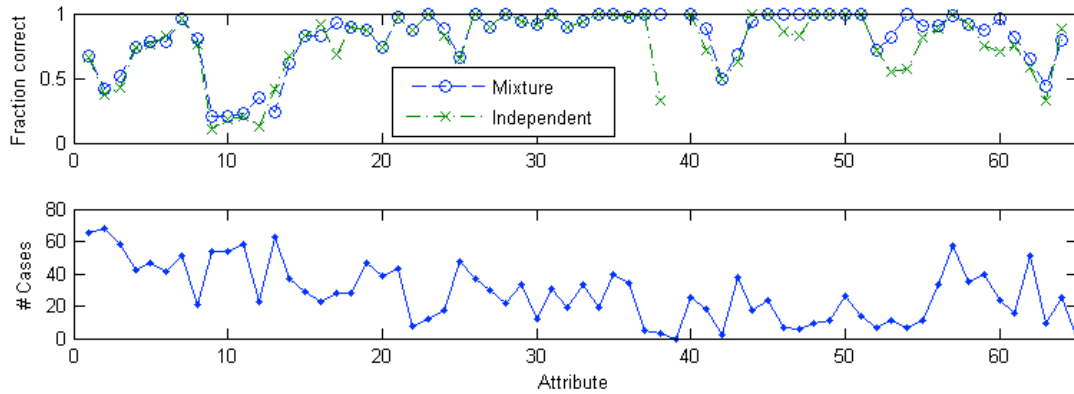


Figure 23. Fraction of correct predictions for each attribute, plotted by the order in which attribute appears in the NGA World Port Index. Also shown is the number of cases for which each attribute was missing. Note that attribute 39 had no cases. Note also the depths, which are attributes 10–14. These have a lower fraction correct because it is more difficult to get exactly the right value.

There are different ways to measure performance. Inference of an attribute gives us the probabilities for each of the attribute’s possible values. If, for a left-out attribute, we choose for the prediction the value that the model says is most likely, we can easily compare with the true value. This was correct in 34.2% of the cases. Simple prevalence was correct in 31.8% of the cases. By attribute, the number of correct predictions is shown in Figure 23.

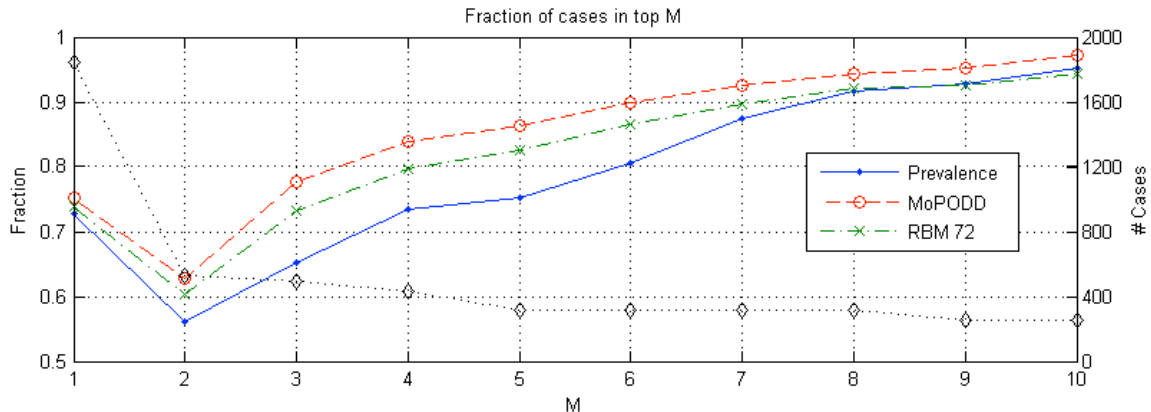


Figure 24. Fraction of cases with the correct attribute value in the  $M$  most likely values, for a MoPODD model, a restricted Boltzmann machine (RBM) and predicting for each attribute as if it were independent of the others (Prevalence). Also shown as the black dotted line with diamonds at the data points, is the number of cases for each  $M$ , which is the number of cases for which the missing attribute had more than  $M$  possible values.

While the MoPODD model is superior, the rate of correct prediction given above may not seem very high. Part of the reason for the low number may be the categorical and ordinal attributes, since there are more opportunities to be wrong for these. This is supported by the results in Figure 23 for the depth attributes. In these cases the correct prediction may not be the most likely, but it may still have high probability. To address this, we looked at the fraction of cases in which the correct value was in the top M values. Note that for a given M this applies only to attributes with more than M values, and therefore the number of cases being tested varies with M. A plot of these results is shown in Figure 24. Note that binary attributes only contribute to the values on the far left of the plot (M=1).

We would like to have compared these results with an alternative besides simple prevalence, say a clustering algorithm, but we are not aware of a clustering algorithm that handles both inhomogeneous attribute types and missing values. However, Restricted Boltzmann Machines (RBMs) <sup>[1],[2]</sup> are currently popular, and can deal with these features. RBMs are a type of neural network that uses a pairwise energy function to express the probability of visible values (our attributes) and hidden values. The latter are usually binary. The energy has no terms for interactions between visible units, and none for interactions between the hidden units. Training is somewhat expensive, as is inference, but not prohibitively so. For the RBM community, our case is a little unusual because of the visible variables with different types. We chose 72 hidden units, following a recipe for choosing this number given by Hinton.<sup>[2]</sup> We actually tried a range of numbers, but none were consistently better. The results are shown in Figure 24. The RBM did not perform as well as the MoPODD model for this problem.

## 5. CONCLUSIONS

The work we have presented is the beginning of a patterns of life analysis system. There is much opportunity for expansion, both into other data types, and into other approaches to extracting actionable information. Interaction with human domain experts can enhance the system in several ways, such as providing human-readable labels for attributes and the unsupervised clusters of entities. This gives both results in a more human friendly form and enables future queries. The anomaly detection functions can easily be extended to changes in PoL of sites from the state analyses, and also to measurement and data quality anomalies. An intriguing possibility is to infer causes for state changes among external events, such as an embargo or natural disaster. A useful capability is to allow users to ask for alerts for specific conditions when a severe change to PoL is detected. The PoL system can be updated to learn what changes are alert-worthy. This is a field with rich opportunities for further work.

## ACKNOWLEDGEMENTS

This work was sponsored by the Department of the Navy, Office of Naval Research, under contract number N00014-08-C-0339.

## REFERENCES

- [1] Freund, Y., and D. Haussler. "A fast and exact learning rule for a restricted class of Boltzmann machines." *Advances in Neural Information Processing Systems* 4 (1992): 912-919.
- [2] Hinton, Geoffrey, "A practical guide to training restricted Boltzmann machines." *Momentum* 9(1), 926 (2010). Also available at <https://www.cs.toronto.edu/~hinton/absps/guideTR.pdf>.
- [3] "Maritime Safety Information," [http://msi.nga.mil/NGAPortal/MSI.portal?nfpb=true&pageLabel=msi\\_portal\\_page\\_62&pubCode=0015](http://msi.nga.mil/NGAPortal/MSI.portal?nfpb=true&pageLabel=msi_portal_page_62&pubCode=0015), (Accessed 11 December 2014).
- [4] Bishop, Christopher M., "Pattern Recognition and Machine Learning," Springer, p. vii. (2006).
- [5] Rosenblatt, F., "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain," *Psychological Review* 65 (6): 386-408 (1958).
- [6] Tahmasebi, P., Hezarkhani, A., Sahimi, M., "Multiple-point geostatistical modeling based on the cross-correlation functions". *Computational Geosciences* 16 (3): 779-797 (2012).
- [7] Smirnov, N., "Table for estimating the goodness of fit of empirical distributions". *Annals of Mathematical Statistics* 19: 279-281 (1948).