Robust resource demand estimation using hierarchical Bayesian model in a distributed service system

Sumanta Mukherjee
sumanm03@in.ibm.com
IBM Research - India
Bangalore, Karnataka, India

Krishnasuri Narayanam
knaraya3@in.ibm.com
IBM Research - India
Bangalore, Karnataka, India

Nupur Aggarwal
nupaggar@in.ibm.com
IBM Research - India
Bangalore, Karnataka, India

Digbalay Bose[*]
University of Southern California
Los Angeles, CA, USA
dbose@usc.edu

Amith Singhee
asinghee@in.ibm.com
IBM Research - India
Bangalore, Karnataka, India

ABSTRACT

Robust resource demand prediction is crucial for efficient allocation of resources to service requests in a distributed service delivery system. There are two problems in resource demand prediction: firstly to estimate the volume of service requests that come in at different time points and at different geo-locations, secondly to estimate the resource demand given the estimated volume of service requests. While a lot of literature exists to address the first problem, in this work, we have proposed a data-driven statistical method for robust resource demand prediction to address the second problem. The method automates the identification of various system operational characteristics and contributing factors that influence the system behavior to generate an adaptive low variance resource demand prediction model. Factors can be either continuous or categorical in nature. The method assumes that each service request resolution involves multiple tasks. Each task is composed of multiple activities. Each task belongs to a task type, based on the type of the resource it requires to resolve that task. Our method supports configurable tasks per service request, and configurable activities per task. The demand prediction model produces an aggregated resource demand required to resolve all the activities under a task by activity sequence modeling; and aggregated resource demand by resource type, required to resolve all the activities under a service request by task sequence modeling.

CCS CONCEPTS

• **Computing Methodology → Modeling and simulation**; *Machine learning*; • **Applied Computing** → *Enterprise Computing*.

KEYWORDS

Distributed service delivery system, factor analysis, hierarchical Bayesian model, robust estimation

ACM Reference Format:
Sumanta Mukherjee, Krishnasuri Narayanam, Nupur Aggarwal, Digbalay Bose, and Amith Singhee. 2021. Robust resource demand estimation using hierarchical Bayesian model in a distributed service system. In *8th ACM IKDD CODS and 26th COMAD (CODS COMAD 2021), January 2–4, 2021, Bangalore, India.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3430984.3431003

1 INTRODUCTION

In a distributed service delivery system, a response to a service request often involves multiple tasks. These tasks are often dependent on each other in sequential order. Dependence order among the tasks is restricted by the service delivery industry domain. Different service requests can be of different types. The type of a service request is defined by a combination of tasks. Further, a task is defined by a combination of activities. Each task is carried out by one or more resources of a specific resource type. All the activities associated with a task are carried out by the same resource type.

A proactive response in such a service delivery system requires estimation of upcoming resource demand at activity resolution across all service requests. Demand estimation is done over a specified time interval. The specific composition of task types defining a service request type is explicitly modeled in our method. And the specific composition of activities defining a task type is also explicitly modeled in our method. Activity is the finest level of granularity of specifying the resource demand. For example, in an electrical utility, the service request is equivalent to a reported power outage. Resources represent the skilled human resources involved in resolving the service requests. Based on the work type, there can be multiple resource types, viz. repair crew, and assessment crew. A service request may involve multiple tasks based on the type of the service request. Each task involves multiple activities, viz., acquire tools, travel to the location, and work at the site.

There are two distinct aspects to this problem, firstly estimating the number of incoming service requests, and secondly, for a given volume of requests estimating the resource demand. The first aspect of the problem can be solved using time-series analysis. But it's not possible to solve the second part using time-series analysis, as the time series models are incapable of producing a sequence of

[*] This author contributed when he is part of IBM

CODS COMAD 2021, January 2–4, 2021, Bangalore, India

ACM ISBN 978-1-4503-8817-7/21/01…$15.00
https://doi.org/10.1145/3430984.3431003

activities with dependency constraints, which is essential for the resource demand estimation. This estimation process involves a high degree of uncertainty, inherent to the system.

Here we will discuss a novel Hierarchical Bayesian Network model for robust estimation of the resource demand profile for a given set of service requests.

2 RELATED WORK

Resource demand estimate for service request resolution under emergency is needed for any service delivery organization for faster resolution of the service requests. The problem of risk management of electrical outages was addressed by [10] with the use of a Poisson regression model for spatial data in a hierarchical Bayesian framework. During a severe weather event, a Tobit model-based system was proposed by [2] for estimating number of outages in a distribution grid. The problem of forecasting weather-driven damages of different types is tackled in [15] by combining a spatial clustering based scheme with data from multiple weather networks. A combination of weather-based simulations, land-cover and outage utility data was used by [7] to calibrate various ensemble-based methods for predicting the spatial distribution of outages in Northeastern USA. Another application of ensemble-based methods was explored by [12], where a boosting-based technique called Adaboost+ was designed for estimating wind and lightning related outages.

In [1], a statistical model based on weather forecasts, asset information, historical damage patterns and geography was built for predicting localized interruptions and was subsequently used by National Grid in its emergency planning efforts. [16] also combined calibrated weather models with historical damage data to design a forecast model for outages. An integrated system called OPRO (Outage Prediction and Response Optimization) [20] was proposed for emergency situations in terms of weather events by integrating weather and damage predictions with resource planning and health aware damage hot-spot analysis. [17] developed a decision support tool based on the model of distribution circuit layout, the placement of protective and switching devices and the location of customers for resource allocation and management. [14] in their work have proposed a simulation based modeling framework to analyze the optimal point of distribution under emergency situation.

A predictive method that utilizes different weather data in a GIS framework was developed for outage maintenance by [3]. The GIS framework in-spite of providing the advantage of handling geo-spatial data efficiently, fails during the time of extreme weather events due to non-availability of location information. [21] explored a fuzzy logic based methodology for crew management in case of large scale multiple outages. A similar approach was considered by [4] where both weather related forecasts and power-system based operational data were integrated with a fuzzy logic approach to aid the outage maintenance system. An unsupervised approach based on ensemble learning method was designed by [19] for predicting the damage of extreme events like wildfires in Australia.

In spite of this large gamut of literature none precisely addressed the problem of robust demand estimation along with a structure of task/activity order dependency. The method can address modeling of any system that conforms to the desired process ontology (as in section 3).

3 DEFINITIONS

In order to produce a generalized model across various systems, we have designed a descriptive process template. Each service request is broken down into a sequence of tasks. Each task resolution involves a sequence of activities. Demand is attributed to each activity. Demand is described in terms of resource hours. E.g., If the activity demand is 8 units, it means that a resource need to spend 8 hours to complete that activity. Equivalently, it takes 4 hours for 2 resources working on the same activity together. The system takes the expected number of service requests (R) over a period of time as input, and produces spatio-temporal resource demand profile over that period (D). The service requests are distributed over geolocation, and arrive at different times. The split of the service requests (R) into their corresponding tasks is represented as (T), and the split of the all the tasks (T) into their corresponding activities is presented as (A).

For each activity, there is a distinct resource demand associated. Demand associated with an activity is affected by few observed variables or attributes, which we call as factors (**F**) in our modeling. The factors can be external (E) and internal (I). Generally, external factors (E) remain constant during the period of service request resolution. Internal factors (I) represent variables that are associated with the service request, task, or activity (Figure 1).

In an electric utility distribution grid, terrain specifics of the outage location, time of the year, time of the day, weather condition are examples of external factors. Equipment associated with an outage, severity of an outage, type of task are examples of internal factors. Internal factors are only known post the service request resolution.

The model assumes a causality relationship between these entities, viz. factors, service requests, tasks, activities, and demands. Imposing this causality restriction helps to disambiguate the dependency order between these entities. Figure 1 depicts this dependency. Assume that I_R, I_T, and I_A are the internal factors associated with the service requests (R), tasks (T), and activities (A) respectively. E, and R are observed entities (i.e., input) at the time of demand estimation (highlighted with double circled nodes in Figure 1). T, and A are generated using sequence models. R, T, and A represent the volume estimate of work at different granularity and are shown as nodes in black in Figure 1. Modeling of T is dependent on E and I_R. Modeling of A is dependent on E and I_T. The demand estimate (D) is obtained using hierarchical Bayes model (marked blue in Figure 1), which intern uses all factors (E, I). Modeling of internal factors (I_R, I_T, and, I_A) is always dependent on external factors (E) (all the factor nodes are marked in red in Figure 1). Details of the internal factor modeling, and demand estimation are explained in the section 4.

4 METHODS

Demand prediction for a specified number of service requests (R) happens via two phases: training, and scoring. In the training phase, statistical model parameters are estimated using the historical data. Here historical data includes details about the service requests, their corresponding tasks, activities, and resource demand along with the associated external factors. In the scoring phase, using the

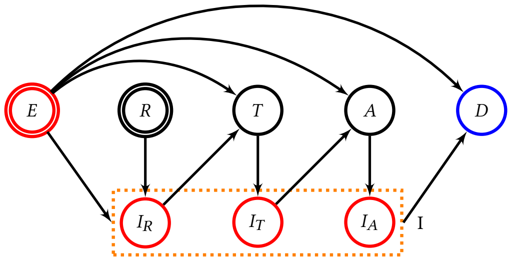


Figure 1: Process template, and causal dependencies.

derived statistical model, demand is estimated for each activity that corresponds to the set of input service requests.

The non-parametric statistical analysis of the resource demand often exhibits multi-modal behavior. If the modes are distinct and well separated, we call them operating characteristics. There may be one or more operating characteristics associated with a input set of service requests.

The training phase includes 4 major sub-phases: (1) Identification of contributing factors to distinct operating characteristics, and partitioning the training data for regression model building, (2) Conditional sequence generator modeling, (3) Generative models for internal factor estimation, and (4) Identification of crucial factors and robust regression model estimation for resource demand.

Scoring step uses the trained model obtained in the training phase, along with the expected volume of the service requests and the observed values of external factors to produce the expected resource demand profile (D). Scoring step performs a series of prediction tasks in the order of entity dependencies in the process template (Figure 2). Score phase involved three distinct computational steps: (1) Estimation of internal factors, (2) Expected task sequence estimation along with activity sequence estimation, and (3) Demand estimation using regression model.

4.1 Input

Inputs to the training are: (1) historical data, (2) input specification. Historical data is a set of N records. Each record composed of values for different data items, e.g., service request, task, activity, task type, task dependency, associated process attributes etc. The input specification explicitly describes the correspondence between each entry of record to the process template. The historical data is transformed to map to a set of tuples $\{t^i_R, t^i_T, t^i_A, E^i_1, \ldots E^i_k, P^i_1, \ldots, P^i_m, R^i, T^i, A^i, D^i\}_{i=1 \ldots N}$, where t^i_R,t^i_T, and t^i_A represent the respective starting time-stamps for service request, tasks, and activities of each historical data entry. E^i represents various external factors, P^i, R^i, T^i, A^i, and D^i, are associated parameters, service request, tasks, activities and resource demand respectively. In a electric utility distribution grid, examples of a parameter are service request type, service request arrival time, task type, equipment used, cause of the outage, etc. These parameters are used to derive the internal factors (I) in our method.

4.2 Identification of distinct operating characteristics

In order to produce a low variance demand estimation model, the demand profile of the training data is analyzed for identification of multi-modality or distinct operating characteristics. The multi-modality is associated with operating characteristics only when it is possible to identify an unique factor, that best explains the multi-modal behavior (Figure 3). Our model imposes a strong association for any operating characteristic with only one factor, in order to identify the appropriate statistical model during demand estimation step. To evaluate the contribution of a factor to the distinct operating characteristics, the data is partitioned into non-intersecting chunks by factor value. If F_i represents a factor which can take possible values $\{f_{i,1}, f_{i,2} \ldots f_{i,k}\}$, then the data $\mathbf{X}$ is partitioned into k non-intersecting sets $X_1, \ldots X_k$, where $X_j = \{x : F_i(X) = f_{i,j}\}$ and $j \in \{1, 2, ..., k\}$.

Kolmogorov-Smirnov (KS)'s two sample test identifies the max separation between two non-parametric cumulative distributions

$$\gamma_{m,n} = \max_x |G_m(x) - G_n(x)|$$

where, G_m and G_n are two cumulative distributions derived from input data partitions m and n respectively. KS distance ($\gamma_{m,n}$) is symmetric and satisfies metric properties [6].

KS distance is evaluated between all partition pairs. δ_D is a decision parameter used to decide whether the KS distance obtained is significant for operating characteristic identification. In order to ascertain the significance of the KS distance we impose a minimum sample size ($\mathbf{N}$) requirement on each partition. Using the distance measure $\Gamma(X|F_i) = \left[\gamma_{X_i,X_j}\right]_{i,j}$, a hierarchical cluster is obtained ($\mathbf{H}(F_i|\Gamma)$). At a distance cutoff of δ_D, the cluster partitions are marked. This partitions the factor F_i values into non-overlapping groups $C^1_{F_i}, \ldots, C^u_{F_i}$, such that KS distance between any two factor values in the same partition is $< \delta_D$, while between any two factor value across partitions is $\geq \delta_D$.

The sup norm of the evaluated distance matrix ($||\Gamma(X|F_i)||_\infty$) is marked as factor association score to operating characteristics. The factor with maximum association score ($\geq \delta_D$), is considered as the explanatory factor.

Explanatory factor identification, splits the training data into non-intersecting partitions. For each data partition, above method of finding operating characteristics is repeated, till no distinct operating characteristics are found.

4.3 Conditional sequence generator

Total resource demand forecast ($\mathbf{D}$) is an aggregated demand across all activities.

$$\mathbf{D} \quad = \quad \sum_r \sum_{t \in \mathcal{T}(r,F)} \sum_{a \in \mathcal{A}(t,F)} D(a,t,r,F)$$

In above expression, r iterates over list of all incoming service requests. $\mathcal{T}(r,F)$, and $\mathcal{A}(t,F)$ are task sequence generator and activity sequence generators respectively, and both these sequence generators are influenced by external and internal factors F (i.e., $E \cup I$).

We have used Markov chain approach for the activity sequence and task sequence modeling [13]. However, the framework is not

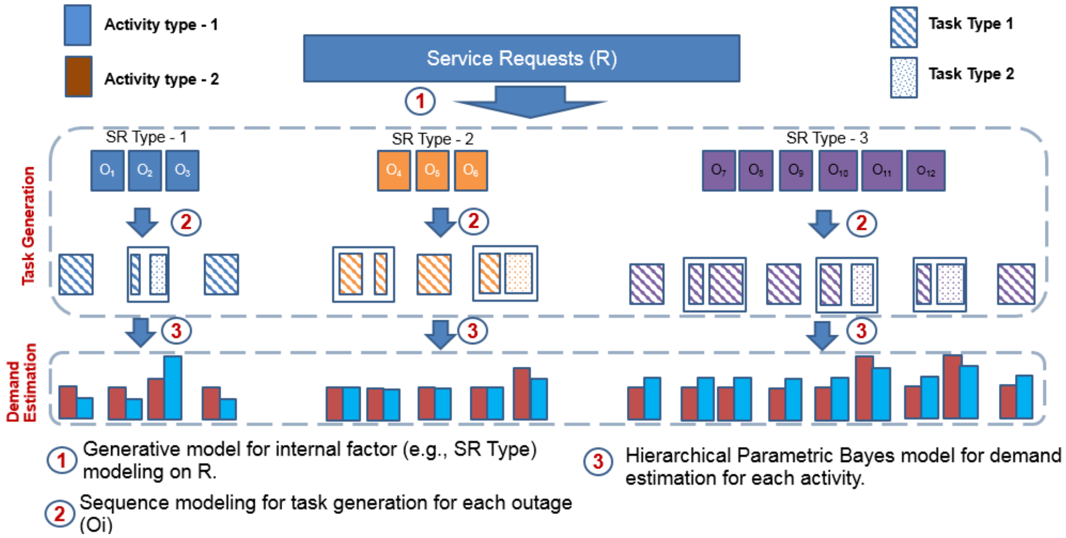

Activity type - 1
Activity type - 2
Service Requests (R)
Task Type 1
Task Type 2
1
SR Type - 1
SR Type - 2
SR Type - 3
O1 O2 O3
O4 O5 O6
O7 O8 O9 O10 O11 O12
Task Generation
2
3
Demand Estimation
1 Generative model for internal factor (e.g., SR Type) modeling on R.
2 Sequence modeling for task generation for each outage (Oi)
3 Hierarchical Parametric Bayes model for demand estimation for each activity.

Figure 2: Steps in the activity based demand estimation method. O_i is used to represent an outage (i.e., a service request). Service request type (i.e., SR Type) is the factor considered here.

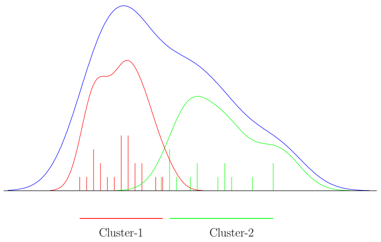


Figure 3: Sample partitioning of data points into two clusters. Factor values are similar, in each cluster.

restricted to markov assumption, and any non-markovian sequence model can also be used for the same. A typical Markov chain is an infinite length sequence generator. In the current context, task and activity sequences are of finite length. To address the same, we introduce two pseudo states, viz., start state (s) and end state (e) (Figure 4). Any sequence generated by this Markov chain model is of type ($s \dots e$). The Markov chains are represented with transition matrix $\mathcal{M}$, which is learned from the historical data using Markov training process. Causal dependence between the factors restricts the factor selection for task sequence generator and activity sequence generator modeling. Influential factor selection for sequence generator is a computationally expensive process. For every factor (F_i), first the data is partitioned by the factor value association. Markov transition matrix is estimated on each sub-partition. $\mathcal{M}(X|\mathbf{F}_{i,j})$, represents the Markov transition matrix derived for j^{th} value of the factor F_i, from a given data X. The distance between given pair of Markov transition matrices ($\mathcal{M}_i$, and $\mathcal{M}_j$) is defined as ‘Frobenius‘ norm of difference matrix [11].

$$d(\mathcal{M}_i, \mathcal{M}_j) = \sqrt{\text{trace}\left((\mathcal{M}_i - \mathcal{M}_j)^T (\mathcal{M}_i - \mathcal{M}_j)\right)}$$

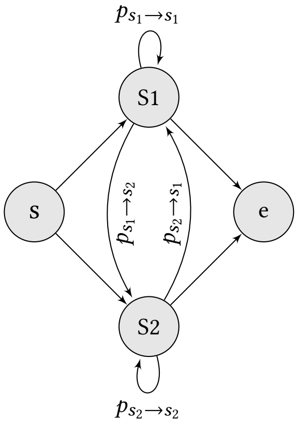


Figure 4: Finite state generator using Markov model.

The influence of a factor F_i on the task sequence or activity sequence ($\mathcal{I}(X|\mathbf{F_i})$) is estimated as the maximum distance from all pairs of transition matrices derived from the partitions for factor (F_i).

$$\mathcal{I}(\mathbf{X}|\mathbf{F_i}) = \max_{j \neq k} d\left(\mathcal{M}(X|\mathbf{F}_{i,j}), \mathcal{M}(X|\mathbf{F}_{i,k})\right)$$

Factors are ranked using influence score. Top ℓ factors are used for final sequence generator modeling. ℓ is a model regularization parameter. Factors are categorical in nature, separate sequence generator models are derived for different combination of influencing factor values.

4.4 Demand estimation model

Deriving the dependency graph is the crucial step in the demand estimation model. Dependency graph incorporates the causal dependency as described in the process template, and also captures statistical association between all entities. Dependency graph is computed multiple times during the training phase. It is used for internal factor modeling, and for resource demand modeling. For internal factor modeling, a single dependency graph is generated for all factors using the complete training data. In resource demand modeling, distinct dependency graphs are derived for each activity demand under each operating characteristic partition.

4.4.1 Dependency Graph. Mutual information (MI) is used to identify statistical association between any pair of entities. The challenge here is the factors are categorical, while the demands are continuous. A nearest-neighbor based method is used for accurate estimation of MI [18]. A normalized mutual information (NMI) measure is used for system-wide analysis [8]. NMI between two random variables X and Y is defined as,

$$NMI(X;Y) = \frac{I(X;Y)}{\min(H(X), H(Y))}$$

where $I(X;Y)$ represents the mutual information between variables X, and Y; $H(X)$, and $H(Y)$ represent the information entropy for variables X , and Y respectively. The NMI is evaluated between every pair of entities E, I and D. In a dependency graph, each vertex represents either a factor or a demand variable. An edge between node pairs describes the statistical association in terms of NMI as edge-weight. A direction is assigned to each edge by causal dependency compliant with process template.

4.4.2 Internal factor modeling. Dependency graph with all factors is the backbone for internal factor modeling. Full training data is used to derive this dependency graph. ℓ is a model regularization parameter; i.e., the maximum number of factors that can be used in modeling of an internal factor is ℓ. Only top ℓ in-coming edges are used in factor modeling. For each internal factor modeling, a joint distribution is generated from the data X. Let's assume an internal factor F_i can take k distinct values, viz. $f_{i,1}, f_{i,2}, \ldots, f_{i,k}$. $F_{i1}, F_{i2}, \ldots F_{i\ell}$, are the modeling factors for F_i derived from dependency graph. The generative statistical model for internal factor F_i captures the conditional probability distribution $\mathrm{P}(F_i|F_{i1}, F_{i2}, \ldots F_{i\ell})$.

4.4.3 Hierarchical Bayesian Estimation. A Hierarchical Bayesian model (HBM) is used for resource demand modeling. A distinct HBM model is derived for each activity, and its operating characteristics. First using the variable dependency graph for a demand associated with an activity (D_a), statistical association of factors to D_a are identified. Top ℓ influencing factors are used in the modeling. Factors are further ranked by their NMI value with D_a in descending order. The training data is then recursively partitioned using the factor values in the rank order in a hierarchical fashion. The root node represents the set of all entries of D_a. In the next level, each partition represents a set of values of D_a associated with a unique factor value. Similarly, the subsequent data partitions are obtained with respective factor values in a ranked order. In the hierarchical data partition, the data size reduces exponentially with increasing depth of the partition tree.

In HBM, parameter estimation is carried out in top-down fashion. First the statistical distribution is obtained for the root partition. For root node, maximum likelihood estimate (*MLE*) is used for the parameter estimation. In subsequent partition, parameter estimation is carried out using Bayesian model, with prior on the statistical parameters derived using the root partition statistics, $\mathbf{P}(\theta|X) \propto \mathbf{P}(X|\theta)\mathbf{P}(\theta)$ [9]. This process ensures derivation of robust statistics less sensitive to outliers even with sparse data (Algorithm 1).

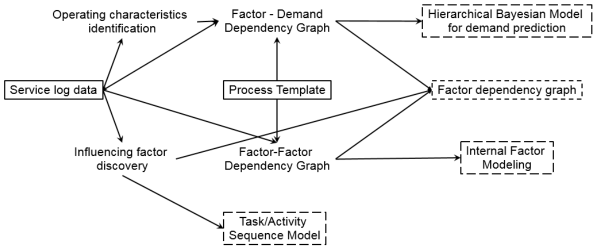

Operating characteristics identification
Factor - Demand Dependency Graph
Hierarchical Bayesian Model for demand prediction
Service log data
Process Template
Factor dependency graph
Influencing factor discovery
Factor-Factor Dependency Graph
Internal Factor Modeling
Task/Activity Sequence Model

Figure 5: Schematic of various concepts in our method, and their execution order dependency. Solid boxes are the inputs to the training phase. The dashed boxes are the final produced models.

Algorithm 1 Hierarchical Bayes Model

```
procedure RECURSIVE PARAMETER ESTIMATION
    Data set $\{x_i\}_{i=1...N}$
    Parameter from parent distribution $\theta_p$
    Sample bootstrap count $N_b$
    $\Theta \leftarrow \emptyset$
    for $i \in \{1 \ldots N_b\}$ do
        $x_s \leftarrow \mathrm{sample}(N|\theta_p)$
        $\theta_s \leftarrow MLE(x_s)$
        $\Theta \leftarrow \{\Theta, \theta_s\}$
    $\alpha \leftarrow MLE(\Theta)$
    $\theta_e \leftarrow MAP(\theta|x_i, \alpha)$        ▷ $\theta$ is the search parameter
    return $\theta_e$
```

4.5 Model Execution

Demand estimation, and task/activity sequence modeling are carried out in parallel. Both these analysis derive their own factor dependency graphs. Finally, factor dependency graphs, and influential factors for sequence models are merged together to yield the final factor dependency graph (Figure 5). Factor dependency graph is the guideline for the order of execution of different models in the scoring phase. At the end of the training process, the final derived model includes following components (1) factor dependency graph, (2) generative model for internal factors, (3) sequence generator models for task and activity, (4) hierarchical Bayesian models for resource demand estimation.

In the scoring phase, for a new input with expected volume of service requests and their associated external factors, the model estimates internal factors in the order of their dependency, as described by the factor dependency graph. Internal factor modeling often involves sequence modeling of tasks and activities. Once all the dependent factors are estimated, resource demand estimation is carried out using the hierarchical Bayesian model.

4.6 Sequence Generator

Let's assume, $\mathcal{M}$ represents the Markov transition matrix for sequence generation, and η represents a feasible state in $\mathcal{M}$. Probability of l-th state being η, can be written in a dynamic programming formulation as,

$$\mathbf{p}(\eta, l) \quad = \quad \sum_x \mathbf{p}(x, l-1)\mathcal{M}(x, \eta)$$

Boundary condition to this formulation is at $l = 0$, $\mathbf{p}(s, 0) = 1$. $\mathcal{M}$ is a stochastic matrix; i.e., for any l, $\sum_x \mathbf{p}(x, l) = 1$. But the end state (e), is an absorbing state, i.e., $\mathbf{p}(e, e) = 1$. The probability of $\mathbf{p}(e, l)$ monotonically increases with increasing l. With the length of finite sequence generated using the Markov sequence model exhibits an exponential decay. Assume a simple single state system. The state space of the Markov transition matrix, contains three states $\{s, \eta, e\}$. Assume p represents the transition probability from state $\eta \rightarrow \eta$. Then the probability vector of l-th state over three respective states can be analytically derived as $(0, p^{l-1}, 1. - p^{l-1}) \ \forall \ l > 1$.

5 EXPERIMENTAL RESULTS

To validate the model accuracy, we have created a simulated dataset for storm event scenarios using the simulator described in section 5.1. The simulation produces 30 distinct storm event scenarios over 8 locations and simulates a log containing $> 10^6$ outage entries. The simulation also generates demand data at the task and resource level granularity for each outage. 20 of these events were used for training the model, while the rest 10 were hold-out for testing purposes. The model prediction accuracy was tested on the hold-out. Demand response for each event is disjointed. Our model predicts the demand at task and resource type for each event independently.

We consider that any location is impacted by at maximum one storm event at any given time. On a given day, there can be simultaneous storm events occurring at different locations. Each storm event on average spans over 4 locations and lasts for 7 days. The solution has been validated over a real date obtained from a smart grid electricity distribution client. However, the provided data sample is not large enough to perform exhaustive experimental analysis, hence we resort to simulated dataset.

5.1 Service Outage Simulator

We have created a custom simulator for outage modeling. The model simulates service outages due to storm events in the electricity distribution grid network. The model first simulates a location topology using a planar graph, where each node of the graph represents a location, and adjacent nodes represent geospatial neighborhood locations. A random walk on the graph produces a storm trajectory along with storm severity/intensity change. Given a storm severity, we simulate daily asset (electrical equipment) outages at each location. Each outage corresponds to one service request. The location-dependent service requests are then ordered by the storm trajectory produced by the random walk. The parameters to the simulator are the number of locations, storm severity, max number of service requests (corresponds to the maximum number of assets at a given location) per day, type of service requests, and maximum storm duration at any location.

5.2 Model Prediction Accuracy

Experiment with the simulated data empirically demonstrates that (Figure 6), the model can capture the actual resource demand precisely at resource type and task level. The model produces prediction at a task and resource type level granularity. Each point in the plot (Figure 6) represents demand prediction (on Y-axis) vs actual demand (on X-axis) at a given location and a day, aggregated over all resource types. The model shows very good accuracy in predicting the total number of task volumes (Figure 6a), with a Pearson correlation coefficient, $\rho \geq 0.99$. This empirically supports that the limiting highest probability sequence is an unbiased estimator for the total task volume. The total demand estimates generated by the model correlates very well with the test data (Figure 6b), with a Pearson correlation coefficient, $\rho \geq 0.98$. This indicates hierarchical parameter estimation was able to learn unbiased estimates of model statistics. The model produces simulated demand estimates at task and resource type granularity. The simulated data does guarantee one to one correspondence to observed data at the individual task level. To compare the demand estimates at the individual task level, we used average demand per service request statistics to establish the baseline. The result shows stable homoskedastic statistics (Figure 6c), with a Pearson correlation coefficient, $\rho \geq 0.82$.

The model produces an unbiased statistics for resource demand. We empirically validated the same on the simulated data set. The simulation was carried out with 3 distinct resource types. Resource demand statistics across these resource types are distinct. The result shows that the task volume prediction and resource demand prediction is well balanced across three resource types (Figure 7).

5.3 Operating characteristic identification

We evaluate the effectiveness of distinct operating characteristics detection algorithm, by conducting experiments on synthetic data. The algorithm must identify only those operating modes which exhibit significant differences in their distribution, also uniquely identified by distinct factors. To test the same, we have designed a parametric strategy for synthesizing experimental data. We assume that the data follows normal distribution. There exists k clusters in the experimental data, and for each cluster there exist distinct statistics (μ_j, σ_j), where $j = 1, \ldots, k$. μ_j and σ_j are the mean and standard deviation of j^{th} cluster respectively. Let F_i is a categorical decision factor, taking p different values. Association of value of F_i with a cluster is described using Multinomial distribution. We assume k = 2, and p = 2.

The parameters for experimental design are α, and Z_D. Here, α models the confusion between the clusters; i.e., how distinctly the factor is characterizing the operating modes, and Z_D controls the cluster separation. The confusion matrix modelled using α can be described as below:

	cluster - 1	**cluster - 2**
factor-value: a	α	$(1-\alpha)$
factor-value: b	$(1-\alpha)$	α

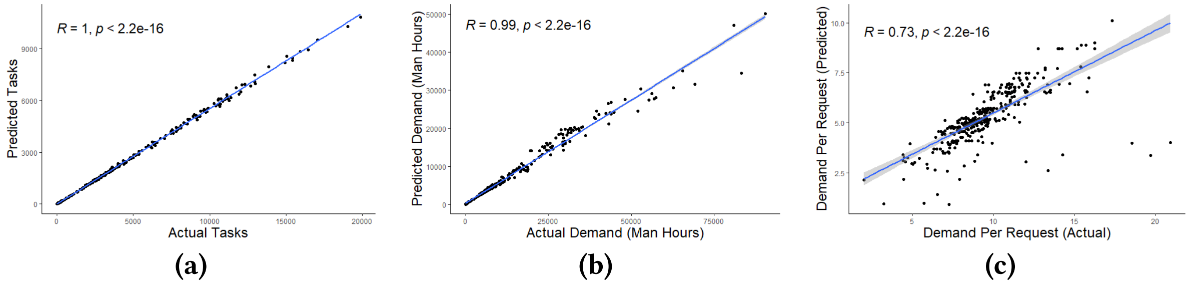


Figure 6: Model predictions aggregated over all resource types, for: (a) number of tasks, (b) resource demand, and (c) resource demand per service request.

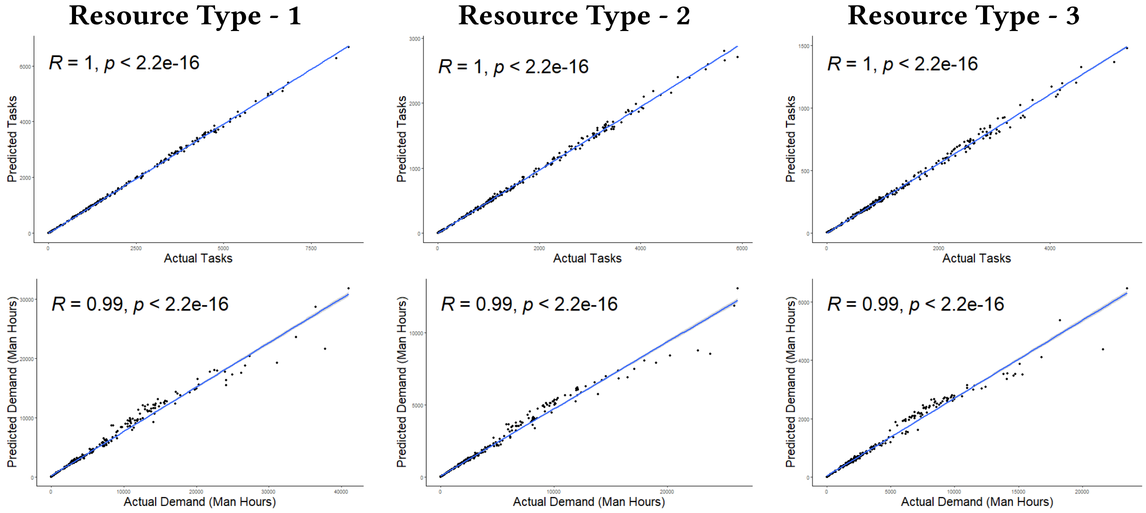


Figure 7: Model predictions presented per resource type(each column), for: number of tasks (first row), and resource demand (second row). The correlation between the prediction and actual is shown in the inset.

α takes values in the range of $[0, 1]$. $\alpha = 1$ signifies one to one correspondence between a cluster and a factor, and $\alpha = 0.5$ suggests no correlation between clusters and factors.

The cluster separation is described as $Z_D = \dfrac{|\mu_1 - \mu_2|}{\sigma_1 + \sigma_2}$. That is higher value of Z_D higher the separation between clusters.

We have carried out experiments with distinct values of δ_D ranging from (0.3 to 0.9) to demonstrate its effect on the operating characteristics detection algorithm (section 4.2). Experimental results (Figure-8) confirms lower values of δ_D results in detection of operating characteristics with high degree of cluster confusion. Higher values of δ_D identifies well separated operation characteristics, and low confusion on factor values. For robust modeling, high value of δ_D (in the range of ≥ 0.85) is best suggested.

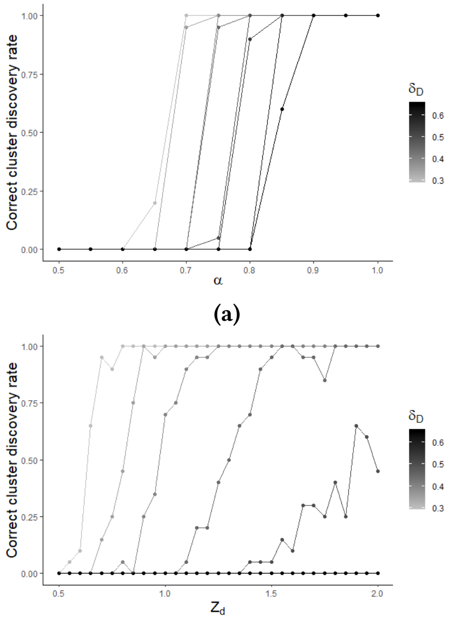
5.4 Sequence Generator

In this section, we study the effect of training data size on the reliable estimation of Markov transition matrix parameters using simulation. We generate a finite set of sequences using a finite Markov transition matrix with fixed parameter values. The generated sequences are then used to estimate the Markov model parameters using maximum likelihood. Frobenius norm of the actual and estimated Markov transition matrices difference is used as a measure of convergence of the sequence model.

Experiment shows (Figure 9) the error in the parameter estimates goes down exponentially with the size of training data. This exponential decay in error estimation guarantees reliable estimate of the dynamical system parameters even with a moderate size training data set.

5.5 Model Comparison

The proposed model automatically discovers the influencing parameters from process historical logs with process template restriction. The sequential modeling mechanism emulates the process dependencies, thereby producing a stable model. To demonstrate the same, we have compared our model with state of the art regression model (Xtreme Gradient Boost) [5]. All the system parameters are made available to the XGBoost model for the demand estimation. The model is trained and tested against the same data set described earlier. XGBoost model parameters are tuned to produce a model with best prediction accuracy on the training set. We have used the



(a)

(b)

Figure 8: Experimental validation of operating characteristic identification algorithm.

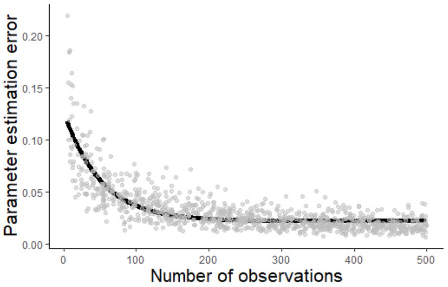
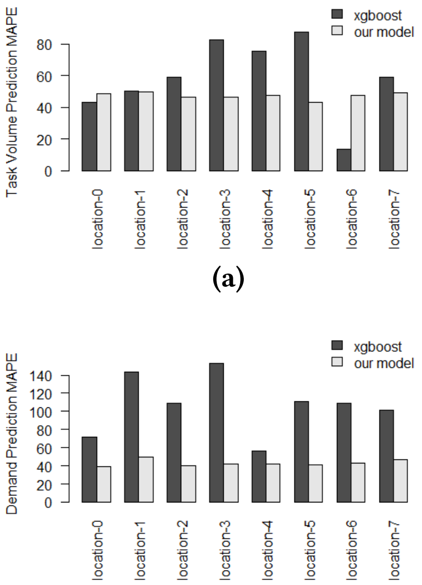


Figure 9: Markov model parameter estimation errors decays exponentially with the number of samples.

mean absolute percentage error (MAPE) as a comparison metric. We have compiled the prediction accuracy metric at all 8 locations. The experiment shows even in few instances (viz. location-0, and location-6 in Figure 10a), XGBoost model produces better task volume estimate compared to our proposed model, but failed to do so in the total resource demand estimation for those instances (Figure 10b). Our approach produces a robust (insensitive to outliers) estimate of the system parameters via hierarchical modelling. Further, due to the fact that our proposed approach estimates the demand distribution, rather than minimizing the prediction error explicitly like XGBoost model, the demand estimate produced by our approach is more accurate along with an uncertainty measure.

Our model produces an average MAPE of 45% and 43% for task and resource demand estimate respectively, compared to the values 57% and 102% produced by the XGBoost model (Figure 10). The bootstrap analysis also showed a much stable prediction behavior of our proposed model compared to the XGBoost model.



(a)

(b)

Figure 10: Mean absolute percentage error (MAPE) : (a) for task volume prediction, (b) for resource demand prediction.

6 CONCLUSIONS AND SUMMARY

Here we propose a generic method for demand estimation in a distributed service delivery system. The process template proposed on which different components are integrated are fairly generic, and can be extended to various domains of distributed service delivery. Novel contributions of our methods are, producing sequence of dependent tasks, producing sequence of dependent activities, generating associated resource demands, and finally producing a measure of uncertainty. This is extremely useful for robust resource planning to allocate resources, for the faster and cost efficient resolution of the service requests.

The proposed method uses Markovian assumption for task sequence and activity sequence generation. This makes the process less computation heavy and amenable to mathematical analysis. There are use-cases, where the activity and task sequence exhibits memory, or higher order correlation. The methods can be extended easily to accommodate such sequences generation models. Even with Markov assumption, we have attained a very robust prediction accuracy on a real life representative data-set.

Output from this method can be consumed by an optimization planner for effective planning of resource allocation to service request resolution. The demand prediction using this method, can be carried out at real time, even for inputs with partially observed data. This feature particularly is very useful for making mid-event management decisions under emergency conditions (e.g., in the event of a storm, or earth quake, or fire, etc.).

REFERENCES

[1] Mallikarjun Angalakudati, Jorge Calzada, Vivek Farias, Jonathan Gonynor, Matthieu Monsch, Anna Papush, Georgia Perakis, Nicolas Raad, Jeremy Schein, Cheryl Warren, Sean Whipple, and John Williams. 2014. Improving emergency storm planning using machine learning. In *2014 IEEE PES Conference and Exposition*. 1–6. https://doi.org/10.1109/TDC.2014.6863406

[2] A.Singhee and Haijing Wang. 2017. Probabilistic forecasts of service outage counts from severe weather in a distribution grid. In *2017 IEEE Power and Energy Society General Meeting*. 1–5. https://doi.org/10.1109/PESGM.2017.8274101

[3] P. C. Chen, T. Dokic, N. Stokes, D. W. Goldberg, and M. Kezunovic. 2015. Predicting weather-associated impacts in outage management utilizing the GIS framework. In *2015 IEEE PES Innovative Smart Grid Technologies Latin America (ISGT LATAM)*. 417–422. https://doi.org/10.1109/ISGT-LA.2015.7381191

[4] P. C. Chen and M. Kezunovic. 2016. Fuzzy Logic Approach to Predictive Risk Analysis in Distribution Outage Management. *IEEE Transactions on Smart Grid* 7, 6 (Nov 2016), 2827–2836. https://doi.org/10.1109/TSG.2016.2576282

[5] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[6] M.H. DeGroot and M.J. Schervish. 2002. *Probability and Statistics*. Addison-Wesley. https://books.google.co.in/books?id=iH4ZAQAAIAAJ

[7] D.W.Wanik, E.N.Anagnostou, B.M.Hartman, M.E.B.Frediani, and M.Astitha. 2015. Storm outage modeling for an electric distribution network in Northeastern USA. In *Natural Hazards*, Vol. 79. 1359–1384. https://doi.org/article/10.1007/s11069-015-1908-2

[8] Pablo A. Estévez, M. Tesmer, Claudio A. Perez, and Jacek M. Zurada. 2009. Normalized Mutual Information Feature Selection. *IEEE Trans. Neural Networks* 20, 2 (2009), 189–201. http://dblp.uni-trier.de/db/journals/tnn/tnn20.html#EstevezTPZ09

[9] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. 2013. *Bayesian Data Analysis, Third Edition*. Taylor & Francis. https://books.google.co.in/books?id=ZXL6AQAAQBAJ

[10] H.Li, L.A.Trenish, and J.R.M. Hosking. 2010. A statistical model for risk management of electric outage forecasts. In *IBM Journal of Research and Development*, Vol. 54. 8.1–8.11. https://doi.org/10.1147/JRD.2010.2044836

[11] Roger A. Horn and Charles R. Johnson. 2012. *Matrix Analysis* (2nd ed.). Cambridge University Press, New York, NY, USA.

[12] Padmavathy Kankanala, Sanjoy Das, and Anil Pahwa. 2014. ADABOOST(+): An Ensemble Learning Approach for Estimating Weather-Related Outages in Distribution Systems. 29 (01 2014), 359–367.

[13] J.G. Kemeny, D.S. Griffeath, J.L. Snell, and A.W. Knapp. 2012. *Denumerable Markov Chains: with a chapter of Markov Random Fields by David Griffeath*. Springer New York. https://books.google.co.in/books?id=DmXmBwAAQBAJ

[14] Young M. Lee, Soumyadip Ghosh, and Markus Ettl. 2009. Simulating distribution of emergency relief supplies for disaster response operations. *Proceedings of the 2009 Winter Simulation Conference (WSC)* (2009), 2797–2808.

[15] Zhiguo Li, A. Singhee, Haijing Wang, A. Raman, S. Siegel, Fook-Luen Heng, R. Mueller, and G. Labut. 2015. Spatio-temporal forecasting of weather-driven damage in a distribution system. In *2015 IEEE Power and Energy Society General Meeting*. 1–5. https://doi.org/10.1109/PESGM.2015.7285788

[16] L.Trenish, A.Praino, and J.Cipriani. 2011. On-going Utilization and Evaluation of a Coupled Weather and Outage Prediction Service for Electric Distribution Operations. *2nd Conference on Weather, Climate, and the New Energy Economy* (Jan 2011).

[17] D. Lubkeman and D. E. Julian. 2004. Large scale storm outage management. In *IEEE Power Engineering Society General Meeting, 2004*. 16–22 Vol.1. https://doi.org/10.1109/PES.2004.1372741

[18] Brian C. Ross. 2014. Mutual information between discrete and continuous data sets. *PLoS ONE* 9, 2 (Feb. 2014), e87357. https://doi.org/10.1371/journal.pone.0087357

[19] Mahsa Salehi, Laura Irina Rusu, Timothy Lynar, and Anna Phan. 2016. Dynamic and Robust Wildfire Risk Prediction System: An Unsupervised Approach. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. ACM, New York, NY, USA, 245–254. https://doi.org/10.1145/2939672.2939685

[20] A. Singhee, Z. Li, A. Koc, H. Wang, J. P. Cipriani, Y. Kim, A. P. Kumar, L. A. Treinish, R. Mueller, G. Labut, R. A. Foltman, and G. M. Gauthier. 2016. OPRO: Precise emergency preparedness for electric utilities. *IBM Journal of Research and Development* 60, 1 (Jan 2016), 6:1–6:15. https://doi.org/10.1147/JRD.2015.2494999

[21] J. S. Wu, T. E. Lee, C. T. Tsai, T. H. Chang, and S. H. Tsai. 2004. A fuzzy rule-based system for crew management of distribution systems in large-scale multiple outages. In *2004 International Conference on Power System Technology, 2004. PowerCon 2004.*, Vol. 2. 1084–1089 Vol.2. https://doi.org/10.1109/ICPST.2004.1460162