Relational Dynamic Bayesian Networks with Locally Exchangeable Measures

Jaesik Choi, Kejia Hu, Alex Sim

Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Relational Dynamic Bayesian Networks with Locally Exchangeable Measures

Jaesik Choi, Kejia Hu, and Alex Sim

Computational Research Division
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
{jaesikchoi,kjhu,asim}@lbl.gov

July 11, 2013

Abstract

Handling large streaming data is essential for various applications such as network traffic analysis, social networks, energy cost trends, and environment modeling. However, it is in general intractable to store, compute, search and retrieve large streaming data. This paper addresses a fundamental issue, which is to reduce the size of large streaming data and still obtain accurate statistical analysis. As an example, when a high-speed network such as 100 Gbps network is monitored, the collected measurement data rapidly grows so that polynomial time algorithms (e.g., Gaussian processes) become intractable. One possible solution to reduce the storage of vast amounts of measured data is to store a random sample, such as one out of 1000 network packets. However, such static sampling methods (linear sampling) have drawbacks: (1) it is not scalable for high-rate streaming data, and (2) there is no guarantee of reflecting the underlying distribution. In this paper, we propose a dynamic sampling algorithm that reduces the storage of data records in exponential scale, and still provides accurate analysis of large streaming data. We also build an efficient Gaussian Process with the fewer samples. We apply this algorithm to large data transfers in high-speed networks, and show that the new algorithm significantly improves the efficiency of network traffic prediction.

1 Introduction

Large streaming data is an essential part of science and engineering such as scientific computing and network communications. However, large streaming data is intractable to store, compute, search and retrieve. This paper addresses the fundamental issue of reducing the volume of large streaming data while maintaining accurate data analysis.

As a running example, suppose that we analyze network traffic measurement data. In high-speed networks, in-depth network analysis is challenging. The gathered traffic measurement data rapidly grows so that even polynomial time algorithms become intractable. In the context of traffic monitoring, one possible solution to reduce the size of the collected measurements is to store a random sample, such as one out of 1000 network packets Claise *et al.* [2009]. However, such static sampling is not scalable, and has no guarantee of reflecting the true traffic pattern. One may also think to use exact or approximate data compression techniques such as spectral analysis. However, existing data compression methods require using global data. Unfortunately, analyzing global data is a not practical problem for large streaming data.

In probability theory, it is shown that the joint probability of an infinite sequence of random measures can be represented by conditional iid (independent identically distributed) random measures, when they are exchangeable with each other de Finetti [1931]; de Finetti [1974]; Aldous [1982]. Furthermore, it is also known that an infinite (or finite) sequence of exchangeable random measures is contractible, which means that the joint probability can be exactly (or approximately) represented by a subset of the sequence Ryll-Nardzewski [1957]. The exchangeability is one of the key principles in machine learning and probabilistic inference models including variational inference Jordan *et al.* [1999]; Wainwright and Jordan [2008] and nonparametric Bayesians Ferguson [1973]; Teh *et al.* [2006] and relational probabilistic inference Carbonetto *et al.* [2005]; de Salvo Braz *et al.* [2005]. Such models have various uses in many applications such as social networks, natural language processing, video retrieval and surface water modeling.

This paper proposes a dynamic sampling algorithm, which reduces redundant network traffic in large streaming data by exploiting the exchangeability of measures. An important question is to determine the appropriate sampling rate. Certainly, sampling rates change over time based on the data pattern. We may have to store more data logs when we are uncertain about incoming data, e.g., at the beginning. However, we may reduce the amounts of data logs when we know the redundancy. We reduce the amount of data logs by exploiting two properties: (1) redundancies of data in time series; and (2) redundancies of data distributions. If data distributions of two blocks of data were similar, it would be reasonable to store less samples because the next block would be similar to the previous block with a high probability. In addition, if we know that the data distribution is concentrated only on specific values (e.g., high entropy), we can also reduce the sampling rate.

In addition to the dynamic sampling method, we demonstrate how our data reduction method can be used to improve the efficiency of a polynomial-time data analysis algorithm with an example of a Gaussian process. Many of previous sparse Gaussian processes have been focused on building a representative set for the given process Smola and Schökopf [2000]; Csató and Opper [2002]. The intuition is to construct a sparse (low-rank) covariance matrix. We show that a good alternative is to use relational Gaussian processes Chu *et al.* [2006]; Xu *et al.* [2009]. We provide a new way to build efficient relational Gaussian processes.

2 Motivations and Background

2.1 Processing Large Streaming Data

Two of main challenges of dealing with large streaming data are the storing and processing the data. In many cases, the large volume of data is brought about by improvements in technology such as large bandwidth for data transfers and high resolution sensing devices. As an example, network traffic information from a high-speed network gathers multiple terabytes per day. As shown in Figure 1 (a), the traffic information may be concentrated on a specific time and throughput range (darker area includes 90% of traffic and thus storage).

However, it may not necessary to store and process all network traffic. For the purpose of analyzing traffic throughput patterns, we can avoid handling (noisy) redundant data. As shown in Figure 1 (b), when each traffic unit is binary size (0 or 1) and iid, the sizes of packets can be modeled by a single Bernoulli parameter θ. However, in reality, such traffic data is not iid. Instead, the order of traffic can be (locally) exchangeable. In the next section, we will define exchangeable random variables and Gaussian processes, and they will be used to explain our new models, which we refer to as Bayesian Online-Locally Exchangeable Measures (BO-LEMs). These models help us solve the network traffic prediction problem addressed in this paper.

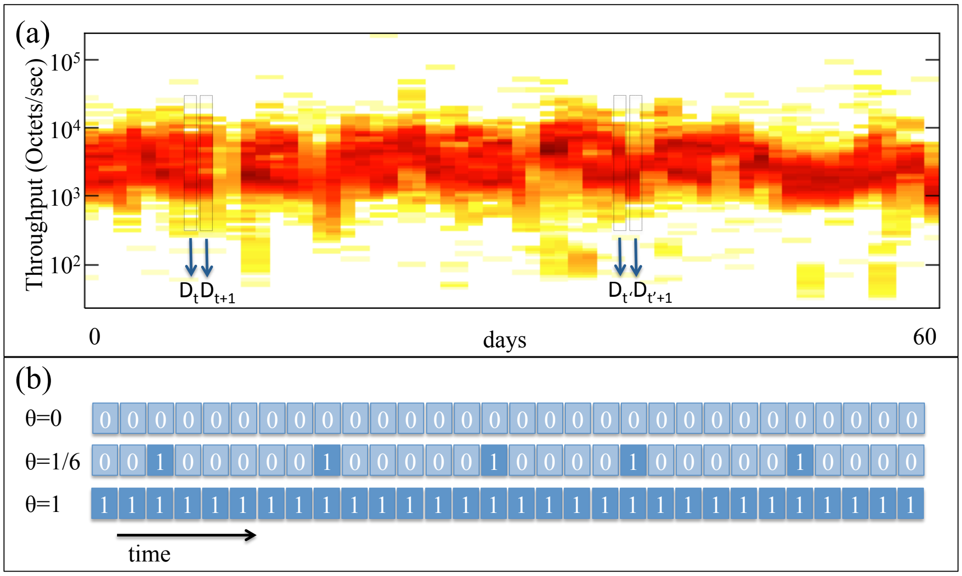


Figure 1: (a): Traffic patterns of a high-speed network router in ESnet. The regions with darker color represent higher network traffic concentrations. This figure represents network transfers as a density at a particular time (x-axis) and a throughput (y-axis). The yellow, red and black colored regions represent small, moderate and large numbers of transfers, respectively. At a specific time step, the figure represents the network transfers. As an example, two regions at D_t and D_{t+1} are network transfer profiles next to each other. As can be seen, the two regions are similar. Meanwhile, two regions at $D_{t'}$ and $D_{t'+1}$ are similar. (b): An illustration of generating iid samples from Bernoulli distribution.

2.2 Exchangeable Random Variables

A set of random variables $x_1,\cdots,x_n$ is exchangeable when the variables satisfy the following property:

$$p(x_1, \cdots, x_n) = p(x_{\pi(1)}, \cdots, x_{\pi(n)}), \quad (1)$$

where $\pi(.)$ is a permutation of n. The exchangeability assumption has been used in scientific experiments. Suppose that $x_1,\cdots,x_n$ are specific biological response in human subjects when a particular drug is injected. It is reasonable to say that the n random variables are locally exchangeable when human subjects have the same biological conditions (e.g., ethnic, gender, dose-level) [1]. Here, the variables of interests are sizes of incoming packets and throughputs (duration/size) of TCP sessions. We assume that the measures (sizes and throughputs) of network packets are exchangeable.[1] In theory, exchangeable random variables of an infinite sequence are identically distributed (but not necessarily independent). Formally, de Finetti proved the following property [2],

$$p(x_1, \cdots, x_n) = \int_{\theta \in \Theta} \prod_i p(x_i|\theta)p(\theta)d(\theta) \tag{2}$$

That is, n random variables are conditionally independent given the parameter θ of underlying (identical) distribution. In case of the binary variables, θ is the Bernoulli parameter such that $p(x_i = 1|\theta) = \theta$ and $p(x_i = 0|\theta) = 1 - \theta$. The main benefit of the de Finettie's theorem is the factorization of the joint probability of n random variables. The joint probability is now represented as the product of conditionally identically distributed random variables.

3 Models: Exchangeable Blocks

Our model focuses on adjusting the amount of data for future analysis. In our model, we assume that randomly measured samples from streaming data sources are locally exchangeable. That is, we can change the order of two measures observed within a short time, e.g., seconds. However, if the two measures are observed with a long time delay, e.g., hours, we do not assume that the two measures are exchangeable. This section describes our dynamic models with exchangeable blocks.

3.1 Locally Exchangeable Measures (LEMs)

Suppose that there is a sequence of n discrete random variables x_i, where i is the index of the variable, $\mathbf{X} = (x_1, \cdots, x_n)$. Our model splits the sequence into blocks, where each block (X_i) includes N measurements, $X_i = (x_{(i-1)N+1}, \cdots, x_{iN})$.[2]

We assume that random variables in each block X_i are exchangeable, and each block is called as a Locally Exchangeable Measure (LEM). Then, we assume that N random variables are exchangeable, when the two measures

[1] As shown in our experiments, network traffic data for some routers can be represented by conditionally iid samples. Thus, it strongly suggests that measures in a long sequence are in fact locally exchangeable each other.

[2] We assume that the block size N is given. In Section 7.2, we also provide an empirical way to find the N..

are included in a block $X_i = (x'_1, \cdots, x'_N)$. This leads to the following property with a given input sequence:

$$p(x'_1, \cdots, x'_N) = p(x'_{\pi(1)}, \cdots, x'_{\pi(N)}). \tag{3}$$

Two LEMs X_i and X_j can be exchangeable as well. This is represented by an indicator function $I_{EX}(,)$:

$$I_{EX}(X_i, X_j) = I_{EX(i,j)} = \begin{cases} 1 & p(X_{ij}) = p(X_{\pi(ij)}) \\ 0 & p(X_{ij}) \neq p(X_{\pi(ij)}) \end{cases} \tag{4}$$

where $I_{EX(i,j)}$ is a shorthand notation; $X_{ij} = X_i X_j = \left(x^i_1, \cdots, x^i_N, x^j_1, \cdots, x^j_N\right)$ $= (x_1, \cdots, x_{2N})$ and $X_{\pi(ij)} = \left(x_{\pi(1)}, \cdots, x_{\pi(2N)}\right)$.

Lemma 1 *Given two LEMs X_i and X_j, if at least one pair of random variables ($x \in X_i$, $x' \in X_j$) is exchangeable with each other, then all random variables in X_i and X_j are exchangeable.*

Proof Let the joint probability be $p(X_i, X_j) = p(x_{i,1}, \cdots, x_{i,N}, x_{j,1}, \cdots, x_{j,N})$. Without loss of generality, suppose that $x_{i,N}$ and $x_{j,N}$ are exchangeable with each other, $p(x_{i,1}, \cdots, x_{i,N}, x_{j,1}, \cdots, x_{j,N}) = p(x_{i,1}, \cdots, x_{j,N}, x_{j,1}, \cdots, x_{i,N})$. For an arbitrary pair of two variables $x'_i \in X_i$ and $x'_j \in X_j$ ($x_i \neq x_{i,N}, x_j \neq x_{i,N}$).

$$\begin{aligned} &p(x_{i,1}, \cdots, \underline{x'_i}, \cdots, x_{i,N}, x_{j,1}, \cdots, \underline{x'_j}, \cdots, x_{j,N}) \\ &\quad = p(x_{i,1}, \cdots, x_{i,N}, \cdots, \underline{x'_i}, x_{j,1}, \cdots, x_{j,N}, \cdots, \underline{x'_j}) \\ &\quad = p(x_{i,1}, \cdots, x_{i,N}, \cdots, \underline{x'_j}, x_{j,1}, \cdots, x_{j,N}, \cdots, \underline{x'_i}) \\ &\quad = p(x_{i,1}, \cdots, \underline{x'_j}, \cdots, x_{i,N}, x_{j,1}, \cdots, \underline{x'_i}, \cdots, x_{j,N}). \end{aligned}$$

Thus, x'_i and x'_j are exchangeable with each other.

4 Dynamic Bayesian Inference with LEMs

4.1 Posterior Distribution by LEMs

Predictions $p(X_{t+1})$ can be formulated by the indicator variables $\mathbf{I_{EX}}$ and their marginalization:

$$\begin{aligned} p(X_{t+1}|X_{1:t}) &= \sum_{I_{EX}} p(X_{t+1}|X_{1:t}, I_{EX(1:t)}) p(I_{EX(1:t)}|X_{1:t}) \\ &= \sum_{I_{EX}} p\left(X_{t+1}|X^{(i_{EX})}_{1:t}\right) p(I_{EX}|X_{1:t}), \end{aligned}$$

where $I_{EX(1:t)}$ refers to a set of maximum partitions[3] of exchangeable LEMs among $X_{1:t}$. A partition $i_{EX} \in I_{EX(1,t)}$ indicates which elements are exchangeable with each other, e.g., $i_{EX} = \{\{t, t-1, t-3\}, \{t-2, t-4, t-5\}, \cdots\}$. $X_{1:t}^{(i_{EX})}$ refers to a projection of $X_{1:t}$ according to the partition i_{EX}.

The important part is to predict the exchangeability of the new LEM, X_t as follow.

$$
\begin{aligned}
p(I_{EX(1:t)}, X_{1:t}) &= \sum_{I_{EX(1:t-1)}} p\left(I_{EX(*:t)}, I_{EX(1:t-1)}, X_t, X_{1:t-1}\right) \\
&= \sum_{I_{EX(1:t-1)}} p\left(I_{EX(*:t)}, X_t | I_{EX(1:t-1)}, X_{1:t-1}\right) \\
&\quad \cdot p\left(I_{EX(1:t-1)}, X_{1:t-1}\right) \\
&= \sum_{i_{EX} \in I_{EX(1:t-1)}} \underbrace{p\left(I_{EX(*,t)} | I_{EX(1:t-1)}\right)}_{\text{Transition Model}} \cdot \underbrace{p\left(X_t | X_{1:t-1}^{(i_{EX})}\right)}_{\text{Predictive Model}} \\
&\quad \cdot \underbrace{p\left(I_{EX(*:t-1)}, X_{1:t-1}\right)}_{\text{prior belief}}
\end{aligned}
$$

In the following, we explain the details of the *Transition Model*, $p\left(I_{EX(*:t)} | I_{EX(1:t-1)}\right)$, and the *Predictive Model* (*PM*), $p\left(X_t | X_{1:t-1}^{(i_{EX})}\right)$.

4.2 The Transition Model $p(I_{EX(*,t)} | I_{EX(1:t-1)})$

To handle the transition models, we provide two models based on Bernoulli processes and Additive Markov Chains.

4.2.1 Bernoulli Processes for the Transition Model

As a simple baseline model, we can use the Bernoulli process. The prediction $p\left(I_{EX(*,t)} | I_{EX(1:t-1)}\right)$ is simplified to $p\left(I_{EX(t,t-1)} | I_{EX(t-1,t-2)}, \cdots, I_{EX(2,1)}\right)$. The Bernoulli process predicts $I_{EX(t,t-1)}$, which indicates that a new LEM X_t is exchangeable with the previous one X_{t-1}. The Bernoulli parameter p_B is the sum of all events divided by the size of events,

$$
\begin{aligned}
p\left(I_{EX(t,t-1)}{=}1\right) &= \sum_{1 \le i \le t-2} \frac{I_{EX(i+1,i)}}{t-2}, \\
p\left(I_{EX(t,t-1)}{=}0\right) &= 1 - p\left(I_{EX(t,t-1)}{=}1\right)
\end{aligned}
$$

[3]The maximum partitions are partitions of largest size as long as LEMs in each partition are exchangeable.

However, this model may not handle the dynamics correctly. Thus, we use the following additive Markov chains.

4.2.2 Additive Markov Chains for the Transition Model

To take dynamic changes into account, an additive Markov chain is built where the exchangeable measures in consecutive blocks are correlated with each other.

$$p(I_{EX(*,t)}|I_{EX(1,t-1)}) \propto \sum_{i=1,\cdots,t-1} I_{EX(i,t)} \cdot exp\left(-\frac{t-i}{\sigma^2}\right)$$
$$\text{s.t. } I_{EX(i,t)} \wedge I_{EX(i,j)} \rightarrow I_{EX(j,t)}.$$

Thus, when all $t-1$ previous LEMs are exchangeable, X_t is also exchangeable with the following probability:

$$p(I_{EX(t,t-1)}=1|I_{EX(t-1,t-2)}=1,\cdots,I_{EX(2,1)}=1) \propto \sum_{i=1,\cdots,t-1} exp\left(-\frac{t-i}{\sigma^2}\right)$$

When none of m previous LEMs are exchangeable, X_i is exchangeable with a certain previous LEM at X_l with probability $c \cdot exp(-l/\sigma^2)$. If the model has the larger σ, then a longer sequence of random variables will be exchangeable. If the model has the smaller σ, it will fluctuate often and make more nonexchangeable blocks.[4] When there is a long sequence of exchangeable LEMs, a new LEM X_t is more likely to be included in the group of exchangeable LEMs.

4.3 The Predictive Model (PM) $p(X_t|X^{i_{EX}}_{1:t-1})$

4.3.1 Kolmogorov-Smirnov test

Now, we focus on computing the exchangeability of the current input X_t and each of existing partitions $X^{i_{EX}}_{1:t-1}$.

One distinctive property of exchangeable random variables is that one can interchange the order of random variables when representing the joint probability distribution. In case of discrete variables, the joint probability of X is proportional to the value histogram of X, $p(X) \propto p(h_X)$, where $h_X = (h_1, \cdots, h_k)$ and $h_i = |\{x|x = i, x \in X\}|$. In general, with continuous variables, the empirical cumulative density function (ecdf) are used as a

[4]The parameter c and σ^2 are estimated by cross-validation.

generalized version of the value histogram. Given the ecdf $F_{X'}$ of an existing partition and the ecdf F_{X_t} at time t, we use the Kolmogorov-Smirnov test [4,5] (K-S test) to measure the distance of two histograms. The K-S test indicates whether the two non-parametric distributions are sampled from the same distribution. If X' and X_t are exchangeable, they are expected to pass the K-S test. As shown in Figure 3, the test score from K-S test ($\mathcal{KS}$) is the maximum absolute difference between two ecdfs. Formally, the K-S test is represented as follow:

$$\mathcal{KS}(X', X^t) = \max_l \left(|F_{X'}(l) - F_{X_t}(l)| \right),$$

where $F_X(l) = \frac{1}{N} \sum_{\substack{x_i \in X s.t. \\ 1 \le i \le |X|}} 1\{x_i \le l\}$.

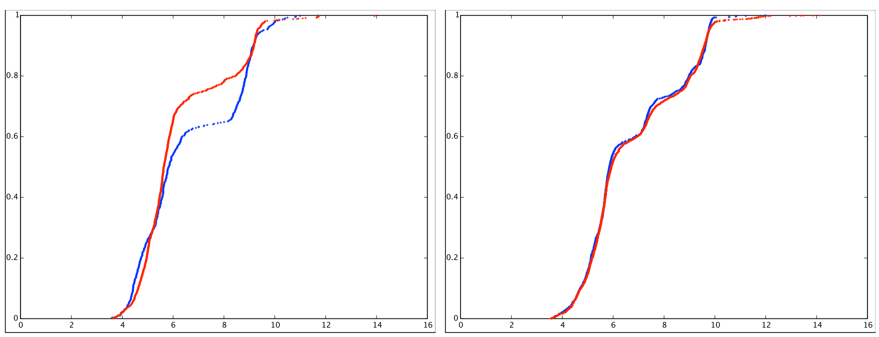


Figure 2: An example of the Kolmogorov-Smirnov test. In each graph, the two lines (the blue and the red) show the empirical distributions of two example LEMs next to each other in time. The figure on the left shows clearly a gap. In case that there are multiple gaps, the K-S test score is calculated for the largest gap between two empirical distributions. The left graph shows an example that the null hypothesis is rejected because the K-S test score is large ($\mathcal{KS}(X', X^t)$=0.1426). The right graph shows that the null hypothesis is accepted because the K-S test score is small enough.

The null hypothesis (H_0) is that the two histograms (X' and X^t) do not follow the same distribution. When $\mathcal{KS}(X', X^t) > \beta$, we reject the hypothesis, and conclude that two datasets are exchangeable. Here, we set a critical value β [4] of the test, such that the null hypothesis is rejected when p-value of the K-S test is less than 5%.

When the null hypothesis is accepted, it indicates that the new data are not exchangeable with previous observations from the previous partition.

Thus, we decrease the depth, and gather more samples from the input data. If the null hypothesis is rejected, it indicates that the new data may be exchangeable with previous observations.[5] Thus, we increase the depth in the hierarchy, and sample fewer data points by half, compared to the previous time step.

5 Algorithm: BayesianOnline-LEMs

The detail of Algorithm BayesianOnline-LEMs (or BO-LEMs) is described in this section. It reduces the data size while providing approximate but accurate data representation. We intentionally minimize the required records (or logs) for the later complex data analysis. The challenging questions are (1) how can we determine the sampling rate, and (2) how can we reduce the data size?

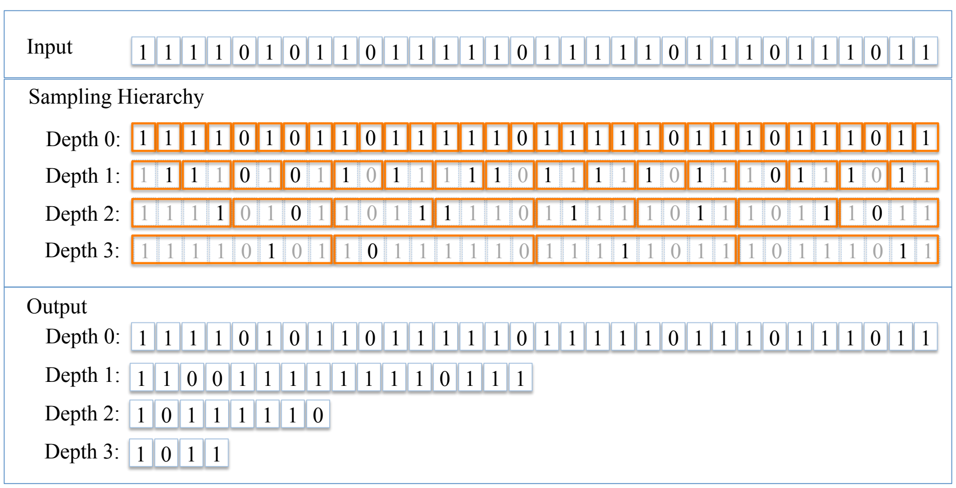


Figure 3: An illustration of the hierarchical sampling. Our algorithm collects samples based on the depth of the hierarchy (determined by the previous trends). When the depth is 0, there is no sampling. When the depth increases by 1, it reduces the number of samples by half.

Algorithm BO-LEMs receives a sequence of measurements as an input, and iterates processing the sequence N measurements per iteration. In each

[5]Even if they pass the K-S test, there is no guarantee in theory that X' and X_t are exchangeable. However, verifying the exchangeability with a limited (sequence) of data is infeasible in practice.

Algorithm 1 BayesianOnline-LEMs

```
Input: sequence of measurments D_in
dp ← 0, G = ∅, i ← 0.
while D_in(i) ≠ ∅ do
  i ← i + 1
  for all g ∈ G do {1. Compute the depth of the LEM. }
    if p(I_EX(g,i) | I_EX(g_e)) > α then
      dp ← dp + 1 {Increase depth. }
    end if
  end for
  h_i ← Sample(D_in(i), dp) {2. Get samples, 1 of 2^dp. }
  for all g ∈ G do {3. Compare with existing partitions. }
    if p-value( KS(g_h, h_i) ) ≥ β then
      (g_e, g_h) ← (g_e ∪ {i}, g_h + h_i) {Insert LEM h_i to g. }
      if |g_h| > S_max then
        (g_e, g_h) ← (g_e, trim(g_h, S_max))
      end if
      i_g ← g
      break;
    end if
  end for
  if i_g = ∅ then
    G ← G ∪ {({i}, hist_i)}
    dp ← dp − 1 {Decrease depth. }
  end if
end while
```

iteration, the algorithm computes the sampling rate of the current LEM; gathers the samples; and compares samples with existing partitions of LEMs.

Specifically, the algorithm receives a sequence of measurements D_{in} and the size of exchangeable measures, N. Initially, the depth dp is set to 0. An index variable i is used to locate current measures to read from the input data. At each step, it chooses sampled data out of the input measures $D_{in}(i)$ with a rate of $1/2^{dp}$, and then converts into a value histogram $hist_i$. The sampled data is converted, and it compares the sampled data with each partition. If there is a partition whose histogram is close enough with the $hist_i$, the algorithm continues until it reaches the last measurements of the sequence.

5.1 Subroutines

5.1.1 Computing the Depth of the Current LEM

This part of the algorithm adjusts the depth (dp) of the current LEM and accumulates the ecdf of each partition. The depth is calculated based on the additive Markov chains (Section 4). If there is a partition, which include a long sequence of exchangeable LEMs, there is a high probability that the current LEM would be included in the partition, even without analyzing it. If there is a high enough probability ($> \alpha$), we increase the depth and reduce the sampling rate by half.

5.1.2 Get Samples

Given a block of exchangeable measures, X_i, and a depth of sampling rate, dp, this subroutine extracts samples based on the depth. When depth is 0, it has no sampling, and uses all N measurement for the analysis. When the depth is increased, we reduce the number of random samples by half. That is, when the depth is k, only $N/2^k$ samples are stored for further analysis.

Figure 3 illustrates an example when N is 32 binary measures. The input sequence includes unprocessed 32 measures (0 or 1). The depth determines where the input sequence is processed. The figure illustrates 4 depths from 0 to 3. As the depth increases, the size of each block (to be sampled) is exponentially increased. One measure is randomly selected per each block. Then, we have a sequence of output measures per different depth.

5.1.3 Comparing with Existing Partitions

In the previous steps, the algorithm estimates the exchangeability of the current LEM. By using the collected samples, we evaluate whether the current LEM is exchangeable with one of existing partitions. If the LEM is shown to be exchangeable with a partition by the K-S test, the LEM is included in the current partition. Otherwise, it will form a singleton cluster. In this case, the sampling depth is decreased by 1.

5.2 Convergence Analysis

We show next that the algorithm BO-LEMs reduces the sampling size exponentially with a high probability.

Theorem 1 *Given a sequence of exchangeable LEMs $X_1, \cdots, X_n$ such that $I_{EX}(X_1, X_2), \cdots, I_{EX}(X_{n-1}, X_n)$, the depth (dp) of BO-LEMs reaches maxDepth in n steps with probability,*

$$\sum_{i=1,\cdots,n'} \binom{n}{i} \alpha^{n-i}(a-\beta)^i, \;\; \textit{where } n' - \left\lceil \frac{(n - maxDepth)}{2} \right\rceil$$

As an example, let β, n and maxDepth be 0.95, 20 and 10. Then, the probability is 0.9997.

5.3 Computational Complexity

Here, we let N be the length of the sequence $|D_{in}|$; n be the size of LEM; k be the maximum partitions that the algorithm holds; and S_{max} be the maximum number of samples. The computational complexity of the BO-LEMs is

$$O\left(N + \frac{N}{n}\left(n \log n + k \cdot \min\left(2n + \frac{N}{n}, S_{max}\right)\right)\right).$$

The first term describes that the algorithm needs to read all the measures in the sequence $O(D_{in})$ at least once. There are $O(\frac{N}{n})$ iterations. The incoming LEM needs to be sorted $O(n \log n)$ for the K-S test which compare the LEM with one of $O(k)$ partitions. Once the incoming LEM and each partition are sorted, the comparison takes only linear time which is bounded by the maximum size of each partition, $\min(O(2n + \frac{N}{n}), S_{max})$.

[4] $|d_{prev}| = \sum_{h=histp} |h|$

6 Application: Scalable Gaussian Processes

Streaming data patterns can change from one flow to another. Here, we propose a new Gaussian process to handle statistical dynamic models on LEMs.

6.1 Gaussian Process

A Gaussian process (GP) is a stochastic process X_t ($t \in T$), for which any finite linear combination of samples has a multivariate Gaussian distribution. GPs are popular models for nonparametric Bayesian, and can be considered as a nonparametric prior over a function space.

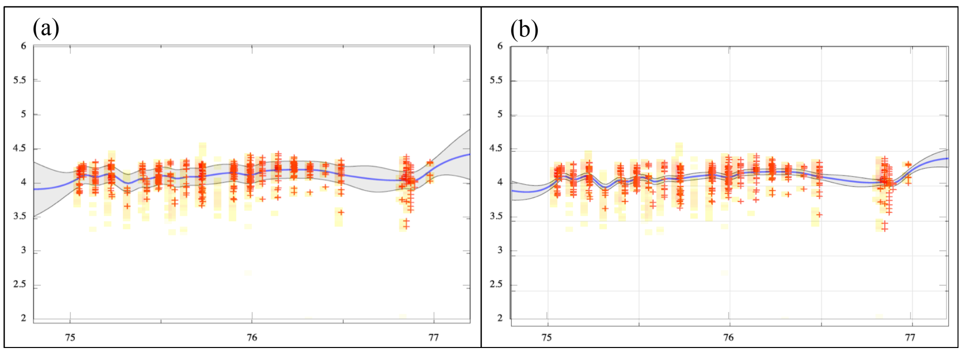


Figure 4: Two GPs with different kernel parameters given the same observations, network throughput in time. Every red dot represents a single tranfer at a particular time and a rate. The blue line is the means over the observations, and gray regions represents the variances.

Given a finite sampling of the time domain, the corresponding vector of function values f are distributed to a multivariate Gaussian with mean 0 and covariance matrix K_{GP}: $f \sim \mathcal{N}(0, K_{GP})$. where K_{GP} is determined by a kernel function. $k_{GP} : [K_{GP}]_{i,j} = k_{GP}(x_i, x_j)$. One of the typical kernel functions has the following form:

$$k_{GP}(x_i, x_j) = v_0 exp\left(-\frac{\|x_i - x_j\|^2}{\lambda}\right) + v_1.$$

This shows that the kernel adjusts the covariance of two time points x_i and x_j based on the L_2 norm of the time points. That is, if two time points are far apart, the two random measures at the points $f(x_i)$ and $f(x_j)$ have a small covariance.[6]

6.2 Kernel Function on LEMs

The key parameter of Gaussian processes is the kernel function $k_{GP}(x_i, x_j)$ that determines the covariance between two random variables. It shows that variances of two random variables in an LEM are identical, and their covariance is also the same with the variables.

Remark Given two random variables x and x' in an LEM X_i, let σ_x^2 and $\sigma_x'^2$ be the variances of x and x' and let $\sigma(x, x')$ be the covariance. Then, σ_x^2 $=\sigma_x'^2=\sigma(x, x')$.

Remark Given random variables $x, x' \in X_i$ and $x'' \in X_j$. Then, $\sigma(x, x'')=\sigma(x', x'')$. The covariance between two random variables is determined by the LEMs where the random variables are included in:

$$k_{GP}(x \in X_i, x' \in X_j) = v_0 \cdot exp\left(\frac{\|X_i - X_j\|^2}{\lambda}\right) + v_1.$$

7 Experiments

7.1 NetFlow Data in ESnet

We applied the BO-LEMs algorithm on ESnet traffic information. ESnet is a high-speed scientific network managed by the Department of Energy. We receive network traffic data from September 2012 to November 2012.[7] The network traffic data is composed of start time, end time, port and size of individual transfers during the time span. We analyze the traffic pattern for 6 backbone network routers (RTs) from RT1 to RT6, which provide NetFlow logs Claise [2004].

7.2 An Exchangeability Test

Given the NetFlow data, we need to determine that the size of LEMs. Here, we provide empirical evaluations of exchangeability of the measures in streaming data. As a reference, we prepare artificial data points which follows linear Gaussian as shown in Figure 5a. We generate 500k data points

[6] $v_1 > 0$, so that K is positive definite.

[7] The data is provided by staff at the Lawrence Berkeley National Laboratory on our request.

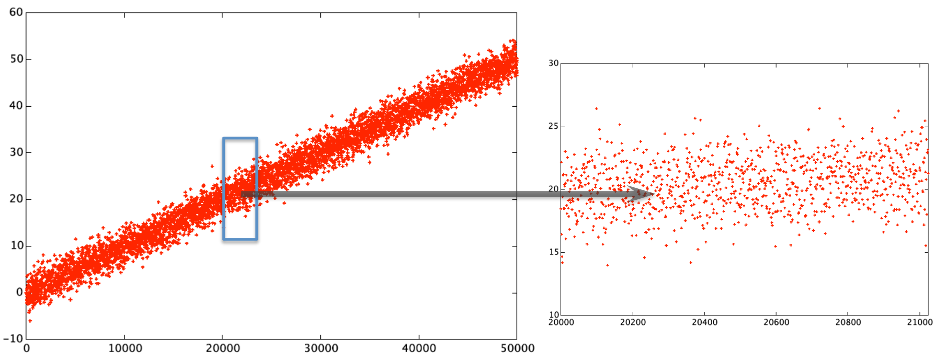
from the following linear Gaussian equation, $y = 0.001 \cdot x + 2 * e_{Normal}$ where e_{Normal} $\mathcal{N}(0, 1)$. Figure 5a shows that the points follows a linear trend. However, when it is examined locally, it is not clear to see the linear trends because of the Gaussian noise. Note that, the coefficient, 0.001, is small compared to the coefficient, 2, of the Gaussian noise. Figure 5b shows sample data points from Netflow dataset.

We measure the exchangeability of the following steps: (1) choosing a size of LEM (N); (2) randomly selecting LEMs of the size N; (3) ordering each LEM and writing down the ranks of the values at each location in LEM; and (4) after iterating the steps (2) and (3), computing the mean value of ranking at each location. Suppose that, N is 4, and we randomly select 2 LEMs, X_1= (0.1, 0.5, 0.3, 0.2) and X_2 (0.3, 0.2, 0.4, 0.5). Rakings of X_1 and X_2 at each location are (1, 4, 3, 2) and (2, 1, 3, 4), respectively. Thus, mean values are (1.5, 2.5, 3, 3). When LEMs with size 4 are in fact exchangeable, after an enough number of iterations, the mean values should be (2.5, 2.5, 2.5, 2.5). If rankings at some locations are biased, it is a strong indication that the LEMs with size N are not exchangeable.

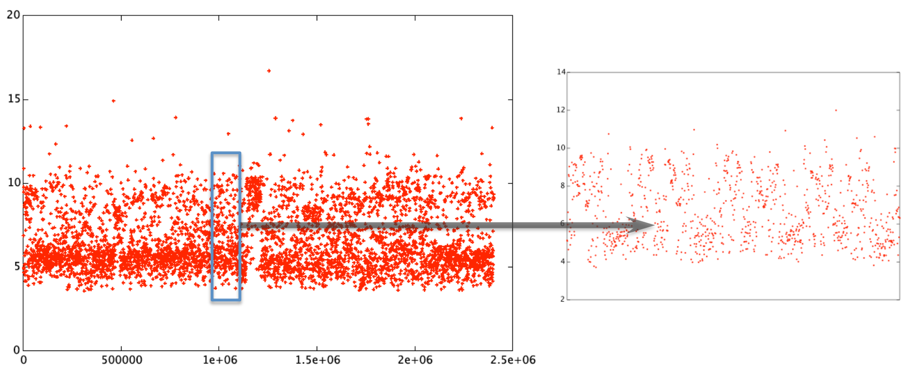
Given two datasets, we gather 2000 samples and conduct the empirical evaluation on the two datasets. Figure 5 presents the results. As we expected, when N is smaller than 64, the both data sets indicate strong exchangeability. When N is larger than 1024, the linear Gaussian data starts to present deviations which strongly indicate that the locations at the beginning in a large LEM have lower rankings than the locations at the end. When N is even larger, it clearly shows that the LEMs are not exchangeable. Meanwhile, in the Netflow data, LEMs with size 1024 indicates relatively strong exchangeability. However, when N becomes larger (N=116384), the rankings start to deviate. In intuition, it is reasonable because 1024 network transfers are occurred less than a minute in most cases.

Furthermore, we quantify the exchangeability of the two data. In theory, if LEMs with size N are exchangeable, the ranking of a particular location should follow the discrete uniform distribution [1,N]. Based on the Central Limit Theorem, we can find a confidence interval (99$) of the mean value of 2000 samples for different LEM sizes. In Table 1, we report the ratio, the number of outliers, locations beyond the 99% confidence interval, divided by N. When the ratio is larger than 1%, it indicates that the LEMs are not exchangeable in theory. Although, the Netflow data shows some deviations for larger N, it maintains relatively small number of outliers. Thus, we consider that LEMs are approximately exchangeable. [8] In the linear Gaussian

[8] Although we may need to consider the error of the Central Limit Theorem for small



(a) The exchangeability of the linear Gaussian data with different sizes of n.



(b) The exchangeability of netflow data with different sizes of n.

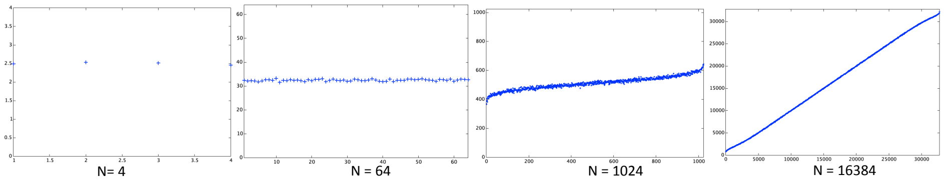
Figure 5: The empirical exchangeability of the two data sets for different LEM sizes N. The ratio is the number of outliers divided by N.

data, it is clear that the LEM are not exchangeable when N is larger than 64.

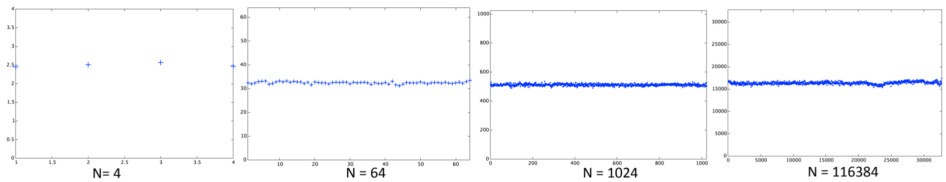
7.3 Data Reduction by BO-LEMs

The input to the BO-LEMs algorithm is a set of sequences (of network transfers) for routers. The BO-LEMs algorithm computes the required samples of each LEM block to represent the underlying distribution correctly. Here, we set the size of block N to be 1024. When the algorithm finds that a long sequence of measures are sampled from the same distribution, the sampling

numbers of samples, an exact evaluation of the exchangeability is the subject of future work.



(a) The exchangeability of the linear Gaussian data with different sizes of N.



(b) The exchangeability of netflow data with different sizes of N.

sizes are reduced by half, that is 512, 256, $\cdots$, 32 (the minimum number of samples). Figure 6 (a) and (b) respectively illustrate the changes of the required samples over time from the routers labeled 'RT5' and 'RT2'. The sample sizes reflects the different characteristics of two routers in different time steps. In general, the 'RT5' router requires less samples than the 'RT2' router. The red lines represent how the required samples change over time. As a reference, we plot the blue line, which represents sizes of input LEMs. The x axes of the figures are the id number of LEM ordered by incoming time. The y axis is the number of samples.

Table 2 presents data reduction rates for 6 routers.

8 Conclusions and future work

In this paper, we present new data reduction models, Locally Exchangeable Measures (LEMs) and a dynamic sampling algorithm, Bayesian-Online LEMs (BO-LEM). As a running example and an application, we applied the BO-LEM algorithm to a high-speed network dedicated to scientific data transfers. We shows that our algorithm can reduce the number of required samples by 66% while achieving accurate data analysis. The LEMs and the BO-LEM algorithm are used to build efficient relational Gaussian process.

We have looked at data comes from a source that varies greatly - network transfer. In this case, we achieved moderate data reductions, but with high accuracy. We worked with this data because we wanted to target a challenging problem. We believe that other sources such as environmental data from sensors (e.g., temperature) which do not vary dramatically, will

N	Netflow	Linear Gaussian
2	0	0
4	0	0
8	0	0
16	0	0
32	0	0
64	3.13	6.25
128	1.56	4.69
256	1.17	17.19
512	1.36	45.7
1024	1.66	69.82
2048	1.12	84.57
4096	1.59	92.53
8192	1.55	95.95
16384	2.32	96.74
32768	5.81	96.67

Table 1: The exchangeability test of two datasets.

have much higher data reduction while still keeping accuracy. We plan to apply our techniques to such data sources in the future.

Another possibility for future research is to modify the parameters of our algorithms dynamically as we get more and more data over time and can see the longer term trends (beyond just looking at the close neighborhood).

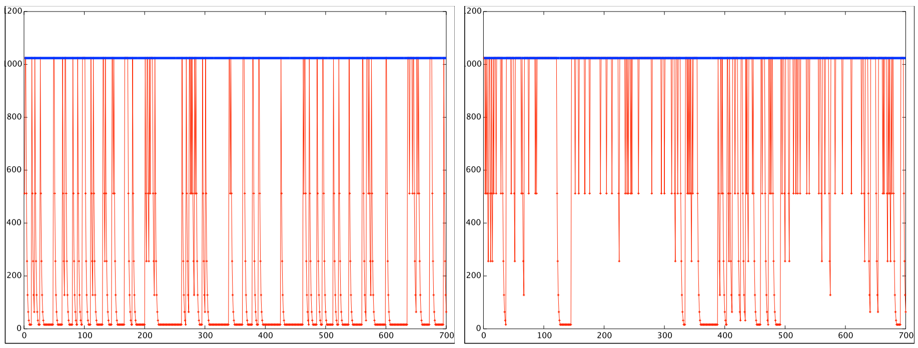
9 Acknowledgement

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

References

David J. Aldous. On exchangeability and conditional independence. In *Exchangeability in probability and statistics (Rome, 1981)*, pages 165–170. North-Holland, Amsterdam, 1982.

Peter Carbonetto, Jacek Kisynski, Nando de Freitas, and David Poole. Non-



(c) Samples at 'RT5' (d) Samples at 'RT2'

Figure 6: A example of data reduction on the RT5 router and the RT2 router. Originally, each LEM has 1024 transfers (blue line). Our BO-LEM algorithm reduces the size of each block (red line) when incoming traffic has similar properties to the previous blocks. Thus, it keeps only the smallest numbers of transfers necessary such as 32, 64, $\cdots$. Each figure represents time on the x-axis and the number of samples on the y-axis. As can be seen, there are red dots at 32, 64, 128, 256, 512 and 1024. The regions in Figure 5 (a) that show multiple dots at 32 represent the regions are very similar, and they only need 32 samples to represent them. The regions on the Figure 5 (b) have many consecutive 512 dots represent higher variation cannot be compressed further.

parametric bayesian logic. In *Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 85–93, Arlington, Virginia, 2005. AUAI Press.

Wei Chu, Vikas Sindhwani, Zoubin Ghahramani, and S. Sathiya Keerthi. Relational learning with gaussian processes. In *NIPS*, pages 289–296, 2006.

B. Claise, A. Johnson, and J. Quittek. Packet Sampling (PSAMP) Protocol Specifications. RFC 5476 (Proposed Standard), March 2009.

B. Claise. Cisco Systems NetFlow Services Export Version 9. RFC 3954 (Informational), October 2004.

Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

Router	Total Transfers	Sampled Transfers	Reduction Rate
RT1	33.6M	11.3M	66.5%
RT2	28.1M	14.7M	47.5%
RT3	15.8M	3.0M	80.9%
RT4	14.4M	2.6M	71.6%
RT5	9.2M	2.7M	70.9%
RT6	10.8M	2.9M	73.6%
Total	112.5M	37.8M	66.4%

Table 2: Total transfers, sampled transfers and data reduction rate of 6 backbone routers in ESnet.

B. de Finetti. *Funzione Caratteristica Di un Fenomeno Aleatorio*, pages 251–299. 6. Memorie. Academia Nazionale del Linceo, 1931.

B. de Finetti. *Theory of Probability.* John Wiley & Sons, 1974.

Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. Lifted first-order probabilistic inference. In *IJCAI*, pages 1319–1325, 2005.

Thomas S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

Michael I. Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

Czeslaw Ryll-Nardzewski. On stationary sequences of random variables and the de finetti's equivalence. *Colloquium Mathematicae*, 4(2):149–156, 1957.

Alex J. Smola and Bernhard Schökopf. Sparse greedy matrix approximation for machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 911–918, 2000.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

Zhao Xu, Kristian Kersting, and Volker Tresp. Multi-relational learning with gaussian processes. In *IJCAI*, pages 1309–1314, 2009.