MPME-DP: Multi-Population Moment Estimation via Dirichlet Process for Efficient Validation of Analog/Mixed-Signal Circuits

Manzil Zaheer and Xin Li
ECE Department, Carnegie Mellon University
Pittsburgh, PA 15213
{manzil, xinli}@cmu.edu

Chenjie Gu
Strategic CAD Labs, Intel Corporation
Hillsboro OR 97124
chenjie.gu@intel.com

ABSTRACT

Moment estimation is one of the most important tasks to appropriately characterize the performance variability of today's nanoscale integrated circuits. In this paper, we propose an efficient algorithm of multi-population moment estimation via Dirichlet Process (MPME-DP) for validation of analog and mixed-signal circuits with extremely small sample size. The key idea is to partition all populations (e.g., different environmental conditions, setup configurations, etc.) into groups. The populations within the same group are similar and their common knowledge can be extracted to improve the accuracy of moment estimation. As will be demonstrated by the silicon measurement data of a high-speed I/O link, MPME-DP reduces the moment estimation error by up to 65% compared to other conventional estimators.

1. INTRODUCTION

For the task of circuit validation, we must accurately evaluate the circuit performances of interest. However, these quantities are today no longer well represented by deterministic models and thus to estimate them one must resort to statistical tools due to two reasons [1]–[2]: (i) aggressive scaling of IC technology brings about large-scale process variations, and (ii) ever increasing system complexity leaves the system susceptible to uncertainties posed by environmental conditions and/or surrounding circuits. Statistically, the randomness of a performance metric must be characterized by its distribution. Once the distribution is estimated, it can be further used to evaluate parametric yield, qualify a product, guide design optimization, and/or facilitate process tuning.

There are numerous approaches for density estimation in the statistics literature [3]–[5]. However, most of these methods come with a caveat: they typically operate in the big data regime, requiring a sufficiently large amount of data to obtain accurate results. For analog and mixed-signal (AMS) circuit validation, the trustworthy data are often collected by post-layout simulation (for pre-silicon validation) or silicon measurement (for post-silicon validation). Such a data collection process is highly expensive, taking several days or even weeks to finish [6]–[8]. As a result, due to short time-to-market windows, only a limited amount of data (e.g., very few samples) may be affordable. The technical challenge here is how to accurately estimate the AMS performance distributions under the "small-sample-size" constraint.

Recently, a novel statistical framework of *Bayesian model fusion* (BMF) has been proposed to address the aforementioned issue of small sample size [9]–[14]. In particular, a *multi-population moment estimation* (MPME) algorithm [11]–[12] has been developed with the assumption that the performance distribution is Gaussian and the data collected at all populations (e.g., different environmental conditions, setup configurations, etc.) are highly correlated. Such correlation information is exploited by MPME to improve the accuracy of moment estimation. To this end, a statistical model is proposed wherein a prior distribution is used to encode the common knowledge across all populations and then maximum-a-posteriori (MAP) estimation is carried out for moment estimation at each population. Once the first-order and second-order moments (i.e., mean and variance) are known, the Gaussian probability density function (PDF) can be uniquely determined for the performance of interest [11]–[12].

The MPME method proposed in [11]–[12], however, is not perfect. It is expected that an AMS system may fail at a particular population and, hence, its performance distribution can be substantially different from other normal cases. In addition, the performance distributions at a number of populations may be similar, but they can be different from the distributions at other populations, as will be demonstrated by a silicon measurement data set in Section 5.2. These observations imply that it is not wise to pool all populations together and extract the common knowledge by assuming that all these populations are similar. Instead, we must first partition the populations into different groups based on their "similarity", and then apply MPME to each group where the populations within the same group are similar.

Nonetheless, this simple and seemingly useful idea cannot be easily implemented in practice, as our data set is plagued by its small sample size (e.g., 5 samples only for each population). Most conventional clustering algorithms [3]–[5] cannot successfully identify the true underlying groups because of large random variations in the observed data. In other words, with extremely small sample size, the boundaries between different groups become "blurred" due to random variations and it is almost impossible to clearly separate these groups with conventional clustering algorithms.

In this paper, we propose to borrow the idea of *Dirichlet process* (DP) [15]–[17] from the statistics community and develop a new algorithm of *multi-population moment estimation via Dirichlet process* (MPME-DP) to overcome the aforementioned technical challenge. In particular, a unified flow is proposed to integrate both clustering and density/moment estimation together which has twofold innovation. First, we build up a detailed statistical model to accurately capture the in-group "correlation" and ignore the between-group "distortion" and thus reduce modelling error. Second, since the boundaries between different groups cannot be clearly defined based on limited data, MPME-DP aims to set up multiple possible boundaries where each boundary is assigned with a probability based on its likelihood of occurrence. In other words, unlike the conventional clustering algorithms that generate a single, deterministic partition, MPME-DP produces multiple possible partitions in order to take into account the randomness of the observed data. Furthermore, unlike the conventional MPME method that is limited to Gaussian distribution [11]–[12], the proposed approach is generally applicable to other performance distributions (e.g., log-normal, chi-squared, etc.). Our experimental results in Section 5 demonstrate that MPME-DP

978-1-4799-6278-5/14/$31.00 ©2014 IEEE

can appropriately handle industrial circuit examples and reduce the moment estimation error by up to 65%, compared to other conventional estimators.

The remainder of paper is organized as follows. In Section 2, we formally describe the problem and review the background. Next, we propose our MPME-DP approach in Section 3 and describe its implementation details in Section 4. The efficacy of MPME-DP is demonstrated by a number of experimental examples, including industrial measurement data, in Section 5. Finally, we conclude in Section 6.

2. BACKGROUND

In this paper, we target the problem of moment estimation for a given AMS performance metric over multiple populations:

$$\{x_m; m = 1, 2, ..., M\}, \tag{1}$$

where x_m denotes the performance metric associated with the m-th population, and M is the total number of populations. Here, a population refers to a particular environmental condition, setup configuration, etc. Suppose that we obtain a set of statistically independent samples:

$$\mathcal{X} = \{x_m^{(n)}; m = 1, 2, ..., M; n = 1, 2, ..., N_m\}, \tag{2}$$

where N_m stands for the number of samples for the m-th population.

The technical challenge of moment estimation stems from the fact that the sample sizes $\{N_m; m = 1, 2, ..., M\}$ are extremely small, because collecting a large amount of data can be highly expensive, if not impossible, for both pre-silicon simulation and post-silicon measurement. Conventionally, the moments are often estimated by averaging the samples in (2). For instance, the mean and variance of the performance metric x_m at the m-th population can be estimated by:

$$\begin{aligned} \tilde{\mu}_m &= \frac{1}{N_m} \sum_{n=1}^{N_m} x_m^{(n)} \\ \tilde{\sigma}_m^2 &= \frac{1}{N_m - 1} \sum_{n=1}^{N_m} \left(x_m^{(n)} - \tilde{\mu}_m\right)^2, \end{aligned} \tag{3}$$

where $\tilde{\mu}_m$ and $\tilde{\sigma}_m^2$ denote the empirical estimators for mean and variance, respectively. However, with an extremely small data set, the conventional estimators in (3) are not sufficiently accurate.

2.1 Multi-Population Moment Estimation

To address the accuracy issue associated with the conventional estimators, MPME has been proposed in [11]–[12] to estimate the mean and variance particularly for Gaussian distributions. Its key idea is to impose a prior distribution:

$$p(\mu_m, \sigma_m^2 | \boldsymbol{\psi}) \quad (m = 1, 2, ..., M), \tag{4}$$

where $p(\cdot)$ denotes the PDF of a continuous random variable, $\left\{(\mu_m, \sigma_m^2) : m = 1, 2, ..., M\right\}$ represents the mean and variance values of M populations, and the vector $\boldsymbol{\psi}$ contains a set of hyper-parameters to parameterize the prior distribution $p(\mu_m, \sigma_m^2 | \boldsymbol{\psi})$. A Bayesian inference is then applied to estimate the moments $\left\{(\mu_m, \sigma_m^2) : m = 1, 2, ..., M\right\}$ via two steps. First, the optimal value of $\boldsymbol{\psi}$ in (4) is learned from the data $\left\{x_m^{(n)}; m = 1, 2, ..., M; n = 1, 2, ..., N_m\right\}$ over all populations by using maximum likelihood estimation (MLE). Next, given the prior distribution $p(\mu_m, \sigma_m^2 | \boldsymbol{\psi})$ in (4) where the parameter $\boldsymbol{\psi}$ is already determined, a MAP estimation is further applied to each population to estimate the mean μ_m, and the variance σ_m^2.

The MPME method, however, has several shortcomings. It is mentioned in [11]–[12] that an abnormal population (i.e., an outlier) cannot be directly handled by MPME, since it may strongly bias the estimation results. In addition, MPME cannot easily handle clustered populations where a number of populations are similar but they are different from other populations. In this case, the prior knowledge of $\left\{(\mu_m, \sigma_m^2) : m = 1, 2, ..., M\right\}$ cannot be simply encoded by a single prior distribution in (4) over all populations. Finally, the MPME implementation described in [11]–[12] is limited to Gaussian distribution only.

2.2 Population Grouping

The aforementioned discussions of MPME suggest that we should not pool all populations together and extract the common knowledge by assuming that all populations are similar. Instead, we must first cluster the populations into different groups based on their similarity. Namely, two populations should be assigned to the same group, if and only if they are both similar to each other. Once the clustering step is complete, MPME should be applied to the populations within the same group for moment estimation.

There are a large number of clustering algorithms developed in the statistics community that can be possibly adopted here. For instance, hierarchical clustering [5] is one of the popular methods. When applying hierarchical clustering to our application of MPME, we take the moments estimated by the conventional estimators (e.g., the empirical estimators in (3) for mean and variance) as the input. Starting with M groups for M populations (i.e., each group initially containing one population), we merge the two most similar groups at each iteration step until there is only a single group left or a stopping criterion is met.

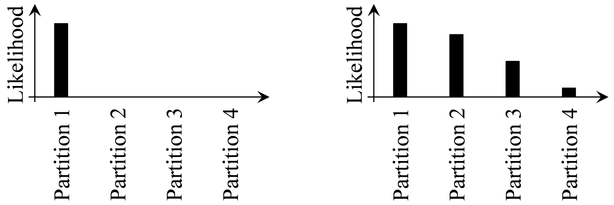
Hierarchical clustering is expected to accurately identify the clustered data structure and detect the abnormal populations (i.e., the outliers). However, when the observed data set is extremely small, the moments estimated by the conventional estimators may not be accurate and, hence, the boundaries between different groups become "blurred". For this reason, hierarchical clustering, as well as other conventional clustering algorithms, often fails to work, as will be demonstrated by our experimental example shown in Figure 5 of Section 5.1. It, in turn, motivates us to fundamentally re-think the conventional wisdom of data clustering in order to further develop a novel algorithm for our proposed application of moment estimation.

3. PROPOSED APPROACH

In this section, we describe the proposed MPME-DP method. In particular, we will explain the details of its Bayesian model based on DP.

3.1 "Elastic" Clustering

As previously discussed, correctly identifying the underlying groups based on an extremely small data set is not trivial. Most conventional clustering algorithms (e.g., hierarchical clustering) attempt to make a "hard" decision on the clustering result. Namely, they deterministically partition the populations into different groups and each population is assigned to a single group only. Such a strategy is built upon the assumption that the underlying groups are clearly separable. However, this fundamental assumption does not hold in our application, since our observed data are subject to large variations. Taking hierarchical clustering as example, it relies on the moments estimated by the conventional estimators that may not be highly accurate due to small sample size. Hence, hierarchical clustering is likely to generate a wrong result that does not match the actual structure of the data.



(a) "Hard" Clustering (b) "Elastic" Clustering

Figure 1: The conventional "hard" clustering finds a single, deterministic partition only whereas the "elastic" clustering in DP finds multiple possible partitions with each partition having its likelihood of occurrence.

To address this technical challenge, we borrow a radically new idea from statistics community [15]–[17] where the objective is not to find a single, deterministic partition. Instead, it aims to make a "elastic" decision on the clustering result. In other words, we will generate "multiple" possible partitions of different sizes where each partition is assigned to a probability based on its likelihood of occurrence. The optimal estimator is obtained through model averaging, i.e. taking the expectation of multiple models associated with these possible partitions.

Figure 1 intuitively illustrates the difference between "hard" and "elastic" clustering. Due to the randomness posed by small sample size, there exist a number of possible partitions over the populations of interest. When a "hard" clustering algorithm is applied, it only finds one (hopefully the "best") of these possible partitions, while ignoring the others, as shown in Figure 1(a). Even if the identified partition has a high probability to occur, the other possible partitions may be equally good or, at least, their likelihood of occurrence is not negligible. Hence, "hard" clustering may not reveal the full structure of the data set and, consequently, fails to work.

On the other hand, "elastic" clustering aims to statistically capture all possible partitions. As a result, it is unlikely to bias towards a specific partition and is expected to be more accurate than "hard" clustering. In this paper, our implementation of "elastic" clustering is based upon the Bayesian framework developed by the statistics community [3]–[5], as will be discussed in detail in the following sub-sections.

3.2 Bayesian Modelling

We assume that the performance x_m at the m-th population follows a parameterized PDF:

$$x_m \sim p(x_m|\boldsymbol{\eta}_m), \tag{5}$$

where $\boldsymbol{\eta}_m$ represents a set of parameters specifying the distribution. There are two important clarifications that should be made for the PDF $p(x_m|\boldsymbol{\eta}_m)$ in (5). First, the parameters in $\boldsymbol{\eta}_m$ are generally different for different populations, implying that the distributions are different for these populations. For instance, Table 1 shows two examples (i.e., Gaussian and log-normal distributions) to define the parameterized PDF $p(x_m|\boldsymbol{\eta}_m)$ where the values of η_1 and η_2 can be different for different populations. Second, since the PDF is uniquely specified by $\boldsymbol{\eta}_m$, the moments of x_m are also uniquely determined by $\boldsymbol{\eta}_m$. In other words, once the parameters $\boldsymbol{\eta}_m$ are known, we can calculate the moments of x_m as a function of $\boldsymbol{\eta}_m$, as shown by the examples in Table 1. For this reason, our moment estimation problem can be cast to an equivalent problem of estimating the unknown parameters $\boldsymbol{\eta}_m$.

Similar to the conventional MPME method [11]–[12], we attempt to exploit the correlation between different populations to maximize the estimation accuracy. To encode this common knowledge across multiple populations, a prior distribution for $\boldsymbol{\eta}_m$ needs to be imposed. However, we must carefully distinguish the populations that are inherently different from each other. Otherwise, arbitrarily imposing a similarity assumption among substantially different populations can introduce a large bias to the estimation result. Therefore, we cannot use a single prior distribution to model all populations; instead, we need different prior distributions for different groups where each group contains one or multiple similar populations.

Table 1: Two examples (i.e. Gaussian and log-normal) of parametric PDFs and their moments up to third order

Distribution	**Gaussian**	**Log-Normal**
Parameters	η_1 and η_2	η_1 and η_2
PDF	$\frac{1}{\sqrt{2\pi\eta_2}}\exp\left\{-\frac{(x-\eta_1)^2}{2\eta_2}\right\}$	$\frac{1}{\sqrt{2\pi\eta_2}x}\exp\left\{-\frac{(\ln x-\eta_1)^2}{2\eta_2}\right\}$
Mean	η_1	$\exp(\eta_1+\eta_2/2)$
Variance	η_2	$[\exp(\eta_2)-1]\exp(2\eta_1+\eta_2)$
Skewness	0	$[\exp(\eta_2)+2]\sqrt{\exp(\eta_2)-1}$

To this end, we first introduce the indicator variables $\{c_m; m = 1, 2, ..., M\}$. The m-th indicator variable c_m defines the group that the m-th population belongs to. If we partition all populations into K groups, each indicator variable can take one of the K integer values: $c_m \in \{1, 2, ..., K\}$. Based on these indicator variables, we define our prior knowledge as a parameterized distribution:

$$\boldsymbol{\eta}_m \sim p(\boldsymbol{\eta}_m|\boldsymbol{\psi}_{c_m}), \tag{6}$$

where $\boldsymbol{\psi}_{c_m}$ contains a set of parameters describing the distribution. For instance, the prior distribution is defined as a parameterized Normal-inverse-Chi-squared (NIX) PDF in [12]. Note that each group (say, the c_m-th group) should be associated with a unique prior distribution $p(\boldsymbol{\eta}_m|\boldsymbol{\psi}_{c_m})$ with its own value for $\boldsymbol{\psi}_{c_m}$. In total, there are K different values for $\boldsymbol{\psi}_{c_m}$, (i.e., $\{\boldsymbol{\psi}_k; k = 1, 2, ..., K\}$, corresponding to K different groups.

Figure 2 illustrates a simple example of using different prior distributions for different groups present in the data. In this example, there are eight populations in total that can be partitioned into two groups. It is natural to make all the populations in the left group to share common knowledge and likewise for the right group, while not between these two groups. To do this, we need to use two different prior distributions $p(\boldsymbol{\eta}_m|\boldsymbol{\psi}_1)$ and $p(\boldsymbol{\eta}_m|\boldsymbol{\psi}_2)$ with different parameters $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$ for the two groups respectively.

Since the values of $\{\boldsymbol{\psi}_k; k = 1, 2, ..., K\}$ are different for different groups, we further assume that they follow a shared prior distribution:

$$\boldsymbol{\psi}_k \sim p(\boldsymbol{\psi}_k). \tag{7}$$

Conceptually, each $\boldsymbol{\psi}_k$ can be viewed as a a random sample drawn from $p(\boldsymbol{\psi}_k)$. It, in turn, results in different values for $\{\boldsymbol{\psi}_k; k = 1, 2, ..., K\}$. The distribution $p(\boldsymbol{\psi}_k)$ can be chosen in several different

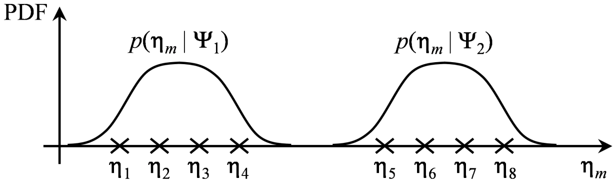


Figure 2: An example of eight populations belonging to two groups is shown for illustration purposes. Two different prior distributions $p(\boldsymbol{\eta}_m|\boldsymbol{\psi}_1)$ and $p(\boldsymbol{\eta}_m|\boldsymbol{\psi}_2)$ with different parameters $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$ are used for these two groups respectively.

ways. For example, it can be a uninformative (e.g. a uniform distribution over the entire range) or engineered to encode the domain-specific knowledge from experts.

When a conventional clustering algorithm is applied, the groups and, hence, the indicator variables $\{c_m; m = 1, 2, ..., M\}$ are defined deterministically. However, since we consider "elastic" clustering in this paper, there exist multiple possible partitions among the populations where each partition is assigned to a probability of occurrence. In this case, the indicator variables $\{c_m; m = 1, 2, ..., M\}$ should be considered as random variables that follow a discrete distribution over $\{1, 2, ..., K\}$:

$$c_m \sim p(c_m) = \sum_{k=1}^{K} \pi_k \delta_k(c_m), \tag{8}$$

where $p(\cdot)$ denotes the probability mass function (PMF) of a discrete random variable,

$$\delta_k(c_m) = \begin{cases} 1 & \text{if } c_m = k \\ 0 & \text{if } c_m \neq k \end{cases} \quad (k = 1, 2, ..., K) \tag{9}$$

denotes the k-th component of $p(c_m)$, and π_k represents the weight of the k-th component. To simplify our notation, we use the same symbol $p(\cdot)$ to represent both PDF and PMF in this paper, since the meaning of $p(\cdot)$ is self-explained in the context.

The weight values $\{\pi_k; k = 1, 2, ..., K\}$ in (8) must satisfy the following constraints:

$$\begin{gathered} \pi_k \geq 0 \quad (k = 1, 2, ..., K) \\ \sum_{k=1}^{K} \pi_k = 1. \end{gathered} \tag{10}$$

Otherwise, the PMF $p(c_m)$ in (8) may not be non-negative or its summation over all possible values of c_m may not equal 1. In this case, $p(c_m)$ in (8) is no longer a valid PMF.

Eq. (5)-(8) defines the Bayesian model for MPME-DP. Two important clarifications should be made here. First, the Bayesian model is defined hierarchically. Namely, the prior distribution $p(x_m|\boldsymbol{\eta}_m)$ is parameterized by $\boldsymbol{\eta}_m$ in (5), $\boldsymbol{\eta}_m$ follows the prior distribution $p(\boldsymbol{\eta}_m|\boldsymbol{\psi}_{c_m})$ parameterized by $\boldsymbol{\psi}_{c_m}$ in (6), and finally $\boldsymbol{\psi}_k$ and c_m are specified by the prior distributions $p(\boldsymbol{\psi}_k)$ in (7) and $p(c_m)$ in (8) respectively. Second, but more importantly, we do not know the number of groups (i.e., K) in practice. Hence, the prior distribution $p(c_m)$ in (8) must cover all possible values of K. To address this challenge, we will adopt the idea of DP from the statistics community to define the prior distribution $p(c_m)$. The details of DP will be discussed in the next sub-section.

3.3 Dirichlet Process

Since the value of K is unknown in (8), the DP [15]–[17] attempts to define the distribution $p(c_m)$ where K can possibly vary from 1 to an extremely large value (e.g., $K \to \infty$ in the extreme case). As such, we are able to consider all possible values for K that must be a positive integer. It is important to note that these different scenarios with different K values do not occur with equal probability. Once the measurement data are observed, we will solve the posterior distribution for $\{c_m; m = 1, 2, ..., M\}$ to identify a small number of "active" scenarios with high probability to occur, as will be discussed in Section 4.

There are several different yet equivalent ways to mathematically describe DP. In this paper, we borrow the stick breaking model described in [18]. It starts from a set of random variables $\{\beta_k; k = 1, 2, ..., K\}$ that follow a Beta distribution:

$$p(\beta_k|\alpha) = \alpha(1 - \beta_k)^{\alpha-1} \quad (k = 1, 2, ..., K), \tag{11}$$

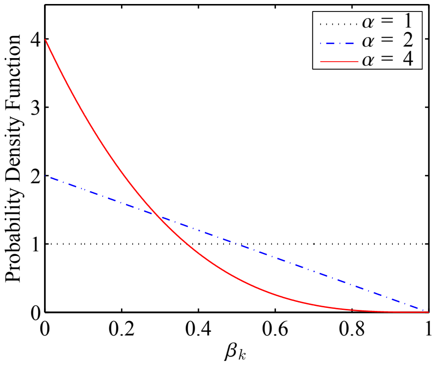


Figure 3: Several different Beta distributions are shown with different values of α.

where β_k is within the interval [0, 1], and α is a parameter that should be empirically set. Several general guidelines to set α can be found in [19]. Figure 3 shows a few different Beta distributions with different values of α.

Next, we define a set of new random variables $\{\pi_k : k = 1, 2, ..., K\}$ based on $\{\beta_k : k = 1, 2, ..., K\}$:

$$\pi_k = \beta_k \prod_{i=1}^{k-1} (1 - \beta_i) \quad (k = 1, 2, ..., K). \tag{12}$$

There are two important properties carried by these new random variables $\{\pi_k : k = 1, 2, ..., K\}$ in (12). First, since $1-\beta_i$ is within the interval [0, 1], the magnitude of π_k decays as k increases. Second, but more importantly, it can be proven that the following equality holds, as K approaches infinite [18]:

$$\lim_{K \to \infty} \sum_{k=1}^{K} \pi_k = 1. \tag{13}$$

The property in (13) implies that the variables $\{\pi_k : k = 1, 2, ..., K\}$ in (12) can be used as the weight values to define our prior distribution $p(c_m)$ when K approaches infinite, i.e.:

$$p(c_m) = \sum_{k=1}^{\infty} \pi_k \delta_k(c_m). \tag{14}$$

In theory, the distribution $p(c_m)$ in (8) is composed of an infinite number of components $\{\delta_k(c_m); k = 1, 2, ...\}$. In practice, however, since π_k decays with k, the value of π_k is almost zero once k is sufficiently large (say, $k \geq K_{DP}$). In other words, only the first K_{DP} components are "active". The number of "active" components (i.e., K_{DP}) heavily depends on the value of α in (11). As shown in Figure 3, if α is small, the values of $\{\beta_k : k = 1, 2, ...\}$ in (11) are likely to be large. In this case, π_k decays quickly with k and, hence, only a small number of components would be active for $p(c_m)$ in (8). Otherwise, if α is large, a large number of components may be active for $p(c_m)$.

Eq. (11)-(14) describes the basic mathematical framework for DP. Combining (5)-(8) and (11)-(14), our hierarchical Bayesian model is now fully defined for MPME-DP. Next, we need to further solve the Bayesian inference to estimate the moments for all populations, as will be discussed in the following sub-section.

3.4 Moment Estimation

Given the measurement data $\mathcal{X}$ in (2) and the Bayesian model defined in Section 3.2-3.3, MPME-DP estimates the moments of interest via three major steps. First, we consider different possible partitions of the populations and solve the Bayesian inference to find the posterior distribution $p(\boldsymbol{\eta}_m|\mathcal{X})$. The posterior PDF $p(\boldsymbol{\eta}_m|\mathcal{X})$ tells us the probability of observing different possible values of $\boldsymbol{\eta}_m$, after the measurement data $\mathcal{X}$ is known. Second, we calculate the Bayes estimator for $\boldsymbol{\eta}_m$:

$$\hat{\boldsymbol{\eta}}_m = \int \boldsymbol{\eta}_m p\left(\boldsymbol{\eta}_m|\mathcal{X}\right) d\boldsymbol{\eta}_m. \qquad (15)$$

The estimator $\hat{\boldsymbol{\eta}}_m$ in (15) essentially averages all possible values of $\boldsymbol{\eta}_m$ based on its posterior probability. Finally, since the PDF $p(x_m|\boldsymbol{\eta}_m)$ in (5) is fully specified by $\boldsymbol{\eta}_m$ for the performance x_m at the m-th population, we can easily calculate the moments of x_m once $\boldsymbol{\eta}_m$ is known.

Implementing the aforementioned three steps, however, is not trivial. In particular, since the DP model in (11)-(14) defines numerous possible partitions for the populations, directly solving the posterior distribution $p(\boldsymbol{\eta}_m|\mathcal{X})$ with consideration of all these partitions may be computationally intractable. To address this issue, we will further develop a computationally efficient implementation for MPME-DP in the next section.

4. IMPLEMENTATION DETAILS

As shown in (15), the value of $\boldsymbol{\eta}_m$ is estimated by calculating the average based on the posterior distribution $p(\boldsymbol{\eta}_m|\mathcal{X})$. Unfortunately, analytically calculating the expectation in (15) is extremely difficult, if not impossible. Hence, we resort to Monte Carlo method to estimate the expected value:

$$\hat{\boldsymbol{\eta}}_m \approx \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\eta}_m^{(t)}, \qquad (16)$$

where $\boldsymbol{\eta}_m^{(t)}$ denotes the t-th random sample drawn from the posterior PDF $p(\boldsymbol{\eta}_m|\mathcal{X})$, and T is the total number of random samples. Towards this goal, we must know the posterior distribution $p(\boldsymbol{\eta}_m|\mathcal{X})$ in the first place, before we can draw random samples from it. This, however, is another challenging task, since it is not trivial to analytically solve $p(\boldsymbol{\eta}_m|\mathcal{X})$ from the Bayesian inference. To overcome this issue, we will take a detour to employ a sampling strategy, referred to as Gibbs sampling in the statistics community [20], for obtaining random samples from the posterior PDF $p(\boldsymbol{\eta}_m|\mathcal{X})$ and finally being able to estimate $\boldsymbol{\eta}_m$ by (16). In what follows, we will describe the Gibbs sampling approach and highlight its novelty.

4.1 Gibbs Sampling

To apply Gibbs sampling, we first re-write the Bayes estimator in (15) as:

$$\hat{\eta}_m = \iiint \boldsymbol{\eta}_m p\left(\boldsymbol{c}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}|\mathcal{X}\right) d\boldsymbol{c}d\boldsymbol{\Lambda}d\boldsymbol{\Psi}, \qquad (17)$$

where

$$\begin{aligned} \boldsymbol{c} &= [\ c_1 \quad c_2 \ \cdots \ c_M\ ] \\ \boldsymbol{\Lambda} &= [\ \boldsymbol{\eta}_1 \quad \boldsymbol{\eta}_2 \ \cdots \ \boldsymbol{\eta}_M\ ] \\ \boldsymbol{\Psi} &= [\ \boldsymbol{\psi}_1 \quad \boldsymbol{\psi}_2 \ \cdots \ \boldsymbol{\psi}_K\ ]. \end{aligned} \qquad (18)$$

At first glance, the expression in (17) seems more complicated than the one in (15). However, as will be explained in this sub-section, sampling the "joint" posterior distribution $p\left(\boldsymbol{c}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}|\mathcal{X}\right)$ in (17) is substantially easier than the "marginal" posterior distribution $p\left(\boldsymbol{\eta}_m|\mathcal{X}\right)$ in (15).

In general, Gibbs sampling is a Markov Chain Monte Carlo method to obtain samples from complicated joint distributions [20]. To achieve this goal, Gibbs sampling iteratively generates a sample from the conditional distribution of each variable. For example, if we want to sample from the joint PDF $p(z_1, z_2, ..., z_R)$, where $\{z_r; r = 1, 2, ..., R\}$ denotes a set of random variables and R is the total number of these random variables, Gibbs sampling starts from an initial point $\left[z_1^{(0)} z_2^{(0)} ... \ z_R^{(0)}\right]$ and it draws a sequence of random samples based on conditional probabilities: $z_1^{(1)} \sim p\left(z_1|z_2^{(0)}, z_3^{(0)}, ..., z_R^{(0)}\right)$, $z_2^{(1)} \sim p\left(z_2|z_1^{(1)}, z_3^{(0)}, ..., z_R^{(0)}\right)$, etc. It can be mathematically proven that these Gibbs samples constitute a Markov chain and their stationary distribution follows the given joint PDF $p(z_1, z_2, ..., z_R)$ [20].

In our case, the joint posterior distribution $p\left(\boldsymbol{c}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}|\mathcal{X}\right)$ in (17) contains three categories of random variables: (i) the indicator variables $\boldsymbol{c}$, (ii) the distribution parameters $\boldsymbol{\Lambda}$, and (iii) the cluster parameters $\boldsymbol{\Psi}$. By following the Gibbs sampling idea, we need to iteratively sample each of these random variables from its conditional posterior distribution given the current values of all other random variables. To this end, we need to form the conditional posterior distributions for $\boldsymbol{c}$, $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$, respectively.

- **The indicator variables $\boldsymbol{c}$**: For each c_m where $m \in \{1, 2, ..., M\}$, we adopt the statistical method from [21] to derive the conditional PDF:

$$p\left(c_m|\boldsymbol{c}_{\backslash\{c_m\}}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}, \mathcal{X}\right) \propto p\left(c_m|\boldsymbol{c}_{\backslash\{c_m\}}\right) p\left(\boldsymbol{\eta}_m|\boldsymbol{\psi}_{c_m}\right), \qquad (19)$$

where $\boldsymbol{c}_{\backslash\{c_m\}}$ denotes the vector $\boldsymbol{c}$ with the element c_m removed. In (19), $p\left(c_m|\boldsymbol{c}_{\backslash\{c_m\}}\right)$ is a discrete PMF that can be analytically derived [21]. Given $p\left(c_m|\boldsymbol{c}_{\backslash\{c_m\}}\right)$ and $p\left(\boldsymbol{\eta}_m|\boldsymbol{\psi}_{c_m}\right)$, we can compute the discrete PMF $p\left(c_m|\boldsymbol{c}_{\backslash\{c_m\}}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}, \mathcal{X}\right)$ by (19). Once $p\left(c_m|\boldsymbol{c}_{\backslash\{c_m\}}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}, \mathcal{X}\right)$ is known, we can draw random samples from it by using inverse transform sampling [20].

- **The cluster parameters $\boldsymbol{\Psi}$**: For each $\boldsymbol{\psi}_k$ where $k \in \{1, 2, ..., K\}$, we need to sample the following conditional distribution:

$$p\left(\boldsymbol{\psi}_k|\boldsymbol{c}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}_{\backslash\{\psi_k\}}, \mathcal{X}\right), \qquad (20)$$

where $\Psi_{\backslash\{\psi_k\}}$ denotes the matrix $\boldsymbol{\Psi}$ with the column $\boldsymbol{\psi}_k$ removed. The PDF $p\left(\boldsymbol{\psi}_k|\boldsymbol{c}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}_{\backslash\{\psi_k\}}, \mathcal{X}\right)$ cannot be analytically found. Hence, we need to apply another Markov chain Monte Carlo in the inner loop of Gibbs sampling to sample from $p\left(\boldsymbol{\psi}_k|\boldsymbol{c}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}_{\backslash\{\psi_k\}}, \mathcal{X}\right)$. For example, the No-U-Turn sampler (NUTS) [22] can be used for this purpose.

- **The distribution parameters $\boldsymbol{\Lambda}$**: To sample $\boldsymbol{\eta}_m$ where $m \in \{1, 2, ..., M\}$, we need to know the following conditional distribution:

$$p\left(\boldsymbol{\eta}_m|\boldsymbol{c}, \boldsymbol{\Lambda}_{\backslash\{\eta_m\}}, \boldsymbol{\Psi}, \mathcal{X}\right), \qquad (21)$$

where $\boldsymbol{\Lambda}_{\backslash\{\eta_m\}}$ denotes the matrix $\boldsymbol{\Lambda}$ with the column $\boldsymbol{\eta}_m$ removed. In general, solving the analytical expression for $p\left(\boldsymbol{\eta}_m|\boldsymbol{c}, \boldsymbol{\Lambda}_{\backslash\{\eta_m\}}, \boldsymbol{\Psi}, \mathcal{X}\right)$ is extremely difficult. Hence, we have to apply another Markov chain Monte Carlo (e.g., NUTS [22]) in the inner loop of Gibbs sampling to sample from $p\left(\boldsymbol{\eta}_m|\boldsymbol{c}, \boldsymbol{\Lambda}_{\backslash\{\eta_m\}}, \boldsymbol{\Psi}, \mathcal{X}\right)$. However, if the prior distribution $p\left(\boldsymbol{\eta}_m|\boldsymbol{\psi}_{c_m}\right)$ in (6) and the performance distribution $p(x_m|\boldsymbol{\eta}_m)$ in (5) are conjugate, the analytical expression of $p\left(\boldsymbol{\eta}_m|\boldsymbol{c}, \boldsymbol{\Lambda}_{\backslash\{\eta_m\}}, \boldsymbol{\Psi}, \mathcal{X}\right)$ can be found and, hence, sampling from it becomes much easier.

When implementing the aforementioned Gibbs sampling algorithm, we must consider a large number of groups formed by the populations. As discussed earlier, the number of groups can approaches infinite (i.e., $K \rightarrow \infty$) in the extreme case. To efficiently generate Gibbs samples, we repeatedly perform the following operations: (i) identifying the group to which each population belongs (i.e. determining the indicator variable c_m) (ii) updating

Algorithm 1 Multi-Population Moment Estimation via Dirichlet Process (MPME-DP)

Inputs: The data set: $\mathcal{X} = \{x_{mn} : m = 1, 2, ..., M; n = 1, 2, ..., N_m\}$
Outputs: The desired moments for performance at each population

```
Set the initial values $\boldsymbol{c}^{(0)}$ for the indicator variables, $\boldsymbol{\Lambda}^{(0)}$ for the distribution parameters, and $\Psi^{(0)}$ for the cluster parameters.
for $t = 1 \to T$ do
    for $m = 1 \to M$ do
        Apply Gibbs sampling to draw a new indicator variable $c_m^{(t)}$ according to the conditional posterior PDF in (19)
    end for
    for $k = 1 \to K$ (active number of clusters only) do
        Apply Gibbs sampling to draw a new cluster parameter $\psi_k^{(t)}$ according to the conditional posterior PDF in (20)
    end for
    for $m = 1 \to M$ do
        Apply Gibbs sampling to draw a new distribution parameters $\eta_m^{(t)}$ according to the conditional posterior PDF in (21)
    end for
end for
Calculate the Bayes estimator $\hat{\boldsymbol{\eta}}_m$ in (17) for the distribution parameters $\boldsymbol{\eta}_m$, where $m \in \{1, 2, ..., M\}$
Based on the PDF in (5), calculate the desired moments for the performance $x_m$ at the $m$-th population, where $m \in \{1, 2, ..., M\}$
```

the cluster parameter $\boldsymbol{\psi}_k$ based on the clustering result defined by the indicator variables $\{c_m; m = 1, 2, ..., M\}$, and (iii) using the common knowledge within the updated groups to re-calculate the distribution parameter $\boldsymbol{\eta}_m$. Note that during the last two operations, i.e., in step (ii) and (iii), we only need to handle the groups which have at least one population assigned to it (i.e., the "active" groups). In practice, only a small number of groups will be simultaneously active for a particular Gibbs sample. However, as we generate more and more Gibbs samples over iterations, we will be able to visit an increasingly large number of possible groups.

The aforementioned discussions summarize the basic idea of solving our proposed Bayesian inference by Gibbs sampling. Once a set of Gibbs samples are generated, we can estimate the expected value of $\boldsymbol{\eta}_m$ by (16) and, consequently, the moments of the performance x_m, where $m \in \{1, 2, ..., M\}$. Due to the page limit, a number of mathematical derivations and implementation details are omitted in the paper. More background information about Gibbs sampling and Markov chain Monte Carlo can be further found in [20]–[22].

4.2 Summary

Algorithm 1 summarizes the major steps of the proposed MPME-DP method. Starting from a set of random samples $\mathcal{X}$, we first initialize the indicator variables $\boldsymbol{c}$, the distribution parameters $\boldsymbol{\Lambda}$, and the cluster parameters $\boldsymbol{\Psi}$. Next, Gibbs sampling is applied to generate a set of random samples $\left\{\left(\boldsymbol{c}^{(t)}, \boldsymbol{\Lambda}^{(t)}, \boldsymbol{\Psi}^{(t)}\right); t = 1, 2, ..., T\right\}$ and the expected values of $\{\boldsymbol{\eta}_m; m = 1, 2, ..., M\}$ are estimated by these samples. Finally, once the parameters $\boldsymbol{\eta}_m$ are determined, we know the performance distribution $p(x_m|\boldsymbol{\eta}_m)$ in (5) and can estimate the moments of each performance x_m based on $\boldsymbol{\eta}_m$.

It is important to note that a number of Gibbs samples generated at the beginning of the iterations in Algorithm 1 may not follow the desired posterior distribution $p(\boldsymbol{c}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}|\mathcal{X})$, since a Markov chain must go through an initial "burn-in" period before reaching its stationary distribution [20]. For this reason, we must remove these initial Gibbs samples when calculating the Bayes estimator $\boldsymbol{\eta}_m$ in (17). Otherwise, these samples may substantially distort our estimation results.

5. EXPERIMENTAL RESULTS

In this section, the efficacy of MPME-DP is demonstrated by several experimental examples. In these examples, we assume that the AMS performance metrics of interest follow Gaussian distribution [1] even though our proposed MPME-DP method is applicable to other non-Gaussian distributions in practice. Our objective here is to estimate the moments $\left\{(\mu_m, \sigma_m^2) : m = 1, 2, ..., M\right\}$ based on the observed data $\left\{x_m^{(n)} : m = 1, 2, ..., M; n = 1, 2, ..., N_m\right\}$ for all populations. For testing and comparison purposes, four different algorithms are implemented:

- **Empirical**: The empirical estimators in (3) are used to estimate the mean and variance for each population separately.

- **MPME**: MPME [11]–[12] is applied to estimate the mean and variance based on the common knowledge extracted from all populations.

- **MPME-HIE**: Hierarchical clustering is first applied to partition the populations into different groups and then MPME [11]–[12] is applied to estimate the mean and variance for the populations within the same group.

- **MPME-DP**: The proposed MPME via Dirichlet process (i.e., Algorithm 1) is applied for mean and variance estimation.

5.1 Synthetic Data

In this sub-section, we consider a synthetic example with 11 populations. The mean and variance values of these 11 populations are defined as:

$$\begin{aligned} \mu_m &= \begin{cases} 8.5 & \text{if } m = 1, 2, ..., 10 \\ 10.5 & \text{if } m = 11 \end{cases} \\ \sigma_m^2 &= \begin{cases} 0.85 & \text{if } m = 1, 2, ..., 10 \\ 1.15 & \text{if } m = 11 \end{cases} \end{aligned} \tag{22}$$

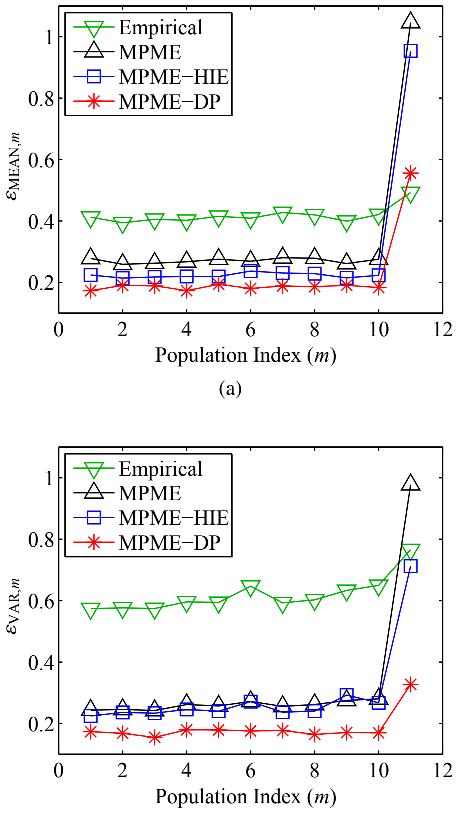
Note that the first 10 populations in (22) share the same mean and variance, while the last population is completely different. In other words, the last population should be considered as an outlier. Ideally, we should first detect and remove the outlier (i.e., the last population), and then apply MPME to extract the common knowledge across the first 10 populations for mean and variance estimation.

In this examples, 5 independent random samples are generated for each population. We apply four different algorithms (i.e., Empirical, MPME, MPME-HIE, and MPME-DP) to estimate the mean and variance values for all populations. We repeatedly run our experiments with independently generated data sets for 500 times and calculate the average errors:

$$\begin{aligned} \varepsilon_{\text{MEAN},m} &= \sqrt{\frac{1}{500}\sum_{i=1}^{500}\left(\hat{\mu}_m^{(i)} - \mu_m\right)^2} \\ \varepsilon_{\text{VAR},m} &= \sqrt{\frac{1}{500}\sum_{i=1}^{500}\left(\hat{\sigma}_m^{2(i)} - \sigma_m^2\right)^2}, \end{aligned} \tag{23}$$

where $\hat{\mu}_m^{(i)}$ and $\hat{\sigma}_m^{2(i)}$ denote the estimated mean and variance of the m-th population at the i-th run, respectively.

Figure 4 plots the estimation errors for all 11 populations. Studying Figure 4 reveals several important observations. First, even though MPME is able to achieve good accuracy for the first



(a)

(b)

Figure 4: Average errors calculated from 500 repeated runs are plotted at 11 populations for (a) mean estimation, and (b) variance estimation.

10 populations, it results in an extremely large error for the last population (i.e., the outlier). This observation is consistent with our expectation that since MPME assumes similarity among all populations including the outlier, it strongly biases the estimation results at the outlier. Since an outlier corresponds to a specific population (e.g., a specific environmental corner), it implies that the validation result is inaccurate and the circuit behavior is not appropriately assessed at this population.

Second, but more importantly, MPME-HIE fails to detect the outlier in this example. To intuitively understand the reason, we plot the hierarchical clustering result in Figure 5 for one of the 500 runs. In this case, even though the actual mean values are 8.5 for the first 10 populations, the empirical mean values estimated from 5 random samples vary from 7.6 to 9.1. Given such a large variation range, hierarchical clustering merges the outlier with other normal populations. Since the outlier is not correctly detected, the results of both mean and variance estimation are strongly biased at the outlier, as shown in Figure 4.

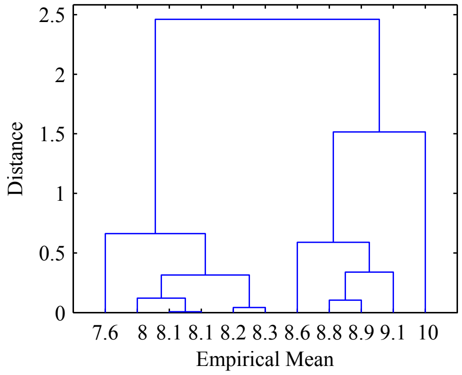


Figure 5: A representative example where hierarchical clustering fails to correctly detect the outlier.

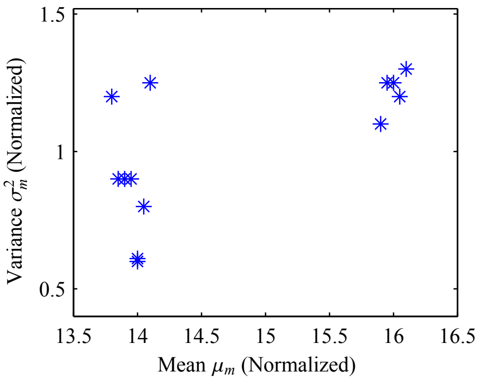


Figure 6: Mean and variance values are plotted for 13 different populations from silicon measurement.

Finally, MPME-DP is able to accurately detect the outlier and, hence, its results are not substantially biased at the outlier, as shown in Figure 4. For the normal populations, MPME-DP provides superior accuracy over all other estimators. It reduces the average error by up to 24% and 42% for mean and variance estimation, respectively. For these reasons, MPME-DP is preferred over other moment estimation algorithms in this example.

5.2 Silicon Measurement Data

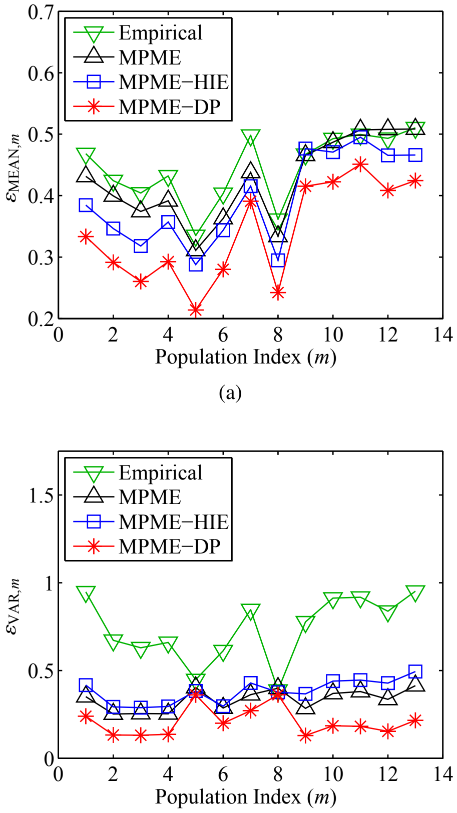
In this sub-section, we apply MPME-DP to a data set that is obtained by measuring the receiver eye width of a high-speed I/O link. Since such a silicon measurement is extremely time-consuming, we are only able to collect the data from 50 dies with 13 different populations. We calculate the empirical mean and variance values based on all 50 dies for different populations, as shown in Figure 6. Note that three different groups are observed for these 13 populations.

To validate the accuracy of the different estimators we adopt a bootstrap approach [4]. Namely, we randomly select 5 dies out of 50 candidates for each population, and apply four different algorithms (i.e., Empirical, MPME, MPME-HIE and MPME-DP) to estimate the mean and variance values based on these selected dies. The average errors are computed by (23) where the empirical mean and variance values calculated from all 50 dies are considered as the "golden" results for error evaluation. Figure 7 compares the accuracy of mean and variance estimation for different algorithms. Note that MPME-DP achieves the minimal error, since it successfully identifies the clustered data structure in this example. Compared to other conventional methods, MPME-DP reduces the average error by up to 26% and 65% for mean and variance estimation, respectively.

Finally, it is worth mentioning that the proposed MPME-DP method takes about 5 minutes to run on a server with 2.2GHz CPU and 64GB memory in this example. In practice, collecting the measurement data often takes a few days or even weeks and is substantially more expensive than running MPME-DP. Hence, the measurement cost often dominates the overall validation cost and the computational time of MPME-DP is almost negligible for the practical application of AMS validation.

6. CONCLUSIONS

In this paper, a novel MPME-DP algorithm is proposed for efficient moment estimation of analog and mixed-signal



(a)

(b)

Figure 7: Average errors calculated from 500 repeated runs are plotted at 13 populations for (a) mean estimation, and (b) variance estimation.

circuits. MPME-DP attempts to improve the estimation accuracy with extremely small sample size by taking advantage of the data collected from multiple populations (e.g., different environmental conditions, setup configurations, etc.). Built upon the conventional MPME method [11]–[12], MPME-DP can further handle clustered data and outliers, which are often observed in practical applications, by adopting a Bayesian approach based on Dirichlet process. The proposed MPME-DP algorithm has been validated on two different data sets, including the silicon measurement data of a high-speed I/O link. Our experimental results demonstrate that MPME-DP consistently out-performs other conventional estimators with up to 65% error reduction. The aforementioned accuracy improvement can be directly translated to a valuable cost reduction for analog and mixed-signal validation.

7. ACKNOWLEDGEMENTS

This work has been supported in part by National Science Foundation under contract CCF-1316363 and Intel Corporation.

8. REFERENCES

[1] X. Li, J. Le, and L. T. Pileggi, *Statistical Performance Modeling and Optimization*, Now Publishers, 2007.

[2] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2011.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd Ed, Springer, 2009.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer New York, 2006.

[5] K. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.

[6] G. Balamurugan, B. Casper, J. Jaussi, M. Mansuri, F. O'Mahony, and J. Kennedy, "Modeling and analysis of high-speed I/O links," *IEEE Trans. on Advanced Packaging*, vol. 32, no. 2, pp. 237–247, May 2009.

[7] J. Keshava, N. Hakim, and C. Prudvi, "Post-silicon validation challenges: how EDA and academia can help," in *ACM/IEEE DAC*, pp. 3–7, June 2010.

[8] C. Gu, "Challenges in post-silicon validation of high-speed I/O links," in *IEEE/ACM ICCAD*, pp. 547–550, Nov 2012.

[9] X. Li, W. Zhang, F. Wang, S. Sun, and C. Gu, "Efficient parametric yield estimation of analog/mixed-signal circuits via bayesian model fusion," in *IEEE/ACM ICCAD*, pp. 627–634, Nov 2012.

[10] F. Wang, W. Zhang, S. Sun, X. Li, and C. Gu, "Bayesian model fusion: large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data," in *ACM/IEEE DAC*, pp. 1–6, May 2013.

[11] C. Gu, E. Chiprout, and X. Li, "Efficient moment estimation with extremely small sample size via bayesian inference for analog/mixed-signal validation," in *ACM/IEEE DAC*, pp. 1–7, May 2013.

[12] C. Gu, M. Zaheer, and X. Li, "Multiple-population moment estimation: exploiting inter-population correlation for efficient moment estimation in analog/mixed-signal validation," *IEEE Trans. on CAD*, vol. 33, no. 7, pp. 961–974, July 2014.

[13] S. Sun, F. Wang, S. Yaldiz, X. Li, L. Pileggi, A. Natarajan, M. Ferriss, J. Plouchart, B. Sadhu, B. Parker, A. Valdes-Garcia, M. Sanduleanu, J. Tierno, and D. Friedman, "Indirect performance sensing for on-chip analog self-healing via Bayesian model fusion," in *IEEE CICC*, pp. 1–4, Sept 2013.

[14] X. Li, F. Wang, S. Sun, and C. Gu, "Bayesian model fusion: a statistical framework for efficient pre-silicon validation and post-silicon tuning of complex analog and mixed-signal circuits," in *IEEE/ACM ICCAD*, pp. 795–802, Nov 2013.

[15] Y. W. Teh, "Dirichlet process," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds, pp. 280–287, Springer, 2010.

[16] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, no. 2, pp. 209–230, 1973.

[17] C. E. Antoniak, "Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems," *The Annals of Statistics*, vol. 2, no. 6, pp. 1152–1174, 1974.

[18] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, vol. 4, pp. 639–650, 1994.

[19] M. West, "Hyperparameter estimation in Dirichlet process mixture models," Institute of Statistics and Decision Sciences, Duke University, Tech. Rep., 1992.

[20] G. Fishman, *A First Course in Monte Carlo*, Cengage Learning Stamford, 2005.

[21] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[22] M. D. Hoffman and A. Gelman, "The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo," *Journal of Machine Learning Research*, vol. 15, pp. 1593–1623, April 2014.