Rethinking Sparsity in Performance Modeling for Analog and Mixed Circuits using Spike and Slab Models

Mohamed Baker Alawieh[1], Sinead A. Williamson* [2,3], and David Z. Pan[1]

[1]Electrical and Computer Engineering Department, UT Austin
[2] Information, Risk and Operations Management & Statistics and Data Science, UT Austin
[3]Amazon.com Inc.

***Abstract*—As integrated circuit technologies continue to scale, efficient performance modeling becomes indispensable. Recently, several new learning paradigms have been proposed to reduce the computational cost associated with accurate performance modeling. A common attribute among most of these paradigms is the leverage of the sparsity feature to build efficient performance models. In this work, we propose a new perspective to incorporate sparsity in the modeling task by utilizing spike and slab feature selection techniques. Practically, our proposed method uses two different priors on the different model coefficients based on their importance. This is incorporated into a mixture model that can be built using a hierarchical Bayesian framework to select the important features and find the model coefficients. Our numerical experiments demonstrate that the proposed approach can achieve better results compared to traditional sparse modeling techniques while also providing valuable insight about the important features in the model.**

I. Introduction

The continuous drive towards scaling integrated circuits (IC) technologies has been accompanied by a trend of increasing complexity of chip functionalities. With such complex designs, and technologies descending deep in the sub-micron spectrum, the challenges associated with retaining the robustness of state-of-art designs continue to exacerbate [1]. With the aggressive scaling, process variation manifests itself among the most prominent factors limiting the yield of analog and mixed-signal (AMS) circuits [2]. Assessing this variation must form part of the design flow of modern ICs [1]- [2]. Towards this end, performance modeling has been conventionally used to capture this variability through analytical models that can be used in various applications such as yield estimation [3]–[5] and design optimization [6], [7].

With the increased size and complexity of modern ICs, traditional performance modeling frameworks that rely on a large number of simulations to achieve highly-accurate models have become obsolete due to the large simulation cost [7]. In the literature, different performance modeling approaches have been proposed to address this challenge [8]–[13]. To reduce the required number of samples, and hence, the simulation cost, recent performance modeling frameworks have incorporated knowledge about model coefficients based on prior assumption and/or historical data into the modeling framework. For instance, sparse regression exploits the assumption that most coefficients are close to zero to effectively build accurate models [8], [9]. On the other hand, Bayesian model fusion (BMF) takes advantage of an early-stage model to efficiently build a model for a later stage [10]–[12]. Moreover, co-learning BMF proposed in [14] leverages performance side information to further reduce modeling cost.

Recently, focus has shifted from the computationally expensive supervised learning paradigm to the semi-supervised learning paradigm, where a smaller number of simulations is needed to build accurate models [15]–[17]. These approaches use different learning schemes to leverage unlabeled data, which requires no additional simulation cost. This offers a path towards efficiently building performance models when a small number of labeled samples is available.

A common feature of all the aforementioned approaches is leveraging sparsity inherent in the performance modeling problem. Although the number of process variables in modern IC designs is large, only few of these variables are required to estimate the performance, with other variables being either uninformative, or highly correlated with other variables. A reasonable model might therefore be expected to have only few non-zero coefficients. A model that captures this inherent sparsity allows us to avoid overfitting to noisy data, and reduces the computational cost associated with model building. Even the most advanced learning schemes proposed [10], [14]–[17] incorporate sparsity as a corner block for the modeling techniques.

Mathematically, sparsity information can be incorporated by setting an upper bound on the number of non-zero coefficients, amounting to incorporating an L-0 norm into the regression problem. However, such a constraint renders the modeling problem intractable. Several approaches have been proposed to address this challenge. Traditionally, these approaches can be divided into two broad categories: (i) relaxation-based approaches and (ii) heuristics .

Under the relaxation-based approaches, the two most common schemes are Lasso [18] and the ridge regression [10], [19], where *L-1 norm* and *L-2 norm* constraints are used as proxies for the constraint on the number of non-zero coefficients. In theory, this is equivalent to using a Laplace or a Gaussian prior on the model coefficients respectively. These constraints are sometimes referred to as shrinkage constraints, because they favor solutions where the value of the performance model coefficients are close to zero. While

*Contribution to this work was done prior to joining Amazon.

DAC '19, June 2-6, 2019, Las Vegas, NV, USA
©2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6725-7/19/06...$15.00
https://doi.org/10.1145/3316781.3317896

this encourages sparsity—particularly in the L-1 case—it also penalizes model coefficients with high values. This behaviour is not always desired and can in fact affect the quality of the model. Further, these models are not true variable selection techniques, where a subset of important variables are identified and handled differently by the model: a variable could have a small coefficient because it is non-important, or because it is important but has a small numeric value.

Another downside of relaxation-based approaches is that they do not directly capture uncertainty in the model, instead providing a point estimate. Bayesian analogues of the Lasso and ridge regression involve placing a Laplace or a Gaussian prior, respectively, on the model coefficients. This allows us to incorporate prior knowledge, and to infer our uncertainty about the model parameters. However, they still penalize high values and do not perform explicit variable selection.

On the other hand, heuristics methods explicitly perform variable selection, by using an iterative method to choose the important variables to include in the model [8], [9]. As a first step, variables highly correlated with the performance of interest (PoI) are iteratively selected. Then, all coefficients corresponding to non-important variables are set to zero, and least squares fitting can be used to find the coefficients of the few important variables [20]. While such methods clearly identify the important features, these features selection process is heuristic and some information can be lost when setting all other coefficients to zero. In addition, experiments in [15] have shown that these heuristics may exhibit some unstable behaviour when the number of labeled samples is very small.

In this work, we propose using a Bayesian spike and slab feature selection technique to efficiently build accurate performance models. Spike and slab models explicitly partition variables into important and non-important, and then models the values of the important variables independently of the feature selection mechanism. A hierarchical Bayesian framework is utilized to determine both the importance and value of the coefficients simultaneously. At its highest level, the hierarchy dictates that a particular coefficient is sampled from one of two zero-mean prior Gaussian distributions: a low variance distribution centered around zero, referred to as the **spike**, and a large variance distribution, referred to as the **slab**. In our method, a Gibbs sampler is proposed to solve for the posterior distribution of the model coefficients based on the spike and slab mixture model [21], [22]. Unlike optimization-based methods, this allows us to directly capture not just a point estimate of the coefficients, but also our uncertainty about those estimates.

Our proposed method addresses the sparsity in performance modeling in a novel approach that tackles the problem from a new perspective. Unlike relaxation-based approaches, our approach determines the importance of a variable separately from determining its value. Therefore, unlike the case of Lasso and ridge regression, the model coefficients are not penalized for high values if they are determined to by important in the model. In addition, unlike the heuristic methods mentioned earlier, important features are selected jointly and the non-important features are not set to zero, but rather sampled from a low variance prior which can reflect weak correlations that are otherwise missed. Hence, our proposed methods introduces a new perspective for leveraging sparsity that can be incorporated with advanced modeling techniques such as semi-supervised learning. The contributions of this work can be summarized as follows.

- We propose a spike and slab model to efficiently leverage sparsity in performance modeling through a mixture of priors.
- A hierarchical Bayesian model is presented to learn the importance and values of coefficients in a sparse model.
- We introduce a Gibbs sampler to obtain the model coefficients based on the mixture model.
- The experimental results demonstrate superior results compared to conventional sparse modeling approaches.

The remainder of this paper is organized as follows. In Section II we review the technical background and then present the proposed approach in Section III. Section IV presents numerical results demonstrating the efficacy of our method, and conclusions are presented in Section V.

II. Background

A. Performance Modeling

Mathematically, a performance model approximates a circuit-level PoI (e.g. gain, power) as an analytical function of the process variables:

$$y \approx f(\mathbf{p}) = \sum_{k=1}^{K} \beta_k . b_k(\mathbf{p}) \tag{1}$$

where y is the PoI, $\mathbf{p}$ is a vector containing the process variables, $f(\mathbf{p})$ is the modeling function, $\{\beta_k; k = 1, 2, \ldots, K\}$ contains the model coefficients, $\{b_k; k = 1, 2, \ldots, K\}$ contains the basis functions, and M denotes the total number of basis functions.

Given a set of samples, the model coefficients in (1) are usually obtained through least-squares fitting by solving the following optimization problem [22], [20]:

$$\min_{\boldsymbol{\alpha}} ||\mathbf{y} - \mathbf{X} \cdot \boldsymbol{\beta}||_2^2 \tag{2}$$

where $|| \bullet ||_2$ is the L_2-norm of a vector, and

$$\mathbf{X} = \begin{bmatrix} b_1(\mathbf{p}^{(1)}) & b_2(\mathbf{p}^{(1)}) & \ldots & b_K(\mathbf{p}^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ b_1(\mathbf{p}^{(N)}) & b_2(\mathbf{p}^{(N)}) & \ldots & b_K(\mathbf{p}^{(N)}) \end{bmatrix} \tag{3}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_M \end{bmatrix}^T \tag{4}$$

$$\mathbf{y} = \begin{bmatrix} y^{(1)} & y^{(2)} & \ldots & y^{(N)} \end{bmatrix}^T . \tag{5}$$

In (3)-(5), N is the total number of samples, and $\mathbf{p}^{(n)}$ and $y^{(n)}$ are the values of $\mathbf{p}$ and y at the $n-$th sample respectively.

However, least-squares will tend to overfit if the number of coefficients is large relative to the number of samples. Given the high dimensionality of the performance models in complex AMS circuit designs, this means the simulation cost for building accurate models can be exorbitant. Hence, most recent performance modeling techniques incorporate additional information about the model, such as the sparse nature of the coefficients vector, to reduce the number of simulations needed to build accurate models [8]–[13].

B. Sparse Modeling

The large number of variables means that generating enough samples to build highly accurate performance models using least squares regression is often infeasible. However, even though the number of variables is large, in many cases variation in the data actually depends only on a small subset of these variables.

To capture the assumption that only a small subset of variables are relevant, we can constrain on the number of non-zero model coefficients in our model. This encourages solutions where a small set of important variables will have non-zero coefficients while all other non-important ones will have zero coefficients. This can be formulated as an optimization problem with the following objective:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & ||\mathbf{y} - \mathbf{X} \cdot \boldsymbol{\beta}||_2^2 \\ \text{subject to} \quad & ||\boldsymbol{\beta}||_0 \leq \lambda \end{aligned} \tag{6}$$

In (6), $|| \bullet ||_0$ is the "L_0-norm" of a vector. The optimization problems in (6) is NP-hard; hence, several heuristics and relaxations have been proposed to efficiently find the sub-optimal solutions $\boldsymbol{\beta}^*$. For example, replacing the L_0 penalty with an L_2 penalty will encourage coefficients to be smaller, shrinking their values towards zero:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & ||\mathbf{y} - \mathbf{X} \cdot \boldsymbol{\beta}||_2^2 \\ \text{subject to} \quad & ||\boldsymbol{\beta}||_2 \leq \lambda. \end{aligned} \tag{7}$$

This can alternatively be expressed in terms of the Lagrangian,

$$\min_{\boldsymbol{\beta}} \quad ||\mathbf{y} - \mathbf{X} \cdot \boldsymbol{\beta}||_2^2 + c||\boldsymbol{\beta}||_2. \tag{8}$$

The solution to this Lagrangian gives the *maximum a posteriori* (MAP) estimator for a Bayesian model where the prior on the coefficients is a Gaussian with variance proportional to c. This captures our intuition that, a priori, values tend to be small.

If we replace the "L_2-norm" in (7) with the "L_1-norm", we recover the Lasso objective [18], which corresponds to the MAP estimator of a Bayesian model with Laplace priors on the coefficient. This will shrink coefficients more aggressively towards zero, and is a common choice in sparse modeling. While this does encourage sparsity, it comes at the cost of forcing all coefficients to move close to zero by penalizing for high values in $\boldsymbol{\beta}$. In other words, all coefficients, despite being important or not are pushed by the prior towards zero. A practice which despite imposing sparsity, penalizes the values of the coefficients which is not always desirable.

III. Spike and Slab Model for Performance Modeling

In this section, we present the details of the proposed spike and slab approach for performance modeling.

A. Overview

As discussed above, we can construct a Bayesian analogue of a ridge regression framework by placing a Gaussian prior on the model coefficients. The standard deviation of the prior describes the expected range of values for the coefficients: a smaller standard deviation will encourage smaller coefficients. Mathematically, this can be represented as:

$$\begin{aligned} (y_i|\mathbf{x}_i, \boldsymbol{\beta}, \sigma^2) &\sim \text{Normal}(\mathbf{x}_i^T \boldsymbol{\beta}, \sigma^2) \quad \text{for } i = 1, \dots, N \\ (\beta_k|\mu_0, \sigma_0^2) &\sim \text{Normal}(\mu_r, \sigma_r^2) \quad \text{for } k = 1, \dots, K \end{aligned} \tag{9}$$

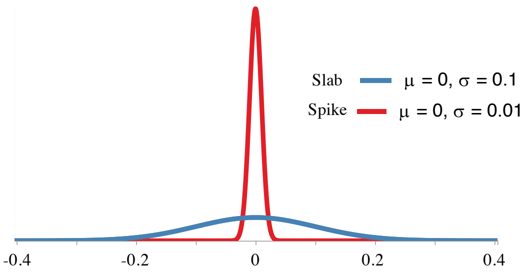

Fig. 1 An example of a Gaussian spike and slab priors mixture is shown. The spike prior is a small variance Gaussian distribution (red) while the slab model has a relatively higher variance (blue).

where $\mathbf{x}_n$ is the $n-$th row of $\mathbf{X}$ and μ_r is typically set to zero. While this formulation can be used to encourage small coefficients, by choosing a small value of σ_r^2, this will encourage *all* coefficients to be small.

Rather than encouraging all coefficients to be small, we want a mechanism that aggressively shrinks non-informative coefficients towards zero, but allows informative coefficients to be large. Starting from the formulation in (9), we propose a new prior model where each model coefficient is associated from one of two components: a spike prior, and a slab prior, as shown in Fig. 1. This is done by introducing a latent variable that selects the distribution from which the posterior of a particular coefficient is sampled. Mathematically, the prior on $\boldsymbol{\beta}$ can be expressed as [21]:

$$(\beta_k|z_k, \tau_k^2) \sim (1 - z_k)\text{Normal}(0, \tau_k^2) + z_k\text{Normal}(0, c_k\tau_k^2) \tag{10}$$

where, $\tau_k > 0$ is a suitably small value, $c_k > 0$ is a suitably large value, and z_k is a binary latent variable. Coefficients with posterior latent variable $z_k = 1$ are those important in the model. Their prior variance is large, which allows the posterior value of β_k to be large. On the other hand, $z_k = 0$ implies that the $k-$th coefficient is not important, and the small prior variance means that the inferred coefficient value β_k will tend to be small. In other words, when $z_k = 1$ the slab prior (i.e., blue curve in Fig. 1) is used, else the spike model (i.e., red curve in Fig. 1) is used.

In this work, we adopt the spike and slab variable selection framework proposed in [21] where a prior hierarchy on β is established and a Gibbs sampler is proposed to solve for the posterior mean of the model coefficients [21], [22]. The details of this framework are presented in the next section.

B. Model Details

In practice the, the prior on β_k in (10) can be represented as $\text{Normal}(0, j_k\tau_k^2)$ where j_k takes the value 1 when $z_k = 1$ and a very small number, $1 >> \nu_0 > 0$ when $z_k = 0$. With the new prior definition on $\boldsymbol{\beta}$, the model can be expressed as follows [21]:

$$\begin{aligned} (y_i|x_i, \boldsymbol{\beta}, \sigma^2) &\sim \text{Normal}(x_i^T \boldsymbol{\beta}, \sigma^2) \\ (\beta_k|z_k, \tau_k^2) &\sim \text{Normal}(0, j_k\tau_k^2). \end{aligned} \tag{11}$$

Next, we define a new latent variable w that represents the probability of a model coefficient being important in the model. In other words, w can be viewed as the ratio of number of important coefficients to the total number of coefficients in the model. In addition, a uniform prior can be set on w. This results in adding the following level of hierarchy to the hierarchical model in (11):

$$\begin{aligned}(j_k|\nu_0, w) \sim& (1-w)\delta_{\nu_0}(\bullet) + w\delta_1(\bullet)\\ w \sim& \text{Uniform}[0,1],\end{aligned} \tag{12}$$

where $\delta_\nu(\bullet)$ is a discrete measure concentrated around ν.

Moreover, variables $\{\tau_k; k = 1, 2, \dots, K\}$ represent the variance of the spike priors for the model coefficients, and σ^2 represents the noise in the model. Hence, Gamma priors can be set on these variables (expressed in terms of precision instead of variance):

$$\begin{aligned}(\tau_2^{-1}|b_1, b_2) \sim& \text{Gamma}(a_1, a_2)\\ \sigma^{-2} \sim& \text{Gamma}(b_1, b_2)\end{aligned} \tag{13}$$

where $a_1, a_2, b_1,$ and b_2 are hyper-parameters.

Equations (11)-(13) form a hierarchical Bayesian mixture representing the spike and slab prior framework. The ultimate goal is to obtain the the posterior distribution of $\boldsymbol{\beta}$, $P(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X})$ while integrating all latent variables and hyper-parameters. However, such distribution does not have a closed form, hence, it is not feasible to directly sample from the the posterior distribution. Instead, we can derive a closed form conditional distribution for $\boldsymbol{\beta}$, and use an iterative Gibbs sampler to sample from the unknown posterior [21], [22]. In principal, Gibbs sampling is a Markov Chain Monte Carlo (MCMC) algorithm for obtaining a group of samples which are approximated from a probability distribution when direct sampling is difficult as in the case of the posterior distribution of $\boldsymbol{\beta}$.

According to the model in (11)-(13), β_k has a Gaussian prior and a Gaussian likelihood. Therefore, conditioned on $\{\tau_k, j_k : k = 1, \dots, K\}$ and σ, the posterior distribution of $\boldsymbol{\beta}$ is Gaussian [21], [22]:

$$\begin{aligned}(\boldsymbol{\beta}|\{\tau_k, j_k : k = 1, \dots, K\}, \sigma) \sim \text{Normal}(\mu, \sigma^2\Sigma)\\ \mu = \Sigma\mathbf{X}^T\mathbf{y} \qquad \Sigma = (\mathbf{X}^T\mathbf{X} + \sigma^2\Lambda^{-1})^{-1}\end{aligned} \tag{14}$$

where Λ is a diagonal matrix with diagonal $\boldsymbol{\gamma}$, and $\{\gamma_k = j_k\tau_k^2 : k = 1, \dots, K\}$.

Next, once $\boldsymbol{\beta}$ is obtained, updated values for $\{j_k : k = 1, \dots, K\}$ can be sampled from the conditional distribution [21]:

$$\begin{aligned}(j_k|\nu_0, w) \sim& \frac{w_{1,k}}{w_{1,k} + w_{2,k}}\delta_{\nu_0}(\bullet) + \frac{w_{2,k}}{w_{1,k} + w_{2,k}}\delta_1(\bullet)\\ &\text{where:}\\ w_{1,k} =& (1-w)\nu_0^{\frac{-1}{2}} \exp(-\frac{\beta_k^2}{2\nu_0\tau_k^2})\\ w_{2,k} =& w \exp(-\frac{\beta_k^2}{2\tau_k^2}).\end{aligned} \tag{15}$$

In (15), the values of $w_{1,k}$ and $w_{2,k}$ represent the unnormalized probabilities that the coefficient β_k comes from the spike and slab priors respectively. To sample j_k, a sample is obtained from a Bernoulli distribution with the normalized values of $w_{1,k}$ and $w_{2,k}$. If the sampled value is zero, j_k is set to ν_0; otherwise, it is set to 1.

Then, τ_k^{-2} can be sampled from its conditional Gamma distribution [21]:

$$(\tau_2^{-1}|\boldsymbol{\beta}, \boldsymbol{J}) \sim \text{Gamma}(a_1 + \frac{1}{2}, a_2 + \frac{\beta_k^2}{2j_k}) \tag{16}$$

Moreover, the complexity parameter w can be sampled from its conditional distribution:

$$(w|\boldsymbol{\gamma}) \sim \text{Beta}(1 + \#\{k : j_k = 1\}, 1 + \#\{k : j_k = \nu_0\}) \tag{17}$$

where $\#\{k : j_k = 1\}$ and $\#\{k : j_k = \nu_0\}$ are the number of coefficients whose corresponding j_k values are equal to 1 and ν_0 respectively. In other words, the two count terms represent the number of important and non-important variables in the model respectively.

Finally, the value of σ representing the model error can be similarly sampled from its conditional distribution:

$$(\sigma^{-2}|\boldsymbol{\beta}, Y) \sim \text{Gamma}(b_1 + \frac{n}{2}, b_2 + \frac{||Y - X\boldsymbol{\beta}||_2^2}{2}). \tag{18}$$

This sampling scheme represents one iteration of the Gibbs sampler used to obtain the final model. At the end of this iteration, updated values are obtained for $\{\gamma_k = j_k\tau_k^2 : k = 1, \dots, K\}$, hence, matrix Λ in (14) can be updated and a new value for $\boldsymbol{\beta}$ can be sampled [21].

This process is done iteratively until the sampling process arrives at the final model solution. The overall sampling algorithm is summarized in the III-C.

C. Gibbs Sampler

Algorithm 1 summarizes the Gibbs sampler used to obtain the optimal model coefficients $\boldsymbol{\beta}^*$ based on the model presented in (11)-(13). As a first step, variables $\{j_k, \tau_k^2 : k = 1, \dots, K\}, w, a_1, a_2, b_1, b_2, \nu_0$ and σ are initialized, and the number of total sampling and burnout iterations, M and B_{out}, are chosen. Here, burnout iterations refer to the early sampling iterations that should be discarded when computing $\boldsymbol{\beta}^*$. Next, iterative sampling is performed while saving the resultant $\boldsymbol{\beta}$ at each iteration. When the sampling is concluded, $\boldsymbol{\beta}^*$ is computed as the average of $\boldsymbol{\beta}$ across all sampling iterations after discarding the first B_{out} iterations [21], [22].

Algorithm 1 Gibbs Sampler for Spike and Slab Model

```
Initialize values for variables $\{j_k, \tau_k^2 : k = 1, \dots, K\}, w, a_1, a_2, b_1, b_2,$ and $\sigma$;
Set the number of sampling iterations $M$ and the burnout value $B_{out}$;
repeat
    Sample $\boldsymbol{\beta}$ from its conditional distribution according to (14);
    Sample $\{j_k : k = 1, \dots, K\}$ from the conditional distribution in (15);
    Sample $\{\tau_k : k = 1, \dots, K\}$ from the conditional distribution in (16);
    Sample $w$ from the conditional distribution in (17);
    Sample $\sigma$ from the conditional distribution in (18);
    Update the values of $\{\gamma_k = j_k\tau_k^2 : k = 1, \dots, K\}$;
    Form the new diagonal matrix $\lambda$ using the updated values of $\{\gamma_k : k = 1, \dots, K\}$;
until number of required sampling iterations is reached.
Compute $\boldsymbol{\beta}^*$ as the average of $\boldsymbol{\beta}$ across all sampling iterations after discarding burnout iterations;
```

Concerning the initialization step, ν_0 can be initialized to a very small number say 0.00005. Also, w can be initially set to a rough estimate of the sparsity level of the model, or it can be simply set to 0.5 implying that the probability of each

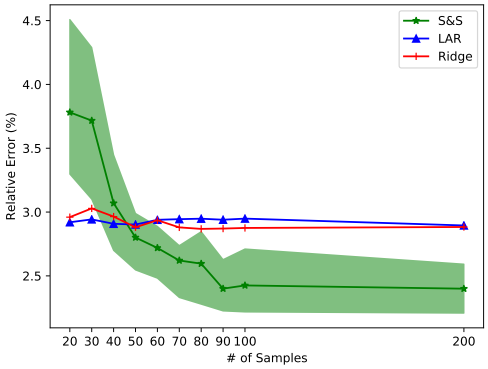

Fig. 2 The modeling error for the three different approaches as a function of the number of samples is shown. For S&S the green line represents the average error over 10 different random runs, while the green shaded region shows the variation across these runs.

TABLE I The initialization scheme for the parameters used in Algorithm 1 is summarized.

Parameter	Value
# of Iterations (M)	2800
Burnout Iterations (B_{out})	700
w	$\frac{1}{3}$
$\{\tau_k : k = 1, \dots, K\}$	1
$\{j_k : k = 1, \dots, K\}$	$\begin{cases} 1 & \text{with Probability } w \\ \nu_0 & \text{with Probability } (1-w) \end{cases}$
a_1, b_1	5
a_2, b_2	50
ν_0	0.0005

coefficient being important is 50%. Once w is set, $\{j_k : k = 1, \dots, K\}$ can be randomly initialized to take the value 1 with probability w and ν_0 with probability $1 - w$. Moreover, the standard deviation values in $\{\tau_k : k = 1, \dots, K\}$ and σ can be initialized to 1 if no further information about the model is available. Finally, the values of the pairs (a_1, a_2) and (b_1, b_2) are set to $(5, 50)$. This choice has been shown to be suitable for the spike and slab Bayesian model in [21].

IV. Experimental Results

In this section, a circuit example implemented using TSMC-40nm technology is used to demonstrate the efficacy of the proposed method. All numerical experiments are performed on a server with 3.4GHz processor and 32GB f memory.

To demonstrate the proposed approach we consider a Strong-ARM latch comparator circuit with power consumption being the performance of interest. In total, 1282 random variables are used to model the process variations for the circuit. The labeled samples are obtained by performing circuit simulations based on Monte Carlo sampling. To show the efficacy of the proposed method, three performance modeling approaches are implemented and compared: (i) least angle regression [9] (LAR),(ii) ridge regression [19] (Ridge) and (iii) the proposed method. For the proposed approach, 10 runs from different random seeds were performed and the average error is computed. In addition, Table I summarizes the initialization scheme used in these runs. For the other two approaches, 10 folds cross-validation was used to get the optimal sparsity parameters. Throughout this section, the error metric used is the relative absolute error (%).

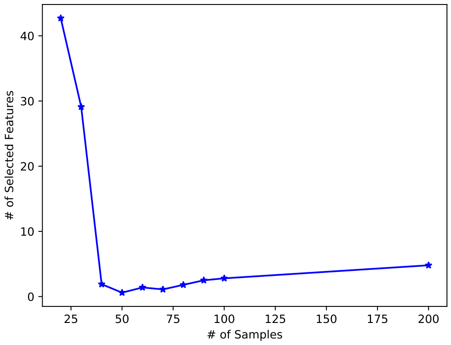

Fig. 3 The average number of selected features as a function of the number of samples is shown. The trend shows that when the model converges, only a small number of features are selected.

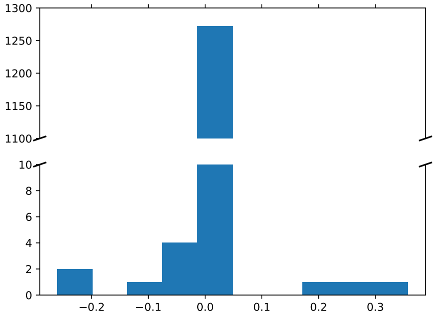

Fig. 4 Histogram showing the values of the coefficients in the final model for S&S with 90 samples.

Fig. 2 shows the modeling error as a function of the number of samples. For the proposed approach the green line represents the average across 10 runs, while the green shaded region shows the variation across the 10 Gibbs sampling runs. As shown in the figure, LAR and Ridge can converge to their final solution faster than the proposed approach. However, our proposed S&S approach can eventually reach a better final solution with a lower modeling error when testing on a separate test data. In fact, when using 90 samples in the training data, S$S can achieve 2.39% modeling error compared to 2.89% and 2.88% for LAR and Ridge respectively. This translates into 17% reduction in modeling error when the same number of simulations is used.

These results are summarized in Table II which shows as well a comparison of runtime for the three approaches. It is important to note that, despite the fact the S&S is more computationally expensive when compared to LAR and Ridge, the overall computational cost is dominated by the simulation cost and the modeling cost is relatively negligible.

In addition, one important feature of the S&S method is that it is a *feature selection* method. In other words, the model can clearly distinguish important features. In practice, the values

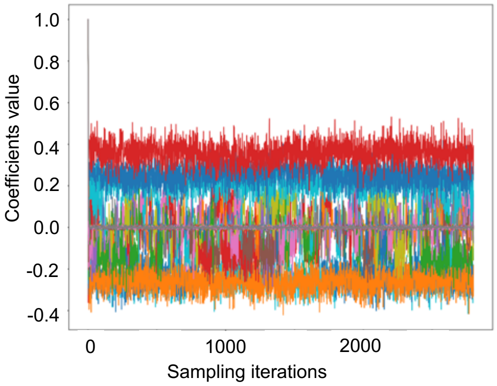

Fig. 5 The sampling process for S&S is shown. Figure shows few non-zero coefficients as the model converges during the last sampling iterations. These represent the selected features.

TABLE II A comparison between the LAR, Ridge and S&S in terms of modeling error and computational cost is summarized.

	LAR	Ridge	S&S
Number of Simulations	90	90	90
Relative Error	2.89%	2.88%	2.39%
Simulation Cost (min.)	2520	2520	2520
Modeling Cost (min.)	0.1	0.1	4

of $\{j_k : k = 1, \dots, K\}$ obtained from Algorithm 1 carry the information about the importance of each feature. Moreover, Fig. 3 shows the trend of the number of selected features as a function of the number of samples. The figure clearly shows that, with a small number of samples, the feature selection is not accurate and a large number of features are selected to fit the model. However, as the number of samples increases, the process of feature selection improves and a small number of features are selected. To further elaborate on this, Fig. 4 shows a histogram of the values of the model coefficient for one run of S&S with 90 samples. As expected, the histogram shows that the vast majority of coefficients are clustered around zero, with only few coefficients with high absolute value. This can be also observed by examining the sampling process for S&S shown in Fig. 5. Although 2800 sampling iterations were performed, the figure shows that similar results can be achieved with only 1000 sampling iterations. After 1000 iterations, the average values of the coefficients does not change significantly. Also, to build the final model, the coefficients are obtained by taking the average of the values across all iterations after the burnout threshold (set to 700 iterations). And it is clear from the figure that the average is converging quite early for most coefficients.

V. Conclusion

In this paper, a new perspective towards incorporating sparsity in performance modeling for analog and mixed circuit using Spike and Slab models is proposed. This approach can be used to incorporate sparsity in different modeling schemes including the recently proposed semi-supervised learning methods. Our proposed approach uses two different priors on the coefficients of the model in a mixture model framework. In practice, the mixture model sets different priors on different coefficients on the model based on their importance and a hierarchical Bayesian framework is utilized to determine both the importance and values of the coefficients simultaneously. To solve for the model coefficients, a Gibbs sampler is proposed to sample from the posterior distribution of these coefficient. The proposed approach demonstrated superior results compared to traditional sparse modeling schemes while also providing a feature selection framework that can easily select important features in the model. Experimental results demonstrated 17% reduction in modeling error compared to traditional sparse modeling approaches.

VI. Acknowledgement

This work is supported in part by NSF under Award No. 1718570.

References

[1] *International Roadmap for Devices and Systems*. Semiconductor Industry Associate, 2016.

[2] X. Li and L. Pileggi, *Statistical Performance Modeling and Optimization*. Now Publishers, 2007.

[3] X. Li, J. L. Le, P. Gopalakrishnan, and L. Pileggi, "Asymptotic probability extraction for non-normal performance distributions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 26, no. 1, pp. 16–37, 2006.

[4] F. Gong, Y. Shi, H. Yu, and L. He, "Variability-aware parametric yield estimation for analog/mixed-signal circuits: Concepts, algorithms, and challenges," *IEEE Design & Test*, vol. 31, no. 4, pp. 6–15, 2014.

[5] H. Zhang, T.-H. Chen, M.-Y. Ting, and X. Li, "Efficient design-specific worst-case corner extraction for integrated circuits," in *ACM/IEEE Design Automation Conference (DAC)*, 2009, pp. 386–389.

[6] Y. Wang, M. Orshansky, and C. Caramanis, "Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization," in *ACM/IEEE Design Automation Conference (DAC)*, 2014, pp. 1–6.

[7] M. B. Alawieh, F. Wang, R. Kang, X. Li, and R. Joshi, "Efficient analog circuit optimization using sparse regression and error margining," in *IEEE International Symposium on Quality Electronic Design (ISQED)*, 2016, pp. 410–415.

[8] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance variability modeling of analog/rf circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 29, no. 11, pp. 1661–1668, 2010.

[9] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance modeling by least angle regression," in *ACM/IEEE Design Automation Conference (DAC)*, 2009, pp. 364–369.

[10] F. Wang, P. Cachecho, W. Zhang, S. Sun, X. Li, R. Kanj, and C. Gu, "Bayesian model fusion: large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 35, no. 8, pp. 1255–1268, 2015.

[11] Q. Huang, F. Fang, Chenlei amd Yang, X. Zeng, and X. Li, "Efficient multivariate moment estimation via bayesian model fusion for analog and mixed-signal circuits," in *ACM/IEEE Design Automation Conference (DAC)*, 2015, pp. 1–6.

[12] C. Fang, Q. Huang, X. Ynag, Fanand Zeng, X. Li, and C. Gu, "Efficient bit error rate estimation for highspeed link by bayesian model fusion," in *IEEE/ACM Proceedings Design, Automation and Test in Eurpoe (DATE)*, 2015, pp. 1024–1029.

[13] Y. Lin, M. B. Alawieh, W. Ye, and D. Pan, "Machine learning for yield learning and optimization," in *IEEE International Test Conference (ITC)*, 2018.

[14] F. Wang, M. Zaheer, X. Li, J.-O. Plouchart, and A. Valdes-Garcia, "Co-learning bayesian model fusion: Efficient performance modeling of analog and mixed-signal circuits using side information," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016, pp. 575–582.

[15] M. B. Alawieh, F. Wang, and X. Li, "Efficient hierarchical performance modeling for analog and mixed-signal circuits via bayesian co-learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, pp. 1–13, 2018.

[16] M. B. Alawieh, F. Wang, and X. Li, "Efficient hierarchical performance modeling for integrated circuits via bayesian co-learning," in *ACM/IEEE Design Automation Conference (DAC)*, 2017, pp. 1–6.

[17] M. B. Alawieh, X. Tang, and D. Pan, "S^{2}PM: semi-supervised learning for efficient performance modeling of analog and mixed signal circuit," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2019.

[18] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society: Series B*, vol. 58, pp. 267–288, 1996.

[19] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," vol. 12, no. 1, pp. 55–67, 1970.

[20] S. boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[21] H. Ishwaran and J. Sunil Rao, "Spike and slab variable selection: Frequentist and bayesian strategies," *The Annals of Statistics*, vol. 33, no. 2, pp. 730–773, 2005.

[22] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.