Softplus Regressions and Convex Polytopes

Mingyuan Zhou*

The University of Texas at Austin, Austin, TX 78712

August 23, 2016

Abstract

To construct flexible nonlinear predictive distributions, the paper introduces a family of softplus function based regression models that convolve, stack, or combine both operations by convolving countably infinite stacked gamma distributions, whose scales depend on the covariates. Generalizing logistic regression that uses a single hyperplane to partition the covariate space into two halves, softplus regressions employ multiple hyperplanes to construct a confined space, related to a single convex polytope defined by the intersection of multiple half-spaces or a union of multiple convex polytopes, to separate one class from the other. The gamma process is introduced to support the convolution of countably infinite (stacked) covariate-dependent gamma distributions. For Bayesian inference, Gibbs sampling derived via novel data augmentation and marginalization techniques is used to deconvolve and/or demix the highly complex nonlinear predictive distribution. Example results demonstrate that softplus regressions provide flexible nonlinear decision boundaries, achieving classification accuracies comparable to that of kernel support vector machine while requiring significant less computation for out-of-sample prediction.

Keywords: Bernoulli-Poisson link; convolution and stack of experts; deep learning; gamma belief network; nonlinear classification; softplus and rectifier functions

*M. Zhou is an assistant professor of statistics in the Department of Information, Risk, & Operations Management and Department of Statistics & Data Sciences at the University of Texas at Austin, Austin, TX 78712, USA. *Email:* `mingyuan.zhou@mccombs.utexas.edu`

1 Introduction

Logistic and probit regressions that use a single hyperplane to partition the covariate space into two halves are widely used to model binary response variables given the covariates [Albert and Chib, 1993, Cox and Snell, 1989, Holmes and Held, 2006, McCullagh and Nelder, 1989]. They are easy to implement and simple to interpret, but neither of them is capable of producing nonlinear classification decision boundaries, and they may not provide large margin to achieve accurate out-of-sample predictions. For two classes not well separated by a single hyperplane, rather than regressing a binary response variable directly on its covariates, it is common to select a subset of covariate vectors as support vectors, choose a nonlinear kernel function, and regress a binary response variable on the kernel distances between its covariate vector and these support vectors [Boser et al., 1992, Cortes and Vapnik, 1995, Schölkopf et al., 1999, Tipping, 2001, Vapnik, 1998]. Alternatively, one may construct a deep neural network to nonlinearly transform the covariates in a supervised manner, and then regress a binary response variable on its transformed covariates [Bengio et al., 2015, Hinton et al., 2006, LeCun et al., 2015].

Both kernel learning and deep learning map the original covariates into a more linearly separable space, transforming a nonlinear classification problem into a linear one. In this paper, we propose a fundamentally different approach for nonlinear classification. Relying on neither the kernel trick nor a deep neural network to transform the covariate space, we construct a family of softplus regressions that exploit two distinct types of interactions between hyperplanes to define flexible nonlinear classification decision boundaries directly on the original covariate space. Since kernel learning based methods such as kernel support vector machines (SVMs) [Cortes and Vapnik, 1995, Vapnik, 1998] may scale poorly in that the number of support vectors often increases linearly in the size of the training dataset, they could be not only slow and memory inefficient to train but also unappealing for making fast out-of-sample predictions [Steinwart, 2003, Wang et al., 2011]. One motivation of the paper is to investigate the potential of using a set of hyperplanes, whose number is directly influenced by how the interactions of multiple hyperplanes can be used to spatially separate two different classes in the covariate space rather than by the training data size, to construct nonlinear classifiers that can match the out-of-sample prediction accuracies of kernel SVMs, but potentially with much lower computational complexity. Another motivation of the paper is to increase the margin of the classifier, related to the discussion in Kantchelian et al. [2014] that for two classes that are linearly separable, even though a single hyperplane is sufficient to separate the two different classes in the training dataset, using multiple hyperplanes to enclose one class may help clearly increase the total margin of the classifier and hence improve the out-of-sample prediction accuracies.

Our motivated construction exploits two distinct operations—convolution and stacking—on the gamma distributions with covariate-dependent scale parameters. The convolution operation convolves differently parameterized probability distributions to increase representation power and enhance smoothness, while the stacking operation mixes a distribution in the stack with a distribution of the same family that is subsequently pushed into the stack. Depending on whether and how the convolution and stacking operations are used, the models in the family differ from each other on how they use the softplus functions to construct highly nonlinear probability density functions, and on how they construct their hierarchical Bayesian models to arrive at these functions. In comparison to the nonlinear classifiers built on kernels or deep neural networks, the proposed softplus regressions all share a distinct advantage in providing interpretable geometric constraints, which are related to either a single or a union of convex polytopes [Grünbaum, 2013], on the classification decision boundaries defined on the original covariate space. In addition, like neither kernel learning, whose number of support vectors often increases linearly in the size of data [Steinwart, 2003], nor deep learning, which often requires carefully tuning both the structure of the deep network and the learning algorithm [Bengio et al., 2015], the proposed nonparametric Bayesian softplus regressions naturally provide probability estimates, automatically learn the complexity of the predictive distribution, and quantify model uncertainties with posterior samples.

The remainder of the paper is organized as follows. In Section 2, we define four different softplus regressions, present their underlying hierarchical models, and describe their distinct geometric constraints on how the covariate space is partitioned. In Section 3, we discuss Gibbs sampling via data augmentation and marginalization. In Section 4, we present experimental results on eight benchmark datasets for binary classification, making comparisons with five different classification algorithms. We conclude the paper in Section 5. We defer to the Supplementary Materials all the proofs, an accurate approximate sampler and some new properties for the Polya-Gamma distribution, the discussions on related work, and some additional example results.

2 Hierarchical Models and Geometric Constraints

2.1 Bernoulli-Poisson link and softplus function

To model a binary random variable, it is common to link it to a real-valued latent Gaussian random variable using either the logistic or probit links. Rather than following the convention, in this paper, we consider the Bernoulli-Poisson (BerPo) link [Dunson and Herring, 2005, Zhou, 2015] to threshold a latent count at one to obtain a binary outcome $y \in \{0, 1\}$ as

$$y = \delta(m \geq 1),\ m \sim \text{Pois}(\lambda), \tag{1}$$

where $m \in \mathbb{Z}$, $\mathbb{Z} := \{0, 1, \ldots\}$, and $\delta(x) = 1$ if the condition x is satisfied and $\delta(x) = 0$ otherwise. The marginalization of the latent count m from the BerPo link leads to

$$y \sim \text{Bernoulli}(p),\ p = 1 - e^{-\lambda}.$$

The conditional distribution of m given y and λ can be efficiently simulated using a rejection sampler [Zhou, 2015]. Since its use in Zhou [2015] to factorize the adjacency matrix of an undirected unweighted symmetric network, the BerPo link has been further extended for big binary tensor factorization [Hu et al., 2015], multi-label learning [Rai et al., 2015], and deep Poisson factor analysis [Henao et al., 2015]. This link has also been used by Caron and Fox [2015] and Todeschini and Caron [2016] for network analysis.

We now refer to $\lambda = -\ln(1-p)$, the negative logarithm of the Bernoulli failure probability, as the BerPo rate for y and simply denote (1) as $y \sim \text{BerPo}(\lambda)$. It is instructive to notice that $1/(1 + e^{-x}) = 1 - \exp[-\ln(1 + e^{x})]$, and hence letting

$$y \sim \text{Bernoulli}\big[\sigma(x)\big],\ \sigma(x) = 1/(1 + e^{-x}) \tag{2}$$

is equivalent to letting

$$y \sim \text{BerPo}\big[\varsigma(x)\big],\ \varsigma(x) = \ln(1 + e^{x}), \tag{3}$$

where $\varsigma(x) = \ln(1 + e^{x})$ was referred to as the softplus function in Dugas et al. [2001]. It is interesting that the BerPo link appears to be naturally paired with the softplus function, which is often considered as a smoothed version of the rectifier, or rectified linear unit

$$\text{ReLU}(x) = \max(0, x),$$

that is now widely used in deep neural networks, replacing other canonical nonlinear activation functions such as the sigmoid and hyperbolic tangent functions [Glorot et al., 2011, Krizhevsky et al., 2012, LeCun et al., 2015, Nair and Hinton, 2010, Shang et al., 2016].

In this paper, we further introduce the stack-softplus function

$$\varsigma(x_1, \ldots, x_t) = \ln\left(1 + e^{x_t} \ln\left\{1 + e^{x_{t-1}} \ln\left[1 + \ldots \ln\left(1 + e^{x_1}\right)\right]\right\}\right), \tag{4}$$

which can be recursively defined with $\varsigma(x_1, \ldots, x_t) = \ln[1 + e^{x_t}\varsigma(x_1, \ldots, x_{t-1})]$. In addition, with r_k as the weights of the countably infinite atoms of a gamma process [Ferguson, 1973], we will introduce the sum-softplus function, expressed as $\sum_{k=1}^{\infty} r_k \, \varsigma(x_k)$, and sum-stack-softplus (SS-softplus) function, expressed as $\sum_{k=1}^{\infty} r_k \, \varsigma(x_{k1}, \ldots, x_{kt})$. The stack-, sum-, and SS-softplus functions constitute a family of softplus functions, which are used to construct nonlinear regression models, as presented below.

2.2 The softplus regression family

The equivalence between (2) and (3), the apparent partnership between the BerPo link and softplus function, and the convenience of employing multiple regression coefficient vectors to parameterize the BerPo rate, which is constrained to be nonnegative rather than between zero and one, motivate us to consider using the BerPo link together with softplus function to model binary response variables given the covariates. We first show how a classification model under the BerPo link reduces to logistic regression that uses a single hyperplane to partition the covariate space into two halves. We then generalize it to two distinct multi-hyperplane classification models: sum- and stack-softplus regressions, and further show how to integrate them into SS-softplus regression. These models clearly differ from each other on how the BerPo rates are parameterized with the softplus functions, leading to decision boundaries under distinct geometric constraints.

To be more specific, for the ith covariate vector $\boldsymbol{x}_i = (1, x_{i1}, \ldots, x_{iV})' \in \mathbb{R}^{V+1}$, where the prime denotes the operation of transposing a vector, we model its binary class label using

$$y_i \,|\, \boldsymbol{x}_i \sim \text{BerPo}[\lambda(\boldsymbol{x}_i)], \tag{5}$$

where $\lambda(\boldsymbol{x}_i)$, given the regression model parameters that may come from a stochastic process, is a nonnegative deterministic function of $\boldsymbol{x}_i$ that may contain a countably infinite number of parameters. Let $G \sim \Gamma\text{P}(G_0, 1/c)$ denote a gamma process [Ferguson, 1973] defined on the product space $\mathbb{R}_+ \times \Omega$, where $\mathbb{R}_+ = \{x : x > 0\}$, c is a scale parameter, and G_0 is a finite and continuous base measure defined on a complete separable metric space Ω, such that $G(A_i) \sim \text{Gamma}(G_0(A_i), 1/c)$ are independent gamma random variables for disjoint Borel sets A_i of Ω. Below we show how the BerPo rate function $\lambda(\boldsymbol{x}_i)$ is parameterized under four different softplus regressions, two of which use the gamma process to support a countably infinite sum in the parameterization, and also show how to arrive at each parameterization using a hierarchical Bayesian model built on the BerPo link together with the convolved and/or stacked gamma distributions.

Definition 1 (Softplus regression)**.** *Given $\boldsymbol{x}_i$, weight $r \in \mathbb{R}_+$, and a regression coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^{V+1}$, softplus regression parameterizes $\lambda(\boldsymbol{x}_i)$ in* (5) *using a softplus function as*

$$\lambda(\boldsymbol{x}_i) = \varsigma(\boldsymbol{x}_i'\boldsymbol{\beta}) = r\ln(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}}). \tag{6}$$

Softplus regression is equivalent to the binary regression model

$$y_i \sim \text{Bernoulli}\left[1 - \left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}}\right)^{-r}\right],$$

which, as proved in Appendix B, can be constructed using the hierarchical model

$$y_i = \delta(m_i \geq 1),\ m_i \sim \text{Pois}(\theta_i),\ \theta_i \sim \text{Gamma}\big(r, e^{\boldsymbol{x}_i'\boldsymbol{\beta}}\big). \tag{7}$$

Definition 2 (Sum-softplus regression)**.** *Given a draw from a gamma process $G \sim \Gamma\text{P}(G_0, 1/c)$, expressed as $G = \sum_{k=1}^{\infty} r_k \delta_{\boldsymbol{\beta}_k}$, where $\boldsymbol{\beta}_k \in \mathbb{R}^{V+1}$ is an atom and r_k is its weight, sum-softplus regression parameterizes $\lambda(\boldsymbol{x}_i)$ in* (5) *using a sum-softplus function as*

$$\lambda(\boldsymbol{x}_i) = \sum_{k=1}^{\infty} r_k\, \varsigma(\boldsymbol{x}_i'\boldsymbol{\beta}_k) = \sum_{k=1}^{\infty} r_k \ln(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k}). \tag{8}$$

Sum-softplus regression is equivalent to the binary regression model

$$y_i \sim \text{Bernoulli}\left[1 - e^{-\sum_{k=1}^{\infty} r_k \varsigma(\boldsymbol{x}_i'\boldsymbol{\beta}_k)}\right] = \text{Bernoulli}\left[1 - \prod_{k=1}^{\infty}\left(\frac{1}{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k}}\right)^{r_k}\right],$$

which, as proved in Appendix B, can be constructed using the hierarchical model

$$y_i = \delta(m_i \geq 1),\ m_i \sim \text{Pois}(\theta_i),\ \theta_i = \sum_{k=1}^{\infty} \theta_{ik},\ \theta_{ik} \sim \text{Gamma}\big(r_k, e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k}\big). \tag{9}$$

Definition 3 (Stack-softplus regression)**.** *With weight $r \in \mathbb{R}_+$ and T regression coefficient vectors $\boldsymbol{\beta}^{(2:T+1)} := (\boldsymbol{\beta}^{(2)}, \ldots, \boldsymbol{\beta}^{(T+1)})$, where $\boldsymbol{\beta}^{(t)} \in \mathbb{R}^{V+1}$, stack-softplus regression with T layers parameterizes $\lambda(\boldsymbol{x}_i)$ in* (5) *using a stack-softplus function as*

$$\begin{aligned}\lambda(\boldsymbol{x}_i) &= r\,\varsigma\big(\boldsymbol{x}_i'\boldsymbol{\beta}^{(2)}, \ldots, \boldsymbol{x}_i'\boldsymbol{\beta}^{(T+1)}\big) \\ &= r\ln\left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(T+1)}} \ln\left\{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(T)}} \ln\left[1 + \ldots \ln\big(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(2)}}\big)\right]\right\}\right).\end{aligned} \tag{10}$$

Stack-softplus regression is equivalent to the regression model

$$\begin{aligned} y_i &\sim \text{Bernoulli}\Big(1 - e^{-r\,\varsigma\left(\boldsymbol{x}_i'\boldsymbol{\beta}^{(2:T+1)}\right)}\Big) \\ &= \text{Bernoulli}\left[1 - \left(1{+}e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(T+1)}}\ln\left\{1{+}e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(T)}}\ln\left[1{+}\ldots\ln\left(1{+}e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(2)}}\right)\right]\right\}\right)^{-r}\ \right], \end{aligned}$$

which, as proved in Appendix B, can be constructed using the hierarchical model that stacks T gamma distributions, whose scales are differently parameterized by the covariates, as

$$\begin{gathered} \theta_i^{(T)} \sim \text{Gamma}\left(r, e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(T+1)}}\right), \\ \cdots \\ \theta_i^{(t)} \sim \text{Gamma}\left(\theta_i^{(t+1)}, e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(t+1)}}\right), \\ \cdots \\ y_i = \delta(m_i \geq 1),\ m_i \sim \text{Pois}(\theta_i^{(1)}),\ \theta_i^{(1)} \sim \text{Gamma}\left(\theta_i^{(2)}, e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(2)}}\right). \end{gathered} \tag{11}$$

Definition 4 (Sum-stack-softplus (SS-softplus) regression)**.** *Given a drawn from a gamma process $G \sim \Gamma\text{P}(G_0, 1/c)$, expressed as $G = \sum_{k=1}^{\infty} r_k \delta_{\boldsymbol{\beta}_k^{(2:T+1)}}$, where $\boldsymbol{\beta}_k^{(2:T+1)}$ is an atom and r_k is its weight, with each $\boldsymbol{\beta}_k^{(t)} \in \mathbb{R}^{V+1}$, SS-softplus regression with $T \in \{1, 2, \ldots\}$ layers parameterizes $\lambda(\boldsymbol{x}_i)$ in* (5) *using a sum-stack-softplus function as*

$$\begin{aligned} \lambda(\boldsymbol{x}_i) &= \sum_{k=1}^{\infty} r_k\,\varsigma\big(\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(2)}, \ldots, \boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T+1)}\big) \\ &= \sum_{k=1}^{\infty} r_k \ln\left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T+1)}}\ln\left\{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T)}}\ln\left[1 + \ldots\ln\left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(2)}}\right)\right]\right\}\right). \end{aligned} \tag{12}$$

SS-softplus regression is equivalent to the regression model

$$\begin{aligned} y_i &\sim \text{Bernoulli}\Big(1 - e^{-\sum_{k=1}^{\infty} r_k\,\varsigma\left(\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(2:T+1)}\right)}\Big) \\ &= \text{Bernoulli}\left[1 - \prod_{k=1}^{\infty}\left(1{+}e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T+1)}}\ln\left\{1{+}e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T)}}\ln\left[1{+}\ldots\ln\left(1{+}e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(2)}}\right)\right]\right\}\right)^{-r_k}\ \right] \end{aligned}$$

which, as proved in Appendix B, can be constructed from the hierarchical model that convolves countably infinite stacked gamma distributions that have covariate-dependent scale *parameters as*

$$\theta_{ik}^{(T)} \sim \text{Gamma}\left(r_k, e^{\boldsymbol{x}'_i \boldsymbol{\beta}_k^{(T+1)}}\right),$$
$$\cdots$$
$$\theta_{ik}^{(t)} \sim \text{Gamma}\left(\theta_{ik}^{(t+1)}, e^{\boldsymbol{x}'_i \boldsymbol{\beta}_k^{(t+1)}}\right),$$
$$\cdots$$
$$y_i = \delta(m_i \geq 1),\ m_i = \sum_{k=1}^{\infty} m_{ik}^{(1)},\ m_{ik}^{(1)} \sim \text{Pois}(\theta_{ik}^{(1)}),\ \theta_{ik}^{(1)} \sim \text{Gamma}\left(\theta_{ik}^{(2)}, e^{\boldsymbol{x}'_j \boldsymbol{\beta}_k^{(2)}}\right) . \tag{13}$$

Below we discuss these four different softplus regression models in detail and show that both sum- and stack-softplus regressions use the interactions of multiple regression coefficient vectors through the softplus functions to define a confined space, related to a convex polytope [Grünbaum, 2013] defined by the intersection of multiple half-spaces, to separate one class from the other in the covariate space. They differ from each other in that sum-softplus regression infers a convex-polytope-bounded confined space to enclose negative examples (*i.e.*, data samples with $y_i = 0$), whereas stack-softplus regression infers a convex-polytope-like confined space to enclose positive examples (*i.e.*, data samples with $y_i = 1$).

The opposite behaviors of sum- and stack-softplus regressions motivate us to unite them as SS-softplus regression, which can place countably infinite convex-polytope-like confined spaces, inside and outside each of which favor positive and negative examples, respectively, at various regions of the covariate space, and use the union of these confined spaces to construct a flexible nonlinear classification decision boundary. Note that softplus regressions all operate on the original covariate space. It is possible to apply them to regress binary response variables on the covariates that have already been nonlinearly transformed with the kernel trick or a deep neural network, which may combine the advantages of these distinct methods to achieve an overall improved classification performance. We leave the integration of softplus regressions with the kernel trick or deep neural networks for future study.

2.3 Softplus and logistic regressions

It is straightforward to show that softplus regression with $r = 1$ is equivalent to logistic regression $y_i \sim \text{Bernoulli}[1/(1+e^{-\boldsymbol{x}'_i \boldsymbol{\beta}})]$, which uses a single hyperplane dividing the covariate space into two halves to separate one class from the other. Similar connection has also been illustrated in Dunson and Herring [2005]. Clearly, softplus regression arising from (7) generalizes logistic regression in allowing $r \neq 1$. Let $p_0 \in (0, 1)$ denote the probability threshold to make a binary decision. One may consider that softplus regression defines a hyperplane to partition the V dimensional covariate space into two halves: one half is defined with $\boldsymbol{x}_i'\boldsymbol{\beta} > \ln\left[(1-p_0)^{-\frac{1}{r}}-1\right]$, assigned with label $y_i = 1$ since $P(y_i = 1 \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}_k) > p_0$ under this condition, and the other half is defined with $\boldsymbol{x}_i'\boldsymbol{\beta} \le \ln\left[(1-p_0)^{-\frac{1}{r}}-1\right]$, assigned with label $y_i = 0$ since $P(y_i = 1 \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}) \le p_0$ under this condition. Instead of using a single hyperplane, the three generalizations in Definitions 2-4 all partition the covariate space using a confined space that is related to a single convex polytope or the union of multiple convex polytopes, as described below.

2.4 Sum-softplus regression and convolved NB regressions

Note that since $m_i \sim \mathrm{Pois}(\theta_i),\ \theta_i = \sum_{k=1}^{\infty}\theta_{ik}$ in (9) can be equivalently written as $m_i = \sum_{k=1}^{\infty} m_{ik},\ m_{ik} \sim \mathrm{Pois}(\theta_{ik})$, sum-softplus regression can also be constructed with

$$y_i = \delta(m_i \ge 1), \quad m_i = \sum_{k=1}^{\infty} m_{ik},\ m_{ik} \sim \mathrm{NB}\left[r_k, 1/(1+e^{-\boldsymbol{x}_i'\boldsymbol{\beta}_k})\right], \tag{14}$$

where $m \sim \mathrm{NB}(r,p)$ represents a negative binomial (NB) distribution [Fisher et al., 1943, Greenwood and Yule, 1920] with shape parameter r and probability parameter p, and $m \sim \mathrm{NB}[r, 1/(1+e^{-\boldsymbol{x}'\boldsymbol{\beta}})]$ can be considered as NB regression [Cameron and Trivedi, 1998, Lawless, 1987, Long, 1997, Winkelmann, 2008] that parameterizes the logit of p with $\boldsymbol{x}'\boldsymbol{\beta}$. To ensure that the infinite model is well defined, we provide the following proposition and present the proof in Appendix B.

Proposition 1. *The infinite product* $e^{-\sum_{k=1}^{\infty} r_k \varsigma(\boldsymbol{x}_i'\boldsymbol{\beta}_k)} = \prod_{k=1}^{\infty}\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k}\right)^{-r_k}$ *in sum-softplus regression is smaller than one and has a finite expectation that is greater than zero.*

As the probability distribution of the sum of independent random variables is the same as the convolution of these random variables' probability distributions [*e.g.*, Fristedt and Gray, 1997], the probability distribution of the BerPo rate θ_i is the convolution of countably infinite gamma distributions, each of which parameterizes the logarithm of its scale using the inner product of the same covariate vector and a regression coefficient vector specific for each k. As in (14), since m_i is the summation of countably infinite latent counts m_{ik}, each of which is a NB regression response variable, we essentially regress the latent count m_i on $\boldsymbol{x}_i$ using a convolution of countably infinite NB regression models. If $\boldsymbol{\beta}_k$ are drawn from a continuous distribution, then $\boldsymbol{\beta}_k \ne \boldsymbol{\beta}_{\tilde{k}}$ a.s. for all $k \ne \tilde{k}$, and hence given $\boldsymbol{x}_i$ and $\{\boldsymbol{\beta}_k\}_k$, the BerPo rate θ_i would not follow the gamma distribution and m_i would not follow the NB distribution.

Note that if we modify the proposed sum-softplus regression model in (9) as

$$y_{ij} = \delta(m_{ij} \geq 1),\ m_{ij} \sim \text{Pois}(\theta_{ij}),\ \theta_{ij} = \sum_{k=1}^{K} \theta_{ijk},\ \theta_{ijk} \sim \text{Gamma}\big(\phi_k^{-1}, \phi_k \lambda_{jk} e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}}\big), \tag{15}$$

then we have $P(y_{ij} = 1 \,|\, \boldsymbol{x}_{ij}) = \left[1 - \prod_{k=1}^{K} \left(1 + \phi_k \lambda_{jk} e^{\boldsymbol{x}'_{ij}\boldsymbol{\beta}}\right)^{-\phi_k^{-1}}\right]$, which becomes the same as Eq. 2.7 of Dunson and Herring [2005] that is designed to model multivariate binary response variables. Though related, that construction is clearly different from the proposed sum-softplus regression in that it uses only a single regression coefficient vector $\boldsymbol{\beta}$ and does not support $K \to \infty$. It is of interest to extend the models in Dunson and Herring [2005] with the sum-softplus construction discussed above and the stack- and SS-softplus constructions to be discussed below.

2.4.1 Convex-polytope-bounded confined space that favors negative examples

For sum-softplus regression arising from (9), the binary classification decision boundary is no longer defined by a single hyperplane. Let us make the analogy that each $\boldsymbol{\beta}_k$ is an expert of a committee that collectively make binary decisions. For expert k, the magnitude of r_k indicates how strongly its opinion is weighted by the committee, $m_{ik} = 0$ represents that it votes "No," and $m_{ik} \geq 1$ represents that it votes "Yes." Since the response variable $y_i = \delta\left(\sum_{k=1}^{\infty} m_{ik} \geq 1\right)$, the committee would vote "No" if and only if all its experts vote "No" (*i.e.*, all the counts m_{ik} are zeros), in other words, the committee would vote "Yes" even if only a single expert votes "Yes."

Let us now examine the confined covariate space for sum-softplus regression that satisfies the inequality $P(y_i = 1 \,|\, \boldsymbol{x}_i) = 1 - e^{-\lambda(\boldsymbol{x}_i)} \leq p_0$, where a data point is labeled as one with a probability no greater than p_0. Although it is not immediately clear what kind of geometric constraints are imposed on the covariate space by this inequality, the following theorem shows that it defines a confined space, which is bounded by a convex polytope defined by the intersection of countably infinite half-spaces.

Theorem 2. *For sum-softplus regression, the confined space specified by the inequality* $P(y_i = 1 \,|\, \boldsymbol{x}_i) = 1 - e^{-\lambda(\boldsymbol{x}_i)} \leq p_0$*, which can be expressed as*

$$\lambda(\boldsymbol{x}_i) = \sum_{k=1}^{\infty} r_k \ln(1 + e^{\boldsymbol{x}'_i\boldsymbol{\beta}_k}) \leq -\ln(1 - p_0), \tag{16}$$

is bounded by a convex polytope defined by the set of solutions to countably infinite inequalities

$$\boldsymbol{x}_i'\boldsymbol{\beta}_k \leq \ln\left[(1-p_0)^{-\frac{1}{r_k}}-1\right], \quad k \in \{1,2,\ldots\}. \tag{17}$$

Proposition 3. *For any data point $\boldsymbol{x}_i$ that resides outside the convex polytope defined by* (17)*, which means $\boldsymbol{x}_i$ violates at least one of the inequalities in* (17) *a.s., it will be labeled under sum-softplus regression with $y_i = 1$ with a probability greater than p_0, and $y_i = 0$ with a probability no greater than $1 - p_0$.*

The convex polytope defined in (17) is enclosed by the intersection of countably infinite V-dimensional half-spaces. If we set $p_0 = 0.5$ as the probability threshold to make binary decisions, then the convex polytope assigns a label of $y_i = 0$ to an $\boldsymbol{x}_i$ inside the convex polytope (*i.e.*, an $\boldsymbol{x}_i$ that satisfies all the inequalities in Eq. 17) with a relatively high probability, and assigns a label of $y_i = 1$ to an $\boldsymbol{x}_i$ outside the convex polytope (*i.e.*, an $\boldsymbol{x}_i$ that violates at least one of the inequalities in Eq. 17) with a probability of at least 50%.

Note that as $r_k \to 0$, $r_k \ln(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k}) \to 0$ and $\ln\left[(1-p_0)^{-\frac{1}{r_k}}-1\right] \to \infty$. Thus expert k with a tiny r_k essentially has a negligible impact on both the decision of the committee and the boundary of the convex polytope. Choosing the gamma process as the nonparametric Bayesian prior sidesteps the need to tune the number of experts in the committee, shrinking the weights of all unnecessary experts and hence allowing a finite number of experts with non-negligible weights to be automatically inferred from the data.

2.4.2 Illustration for sum-softplus regression

A clear advantage of sum-softplus regression over both softplus and logistic regressions is that it could use multiple hyperplanes to construct a nonlinear decision boundary and, similar to the convex polytope machine of Kantchelian et al. [2014], to separate two different classes by a large margin. To illustrate the imposed geometric constraints, we first consider a synthetic two dimensional dataset with two classes, as shown in Fig. 1 (a), where most of the data points of Class B reside within a unit circle and these of Class A reside within a ring outside the unit circle.

We first label the data points of Class A as "1" and these of Class B as "0." Shown in Fig. 1 (b) are the inferred weights r_k of the experts, using the MCMC sample that has the highest log-likelihood in fitting the training data labels. It is evident from Figs. 1 (b) and (c) that sum-softplus regression infers four experts (hyperplanes) with significant weights. The convex polytope in Fig. 1 (c) that encloses the space marked as zero is intersected by these four hyperplanes, each of which is defined as in (17) with $p_0 = 50\%$. Thus outside the convex polytope are data points that would be labeled as "1" with at least 50% probabilities and inside it are data points that would be labeled as "0" with relatively high probabilities. We further show in Fig. 1 (d) the contour map of the inferred probabilities

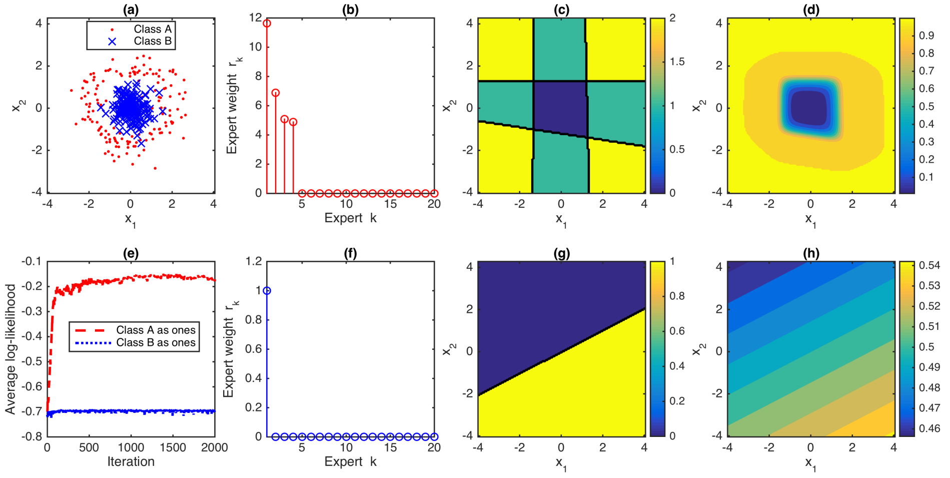

Figure 1: Visualization of sum-softplus regression with $K_{\max} = 20$ experts on a binary classification problem under two opposite labeling settings. For each labeling setting, 2000 Gibbs sampling iterations are used and the MCMC sample that provides the maximum likelihood on fitting the training data labels is used to display the results. (a) A two dimensional dataset that consists of 150 data points from Class A, whose radiuses are drawn from $\mathcal{N}(2, 0.5^2)$ and angles are distributed uniformly at random between 0 and 360 degrees, and another 150 data points from Class B, whose both x-axis and y-axis values are drawn from $\mathcal{N}(0, 0.5^2)$. With data points in Classes A and B labeled as "1" and "0," respectively, and with $p_0 = 0.5$, (b) shows the inferred weights r_k of the experts, ordered by their values, (c) shows a contour map, the value of each point of which represents how many inequalities specified in (17) are violated, and whose region with zero values corresponds to the convex polytope enclosed by the intersection of the hyperplanes defined in (17), and (d) shows the the contour map of the predicted class probabilities. (f)-(h) are analogous plots to (b)-(d), with the data points in Classes A and B relabeled as "0" and "1," respectively. (e) The average per data point log-likelihood as a function of MCMC iteration, for both labeling settings.

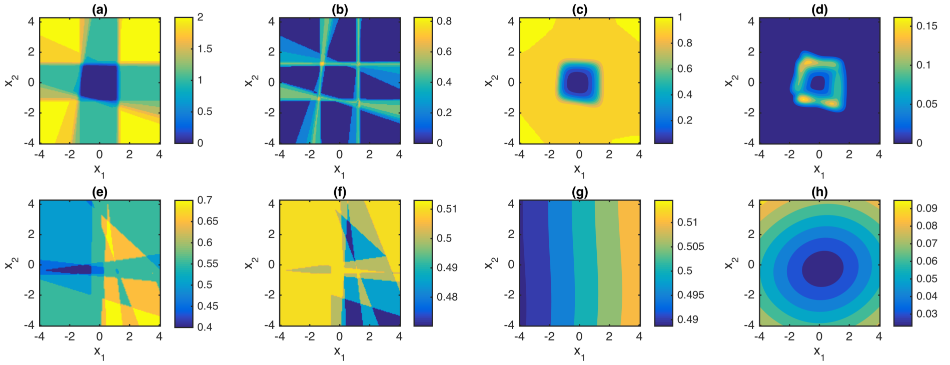

Figure 2: Visualization of the posteriors of sum-softplus regression based on 20 MCMC samples, collected once per every 50 iterations during the last 1000 MCMC iterations, with the same experimental setting used for Fig. 1. With $p_0 = 0.5$, (a) and (b) show the contour maps of the posterior means and standard deviations, respectively, of the number of inequalities specified in (17) that are violated, and (c) and (d) show the contour maps of the posterior means and standard deviations, respectively, of predicted class probabilities. (e)-(h) are analogous plots to (a)-(d), with the data points in Classes A and B relabeled as "0" and "1," respectively.

for $P(y_i = 1 \,|\, \boldsymbol{x}_i) = 1 - e^{-\lambda(\boldsymbol{x}_i)}$, where $\lambda(\boldsymbol{x}_i)$ are calculated with (8). Note that due to the model construction, a single expert's influence on the decision boundary can be conveniently measured, and the exact decision boundary is bounded by a convex polytope. Thus it is not surprising that the convex polytope in Fig. 1 (c), which encloses the space marked as zero, aligns well with the contour line of $P(y_i = 1 \,|\, \boldsymbol{x}_i) = 0.5$ shown in Fig. 1 (d).

Despite being able to construct a nonlinear decision boundary bounded by a convex polytope, sum-softplus regression has a clear restriction in that if the data labels are flipped, its performance may substantially deteriorate, becoming no better than that of logistic regression. For example, for the same data shown in Fig. 1 (a), if we choose the opposite labeling setting where the data points of Class A are labeled as "0" and these of Class B are labeled as "1," then sum-softplus regression infers a single expert (hyperplane) with non-negligible weight, as shown in Figs. 1 (f)-(g), and fails to separate the data points of two different classes, as shown in Figs. 1 (g)-(h). The data log-likelihood plots in Fig. 1 (e) also suggest that sum-softplus regression could perform substantially better if the training data are labeled in favor of its geometric constraints on the decision boundary. An advantage of a Bayesian hierarchical model is that with collected MCMC samples, one may estimate not only the posterior means but also uncertainties. The standard deviations shown in Figs. 2 (b) and (d) clearly indicate the uncertainties of sum-softplus regression on its decision boundaries and predictive probabilities in the covariate space, which may be used to help decide how to sequentially query the labels of unlabeled data in an active learning setting [Cohn et al., 1996, Settles, 2010].

The sensitivity of sum-softplus regression to how the data are labeled could be mitigated but not completely solved by combining two sum-softplus regression models trained under the two opposite labeling settings. In addition, sum-softplus regression may not perform well no matter how the data are labeled if neither of the two classes could be enclosed by a convex polytope. To fully resolve these issues, we first introduce stack-softplus regression, which defines a convex-polytope-like confined space to enclose positive examples. We then show how to combine the two distinct, but complementary, softplus regression models to construct SS-softplus regression that provides more flexible nonlinear decision boundaries.

2.5 Stack-softplus regression and stacked gamma distributions

The model in (11) combines the BerPo link with a gamma belief network that stacks differently parameterized gamma distributions. Note that here "stacking" is defined as an operation that mixes the shape parameter of a gamma distribution at layer t with a gamma distribution at layer $t + 1$, the next one pushed into the stack, and pops out the covariate-dependent gamma scale parameters from layers $T+1$ to 2 in the stack, following the last-in-first-out rule, to parameterize the BerPo rate of the class label y_i shown in (10).

2.5.1 Convex-polytope-like confined space that favors positive examples

Let us make the analogy that each $\boldsymbol{\beta}^{(t)}$ is one of the T criteria that an expert examines before making a binary decision. From (10) it is clear that as long as a single criterion $t \in \{2, \dots, T+1\}$ of the expert is strongly violated, which means that $\boldsymbol{x}_i' \boldsymbol{\beta}^{(t)}$ is much smaller than zero, then the expert would vote "No" regardless of the values of $\boldsymbol{x}_i' \boldsymbol{\beta}^{(\tilde{t})}$ for all $\tilde{t} \neq t$. Thus the response variable could be voted "Yes" by the expert only if none of the T expert criteria are strongly violated. For stack-softplus regression, let us specify a confined space using the inequality $P(y_i = 1 \,|\, \boldsymbol{x}_i) = 1 - e^{-\lambda(\boldsymbol{x}_i)} > p_0$, which can be expressed as

$$\boldsymbol{x}_i' \boldsymbol{\beta}^{(T+1)} + \ln \ln \left\{ 1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(T)}} \ln \left[1 + \dots \ln \left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(2)}} \right) \right] \right\} > \ln \left[(1-p_0)^{-\frac{1}{r}} - 1 \right], \tag{18}$$

and hence any data point $\boldsymbol{x}_i$ outside the confined space (*i.e.*, violating the inequality in Eq. 18 a.s.) will be labeled as $y_i = 0$ with a probability no less than $1 - p_0$.

Considering the covariate space

$$\mathcal{T}^{-t} := \left\{ \boldsymbol{x}_i : \boldsymbol{x}_i' \boldsymbol{\beta}_{\tilde{t}} \geq 0 \text{ for } \tilde{t} \neq t \right\}, \tag{19}$$

where all the criteria except criterion t of the expert tend to be satisfied, the decision boundary of stack-softplus regression in $\mathcal{T}^{-t}$ would be clearly influenced by the satisfactory level of criterion t, whose hyperplane partitions $\mathcal{T}^{-t}$ into two parts as

$$y_i = \begin{cases} 1, & \text{if } 1 - \left(1^{T+1} + \ln \left\{ 1^T + \ln \left[1^{T-1} + \dots + \ln \left(1^t + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(t)}} g_{t-1} \right) \right] \right\} \right)^{-r} > p_0, \\ 0, & \text{otherwise}, \end{cases} \tag{20}$$

for all $\boldsymbol{x}_i \in \mathcal{T}^{-t}$. Let us define g_t with $g_1 = 1$ and the recursion $g_t = \ln(1 + g_{t-1})$ for $t = 2, \dots, T$, and define h_t with $h_{T+1} = (1-p_0)^{-\frac{1}{r}} - 1$ and the recursion $h_t = e^{h_{t+1}} - 1$ for $t = T, T-1, \dots, 2$. Using the definition of g_t and h_t, combining all the T expert criteria, the confined space of stack-softplus regression specified in (18) can be roughly related to a convex polytope, which is specified by the solutions to a set of T inequalities as

$$\boldsymbol{x}_i' \boldsymbol{\beta}^{(t)} > \ln(h_t) - \ln(g_{t-1}), \ t \in \{2, \dots, T+1\}. \tag{21}$$

The convex polytope is enclosed by the intersection of T V-dimensional hyperplanes, and since none of the T criteria would be strongly violated inside the convex polytope, the label

$y_i = 1$ ($y_i = 0$) would be assigned to an $\boldsymbol{x}_i$ inside (outside) the convex polytope with a relatively high (low) probability.

Unlike the confined space of sum-softplus regression defined in (16) that is bounded by a convex polytope defined in (17), the convex polytope defined in (21) only roughly corresponds to the confined space of stack-softplus regression, as defined in (18). Nevertheless, the confined space defined in (18) is referred to as a convex-polytope-like confined space, due to both its connection to the convex polytope in (21) and the fact that (18) is likely to be violated if at least one of the T criteria is strongly dissatisfied (*i.e.*, $e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(t)}} \rightarrow 0$ for some t).

2.5.2 Illustration for stack-softplus regression

Let us examine how stack-softplus regression performs on the same data used in Fig. 1. When Class B is labeled as "1," as shown in Fig. 3 (g), stack-softplus regression infers a convex polytope that encloses the space marked as $T = 20$ using the intersection of all $T = 20$ hyperplanes, each of which is defined as in (21); and as shown in Fig. 3 (h), it works well by using a convex-polytope-like confined space to enclose positive examples. However, as shown in Figs. 3 (c)-(e), its performance deteriorates when the opposite labeling setting is used. Note that due to the model construction that introduces complex interactions between the T hyperplanes, (21) can only roughly describe how a single hyperplane could influence the decision boundary determined by all hyperplanes. Thus it is not surprising that neither the convex polytope in Fig. 3 (c), which encloses the space marked with the largest count there, nor the convex polytope in Fig. 3 (g), which encloses the space marked with T, align well with the contour lines of $P(y_i = 1 \,|\, \boldsymbol{x}_i) = 0.5$ in Figs. 3 (d) and (h), respectively.

While how the latent count $m_{\cdot}^{(t)}$ decreases as t increases does not indicate a clear cutoff point for the depth T, neither do we observe a clear sign of overfitting when T is set as large as 100 in our experiments. Both Figs. 3 (c) and (g) indicate that most of the hyperplanes are far from any data points and tend to vote "Yes" for all training data. The standard deviations shown in Figs. 4 (f) and (h) clearly indicate the uncertainties of stack-softplus regression on its decision boundaries and predictive probabilities in the covariate space.

Like sum-softplus regression, stack-softplus regression also generalizes softplus and logistic regressions in that it uses the boundary of a confined space rather than a single hyperplane to partition the covariate space into two parts. Unlike the convex-polytope-bounded confined space of sum-softplus regression that favors placing negative examples inside it, the convex-polytope-like confined space of stack-softplus regression favors placing positive examples inside it. While both sum- and stack-softplus regressions could be sensitive to how the data are labeled, their distinct behaviors under the same labeling setting motivate us to combine them together as SS-softplus regression, as described below.

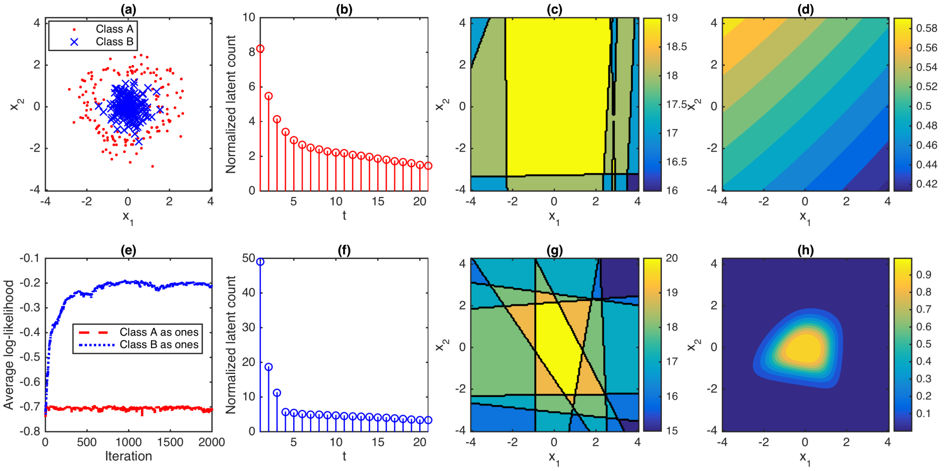

Figure 3: Analogous figure to Fig. 1 for stack-softplus regression with $T = 20$ expert criteria, with the following differences: (b) shows the average latent count per positive sample, $\sum_i m_i^{(t)} / \sum_i \delta(y_i = 1)$, as a function of layer t, (c) shows a contour map, the value of each point of which represents how many inequalities specified in (21) are satisfied, and whose region with the values of $T = 20$ corresponds to the convex polytope enclosed by the intersections of the hyperplanes defined in (21),

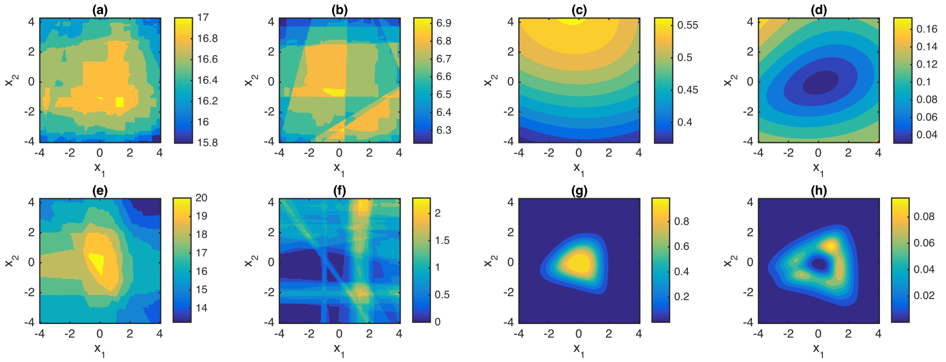

Figure 4: Analogous figure to Fig. 2 for stack-softplus regression, with the following differences: (a) and (b) show the contour maps of the posterior means and standard deviations, respectively, of the number of inequalities specified in (21) that are satisfied. (e)-(f) are analogous plots to (a)-(b) under the opposite labeling setting.

2.6 Sum-stack-softplus (SS-softplus) regression

Note that if $T = 1$, SS-softplus regression reduces to sum-softplus regression; if $K = 1$, it reduces to stack-softplus regression; and if $K = T = 1$, it reduces to softplus regression, which further reduces to logistic regression if the weight of the single expert is fixed at $r = 1$. To ensure that the SS-softplus regression model is well defined in its infinite limit, we provide the following proposition and present the proof in Appendix B.

Proposition 4. *The infinite product in sum-stack-softplus regression as*

$$e^{-\sum_{k=1}^{\infty} r_k\, \varsigma\left(\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(2:T+1)}\right)} = \prod_{k=1}^{\infty}\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T+1)}}\ln\left\{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T)}}\ln\left[1+\dots\ln\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(2)}}\right)\right]\right\}\right)^{-r_k}$$

is smaller than one and has a finite expectation that is greater than zero.

2.6.1 Union of convex-polytope-like confined spaces

We may consider SS-softplus regression as a multi-hyperplane model that employs a committee, consisting of countably infinite experts, to make a decision, where each expert is equipped with T criteria to be examined. The committee's distribution is obtained by convolving the distributions of countably infinite experts, each of which mixes T stacked covariate-dependent gamma distributions. For each $\boldsymbol{x}_i$, the committee votes "Yes" as long as at least one expert votes "Yes," and an expert could vote "Yes" if and only if none of its T criteria are strongly violated. Thus the decision boundary of SS-softplus regression can be considered as a union of convex-polytope-like confined spaces that all favor placing positively labeled data inside them, as described below, with the proofs deferred to Appendix B.

Theorem 5. *For sum-stack-softplus regression, the confined space specified by the inequality* $P(y_i = 1 \,|\, \boldsymbol{x}_i) = 1 - e^{-\lambda(\boldsymbol{x}_i)} > p_0$*, which can be expressed as*

$$\lambda(\boldsymbol{x}_i) = \sum_{k=1}^{\infty} r_k\, \varsigma\left(\boldsymbol{x}'\boldsymbol{\beta}_k^{(2)}, \dots, \boldsymbol{x}'\boldsymbol{\beta}_k^{(T+1)}\right) > -\ln(1-p_0), \tag{22}$$

encompasses the union of convex-polytope-like confined spaces, expressed as

$$\mathcal{D}_\star = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \dots,$$

where the kth convex-polytope-like confined space $\mathcal{D}_k$ *is specified by the inequality*

$$\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T+1)} + \ln\ln\left\{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T)}}\ln\left[1 + \dots \ln\left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(2)}}\right)\right]\right\} > \ln\left[(1-p_0)^{-\frac{1}{r_k}} - 1\right]. \tag{23}$$

Corollary 6. *For sum-stack-softplus regression, the confined space specified by the inequality* $P(y_i = 1 \,|\, \boldsymbol{x}_i) = 1 - e^{-\lambda(\boldsymbol{x}_i)} \leq p_0$ *is bounded by* $\bar{\mathcal{D}}_\star = \bar{\mathcal{D}}_1 \cap \bar{\mathcal{D}}_2 \cap \dots$

Proposition 7. *For any data point* $\boldsymbol{x}_i$ *that resides inside the union of countably infinite convex-polytope-like confined spaces* $\mathcal{D}_\star = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \dots$*, which means* $\boldsymbol{x}_i$ *satisfies at least one of the inequalities in* (23)*, it will be labeled under sum-stack-softplus regression with* $y_i = 1$ *with a probability greater than* p_0*, and* $y_i = 0$ *with a probability no greater than* $1 - p_0$.

2.6.2 Illustration for sum-stack-softplus regression

Let us examine how SS-softplus regression performs on the same dataset used in Fig. 1. When Class A is labeled as "1," as shown in Figs. 5 (b)-(c), SS-softplus regression infers about eight convex-polytope-like confined spaces, the intersection of six of which defines the boundary of the covariate space that separates the points that violate all inequalities in (23) from the ones that satisfy at least one inequality in (23). The union of these convex-polytope-like confined spaces defines a confined covariate space, which is included within the covariate space satisfying $P(y_i \,|\, \boldsymbol{x}_i) > 0.5$, as shown in Fig. 5 (d).

When Class B is labeled as "1," as shown in Fig. 5 (f)-(g), SS-softplus regression infers about six convex-polytope-like confined spaces, one of which defines the boundary of the covariate space that separates the points that violate all inequalities in (23) from the others for the covariate space show in Fig. 5 (g). The union of two convex-polytope-like confined spaces defines a confined covariate space, which is included in the covariate space with $P(y_i \,|\, \boldsymbol{x}_i) > 0.5$, as shown in Fig. 5 (h). Figs. 5 (f)-(g) also indicate that except for two convex-polytope-like confined spaces, the boundaries of all the other convex-polytope-like confined spaces are far from any data points and tend to vote "No" for all training data. The standard deviations shown in Figs. 6 (b), (d), (f), and (h) clearly indicate the uncertainties of SS-softplus regression on classification decision boundaries and predictive probabilities.

3 Gibbs sampling via data augmentation and marginalization

Since logistic, softplus, sum-softplus, and stack-softplus regressions can all be considered as special cases of SS-softplus regression, below we will focus on presenting the nonparametric Bayesian hierarchical model and Bayesian inference for SS-softplus regression.

The gamma process $G \sim \Gamma\mathrm{P}(G_0, 1/c)$ has an inherent shrinkage mechanism, as in the prior the number of atoms with weights larger than $\epsilon > 0$ follows $\mathrm{Pois}\left(\gamma_0 \int_\epsilon^\infty r^{-1}e^{-cr}dr\right)$, whose mean is finite a.s., where $\gamma_0 = G_0(\Omega)$ is the mass parameter of the gamma process. In practice, the atom with a tiny weight generally has a negligible impact on the final decision boundary of the model, hence one may truncate either the weight to be above ϵ or the number of atoms to be below K. One may also follow Wolpert et al. [2011] to use a reversible jump MCMC [Green, 1995] strategy to adaptively truncate the number of atoms for a gamma process, which often comes with a high computational cost. For the convenience of implementation, we truncate the number of atoms in the gamma process to be K by choosing a finite discrete base measure as $G_0 = \sum_{k=1}^K \frac{\gamma_0}{K}\delta_{\alpha_k}$, where K will be set sufficiently large to achieve a good approximation to the truly countably infinite model.

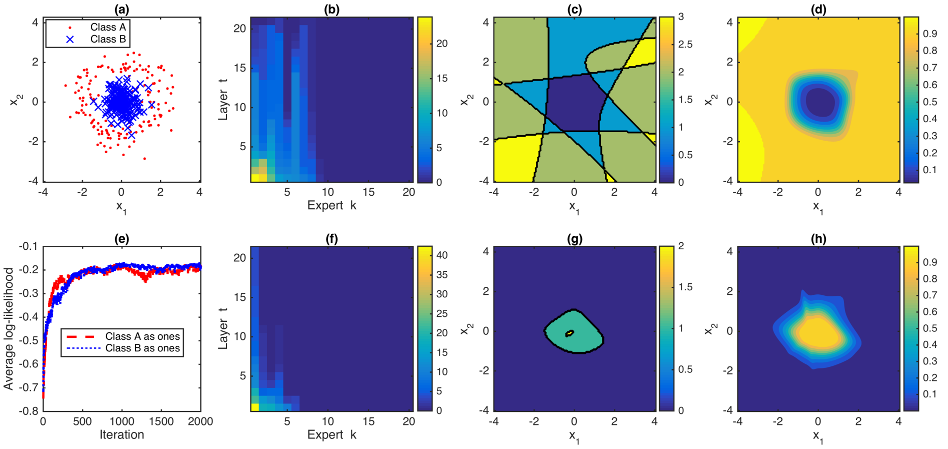

Figure 5: Analogous figure to Figs. 1 and 3 for SS-softplus regression with $K_{\max} = 20$ experts and $T = 20$ criteria for each expert, with the following differences: (b) shows the average latent count per positive sample, $\sum_i m_{ik}^{(t)} / \sum_i \delta(y_i = 1)$, as a function of both the expert index k and layer index t, where the experts are ordered based on the values of $\sum_i m_{ik}^{(1)}$, (c) shows a contour map, the value of each point of which represents how many inequalities specified in (23) are satisfied, and whose region with nonzero values corresponds to the union of convex-polytope-like confined spaces, each of which corresponds to an inequality defined in (23), and (f) and (g) are analogous plots to (b) and (c) under the opposite labeling setting where data in Class B are labeled as "1."

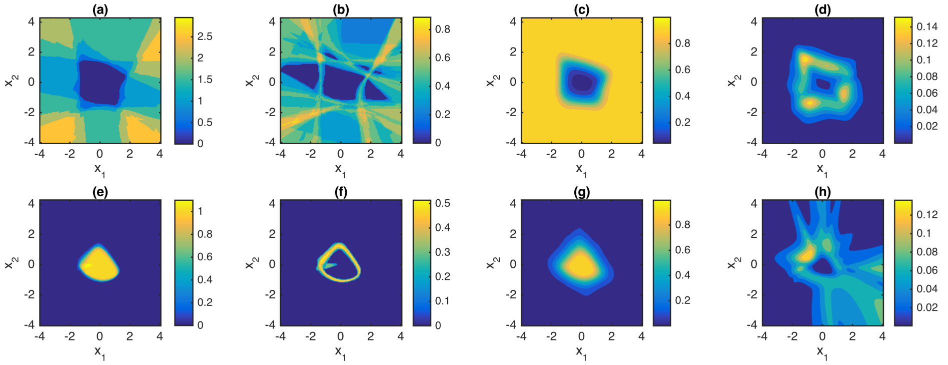

Figure 6: Analogous figure to Fig. 2 for SS-softplus regression, with the following differences: (a) and (b) show the contour maps of the posterior means and standard deviations, respectively, of the number of inequalities specified in (23) that are satisfied. (e)-(f) are analogous plots to (a)-(b).

We express the truncated SS-softplus regression model using (13) together with

$$r_k \sim \text{Gamma}(\gamma_0/K, 1/c_0),\ \gamma_0 \sim \text{Gamma}(a_0, 1/b_0),\ c_0 \sim \text{Gamma}(e_0, 1/f_0),$$

$$\boldsymbol{\beta}_k^{(t)} \sim \prod_{v=0}^{V} \mathcal{N}(0, \alpha_{vtk}^{-1}),\ \alpha_{vtk} \sim \text{Gamma}(a_t, 1/b_t), \tag{24}$$

where $t \in \{2, \dots, T+1\}$. Related to Tipping [2001], the normal gamma construction in (24) is used to promote sparsity on the regression coefficients $\boldsymbol{\beta}_k^{(t)}$. We derive Gibbs sampling by exploiting local conjugacies under a series of data augmentation and marginalization techniques. We comment here that while the proposed Gibbs sampling algorithm is a batch learning algorithm that processes all training data samples in each iteration, the local conjugacies revealed under data augmentation and marginalization may be of significant value in developing efficient mini-batch based online learning algorithms, including those based on stochastic gradient MCMC [Girolami and Calderhead, 2011, Ma et al., 2015, Patterson and Teh, 2013, Welling and Teh, 2011] and stochastic variation inference [Hoffman et al., 2013]. We leave the maximum likelihood, maximum a posteriori, (stochastic) variational Bayes inference, and stochastic gradient MCMC for softplus regressions for future research.

For a model with $T = 1$, we exploit the data augmentation techniques developed for the BerPo link in Zhou [2015] to sample m_i, these developed for the Poisson and multinomial distributions [Dunson and Herring, 2005, Zhou et al., 2012a] to sample m_{ik}, these developed for the NB distribution in Zhou and Carin [2015] to sample r_k and γ_0, and these developed for logistic regression in Polson and Scott [2011] and further generalized to NB regression in Zhou et al. [2012b] and Polson et al. [2013] to sample $\boldsymbol{\beta}_k$. We exploit local conjugacies to sample all the other model parameters. For a model with $T \geq 2$, we further generalize the inference technique developed for the gamma belief network in Zhou et al. [2015a] to sample the model parameters of deep hidden layers. Below we provide a theorem, related to Lemma 1 for the gamma belief network in Zhou et al. [2015a], to show that each regression coefficient vector can be linked to latent counts under NB regression. Let $m \sim \text{SumLog}(n, p)$ represent the sum-logarithmic distribution described in Zhou et al. [2015b], Corollary 9 further shows an alternative representation of (13), the hierarchical model of SS-softplus regression, where all the covariate-dependent gamma distributions are marginalized out.

Theorem 8. *Let us denote* $p_{ik}^{(t)} = 1 - e^{-q_{ik}^{(t)}}$, *i.e.,* $q_{ik}^{(t)} = -\ln(1 - p_{ik}^{(t)})$, *and* $\theta_{ik}^{(T+1)} = r_k$. *With* $q_{ik}^{(1)} := 1$ *and*

$$q_{ik}^{(t+1)} := \ln\left(1 + q_{ik}^{(t)} e^{\boldsymbol{x}_i \boldsymbol{\beta}_k^{(t+1)}}\right) \tag{25}$$

for $t = 1, \ldots, T$, *which means*

$$\begin{aligned} q_{ik}^{(t+1)} &= \varsigma\big(\boldsymbol{x}_i' \boldsymbol{\beta}_k^{(2)}, \ldots, \boldsymbol{x}_i' \boldsymbol{\beta}_k^{(t+1)}\big) \\ &= \ln\left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_k^{(t+1)}} \ln\left\{1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_k^{(t)}} \ln\left[1 + \ldots \ln\left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_k^{(2)}}\right)\right]\right\}\right), \end{aligned} \tag{26}$$

one may find latent counts $m_{ik}^{(t)}$ *that are connected to the regression coefficient vectors as*

$$m_{ik}^{(t)} \sim \text{NB}(\theta_{ik}^{(t+1)}, \quad 1 - e^{-q_{ik}^{(t+1)}}) = \text{NB}\left(\theta_{ik}^{(t+1)}, \frac{1}{1 + e^{-\boldsymbol{x}_i' \boldsymbol{\beta}_k^{(t+1)} - \ln(q_{ik}^{(t)})}}\right). \tag{27}$$

Corollary 9. *With* $q_{ik}^{(t)} = -\ln(1 - p_{ik}^{(t)})$ *defined as in* (26) *and hence* $p_{ik}^{(t)} = 1 - e^{-q_{ik}^{(t)}}$*, the hierarchical model of sum-stack-softplus regression can also be expressed as*

$$
\begin{aligned}
& m_{ik}^{(T+1)} \sim \mathrm{Pois}(r_k q_{ik}^{(T+1)}),\ r_k \sim \mathrm{Gamma}(\gamma_0/K, 1/c_0), \\
& m_{ik}^{(T)} \sim \mathrm{SumLog}(m_{ik}^{(T+1)}, p_{ik}^{(T+1)}),\ \boldsymbol{\beta}_k^{(T+1)} \sim \prod_{v=0}^{V} \mathcal{N}(0, \alpha_{v(T+1)k}^{-1}), \\
& \cdots \\
& m_{ik}^{(t)} \sim \mathrm{SumLog}(m_{ik}^{(t+1)}, p_{ik}^{(t+1)}),\ \boldsymbol{\beta}_k^{(t+1)} \sim \prod_{v=0}^{V} \mathcal{N}(0, \alpha_{v(t+1)k}^{-1}), \\
& \cdots \\
& y_i = \delta(m_i \geq 1),\ m_i = \sum_{k=1}^{K} m_{ik}^{(1)},\ m_{ik}^{(1)} \sim \mathrm{SumLog}(m_{ik}^{(2)}, p_{ik}^{(2)}),\ \boldsymbol{\beta}_k^{(2)} \sim \prod_{v=0}^{V} \mathcal{N}(0, \alpha_{v2k}^{-1}),
\end{aligned} \tag{28}
$$

We outline Gibbs sampling in Algorithm 1 of Appendix E, where to save computation, we consider setting $K_{\max}$ as the upper-bound of the number of experts and deactivating experts assigned with zero counts during MCMC iterations. We provide several additional model properties in Appendix C.2 to describe how the latent counts propagate across layers, which may be used to decide how to set the number of layers T. For simplicity, we consider the number of criteria for each expert as a parameter that determines the model capacity and we fix it as T for all experts in this paper. We defer the details of all Gibbs sampling update equations to Appendix C, in which we also describe in detail how to ensure numerical stability in a finite precision machine. Note that except for the sampling of $\{m_{ik}^{(1)}\}_k$, the sampling of all the other parameters of different experts are embarrassingly parallel.

4 Example Results

We compare softplus regressions with logistic regression, Gaussian radial basis function (RBF) kernel support vector machine (SVM) [Boser et al., 1992, Schölkopf et al., 1999, Vapnik, 1998], relevance vector machine (RVM) [Tipping, 2001], adaptive multi-hyperplane machine (AMM) [Wang et al., 2011], and convex polytope machine (CPM) [Kantchelian et al., 2014]. Except for logistic regression that is a linear classifier, both kernel SVM and RVM are widely used nonlinear classifiers relying on the kernel trick, and both AMM and CPM use the intersection of multiple hyperplanes to construct their decision boundaries. We discuss the connections between softplus regressions and previous work in Appendix D.

Following Tipping [2001], we consider the following datasets: banana, breast cancer, titanic, waveform, german, and image. For each of these six datasets, we consider the first ten predefined random training/testing partitions, and report both the mean and standard deviation of the testing classification errors. Since these datasets, originally provided by Rätsch et al. [2001], were no longer available on the authors' websites, we use the version provided by Diethe [2015]. We also consider two additional datasets—ijcnn1 and a9a—that come with a default training/testing partition, for which we report the results of logistic regression, SVM, and RVM based on a single trial, and report the results of all the other algorithms based on five independent trials with different random initiations. We summarize in Tab. 3 of Appendix E the basic information of these benchmark datasets.

Since the decision boundaries of all softplus regressions depend on whether the two classes are labeled as "1" and "0" or labeled as "0" and "1," we consider repeating the same softplus regression algorithm twice, using both $y_{i1} \sim \text{BerPo}[\lambda_1(\boldsymbol{x}_i)]$ and $y_{i2} \sim \text{BerPo}[\lambda_2(\boldsymbol{x}_i)]$, where y_{i1} and $y_{i2} := 1 - y_{i1}$ are the labels under two opposite labeling settings. We combine them to the following predictive distribution $y_i \,|\, \boldsymbol{x}_i \sim \text{Bernoulli}\left[(1 - e^{-\lambda_1(\boldsymbol{x}_i)} + e^{-\lambda_2(\boldsymbol{x}_i)})/2\right]$, which no longer depends on how the data are labeled. If we set $p_0 = 0.5$ as the probability threshold to make binary decisions, then y_i would be labeled as "1" if $\lambda_1(\boldsymbol{x}_i) > \lambda_2(\boldsymbol{x}_i)$ and labeled as "0" otherwise. This simple strategy to train the same asymmetric model under two opposite labeling settings and combine their results together is related to the one used in Kantchelian et al. [2014], which, however, lacks of probabilistic interpretation. We leave more sophisticated training and combination strategies to future study.

For all datasets, we consider 1) softplus regression, which generalizes logistic regression with $r \neq 1$, 2) sum-softplus regression, which reduces to softplus regression if the number of experts is $K = 1$, 3) stack-softplus regression, which reduces to softplus regression if the number of layers is $T = 1$, and 4) sum-stack-softplus (SS-softplus) regression, which reduces to sum-softplus regression if $T = 1$, to stack-softplus regression if $K = 1$, and to softplus regression if $K = T = 1$. For sum-softplus regresion, we set the upperbound on the number of experts as $K_{\max} = 20$, for deep softplus regression, we consider $T \in \{1, 2, 3, 5, 10\}$, and for SS-softplus regression, we set $K_{\max} = 20$ and consider $T \in \{1, 2, 3, 5, 10\}$. For all softplus regressions, we consider 5000 Gibbs sampling iterations and record the maximum likelihood sample found during the last 2500 iterations as the point estimates of r_k and $\boldsymbol{\beta}_k^{(t)}$, which are used for out-of-sample predictions. We set $a_0 = b_0 = 0.01$, $e_0 = f_0 = 1$, and $a_t = b_t = 10^{-6}$ for $t \in \{2, \ldots, T + 1\}$. As in Algorithm 1 shown in Appendix E, we deactivate inactive experts for iterations in $I_{Prune} = \{525, 575, \ldots, 4975\}$. For a fair comparison, to ensure that the same training/testing partitions are used for all algorithms for all datasets, we report the results by using either widely used open-source software packages or the code made public available by the original authors. We describe in Appendix E the settings of all the algorithms that are used for comparison.

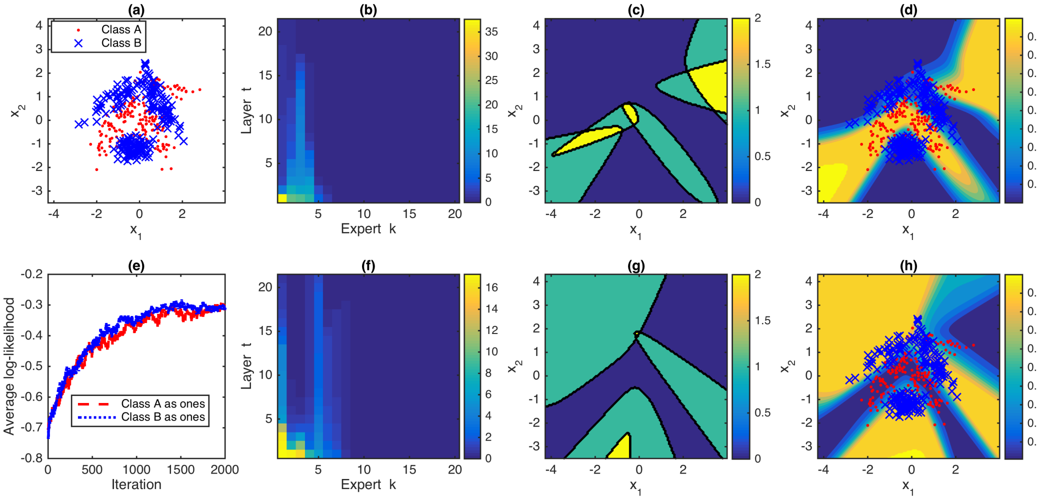

Figure 7: Analogous figure to Fig. 5 for the “banana” dataset.

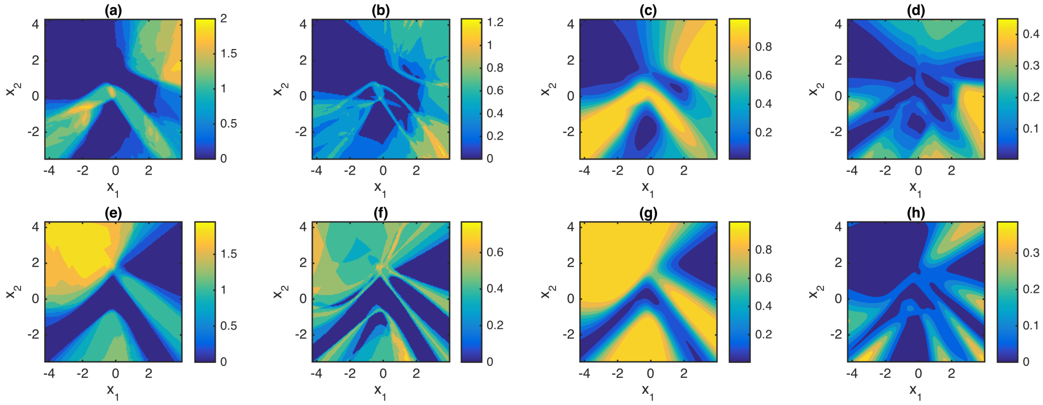

Figure 8: Analogous figure to Fig. 6 for the “banana” dataset.

4.1 Illustrations

With a synthetic dataset, Figs. 1-6 illustrate the distinctions and connections between the sum-, stack-, and SS-softplus regressions. While both sum- and stack-softplus could work well for the synthetic dataset if the two classes are labeled in their preferred ways, as shown in Figs. 1 and 3, SS-softplus regression, as shown in Fig. 5, works well regardless of how the data are labeled. To further illustrate how the distinct, but complementary, behaviors of the sum- and stack-softplus regressions are combined together in SS-softplus regression, let us examine how SS-softplus regression performs on the banana dataset shown in Fig. 7 (a). When Class A is labeled as “1,” as shown in Figs. 7 (b)-(c), SS-softplus regression infers about six convex-polytope-like confined spaces, the intersection of five of which defines the boundary of the covariate space that separates the points that satisfy at least one inequality in (23) from the ones that violate all inequalities in (23). The union of these convex-polytope-like confined spaces defines a confined covariate space, which is included within the covariate space satisfying $P(y_i \,|\, \boldsymbol{x}_i) > 0.5$, as shown in Fig. 7 (d).

When Class B is labeled as "1," as shown in Fig. 7 (f)-(g), SS-softplus regression infers about eight convex-polytope-like confined spaces, three of which define the boundary of the covariate space that separates the points that satisfy at least one inequality in (23) from the others for the covariate space show in Fig. 7 (g). The union of four convex-polytope-like confined spaces defines a confined covariate space, which is included in the covariate space with $P(y_i \,|\, \boldsymbol{x}_i) > 0.5$, as shown in Fig. 7 (h). Figs. 7 (f)-(g) also indicate that except for four convex-polytope-like confined spaces, all the other inferred convex-polytope-like confined spaces are far away from and tend to vote "No" for all training data. The standard deviations shown in Figs. 8 (b), (d), (f), and (h) indicate the uncertainties of SS-softplus regression on classification decision boundaries and predictive probabilities in the covariate space.

In Figs. 12-15 of Appendix E, we further illustrate SS-softplus regression on an exclusive-or (XOR) dataset and a double-moon dataset used in Haykin [2009]. For the banana, XOR, and double-moon datasets, where the two classes cannot be well separated by a single convex-polytope-like confined space, neither sum- nor stack-softplus regressions work well regardless of how the data are labeled, whereas SS-softplus regression infers the union of multiple convex-polytope-like confined spaces that successfully separates the two classes.

4.2 Classification performance on benchmark data

We summarize in Tab. 1 the results for the first six benchmark datasets described in Tab. 3, for each of which we report the results based on the first ten predefined random training/testing partitions. Overall for these six datasets, the RBF kernel SVM has the highest average out-of-sample prediction accuracy, followed closely by SS-softplus regression, whose mean of the errors normalized by these of of the SVM is as small as 1.033, and then by RVM, whose mean of normalized errors is 1.095. Overall, logistic regression does not perform well, which is not surprising as it is a linear classifier that uses a single hyperplane to partition the covariate space into two halves to separate one class from the other. Softplus regression, which uses an additional parameter over logistic regression, fails to reduce the classification errors of logistic regression; both sum-softplus regression, a multi-hyperplane generalization using the convolution operation, and stack-softplus regression, a multi-hyperplane generalization using the stacking operation, clearly reduce the classification errors; and SS-regression that combines both the convolution and stacking operations further improves the overall performance. Both CPM and AMM perform similarly to sum-softplus regression, which is not surprising given their connections discussed in Appendix D.2.

For out-of-sample prediction, the computation of a classification algorithm generally in-

Table 1: Comparison of classification errors of logistic regression (LR), RBF kernel support vector machine (SVM), relevance vector machine (RVM), adaptive multi-hyperplane machine (AMM), convex polytope machine (CPM), softplus regression, sum-softplus (sum-ς) regression with $K_{\max} = 20$, stack-softplus (stack-ς) regression with $T = 5$, and SS-softplus (SS-ς) regression with $K_{\max} = 20$ and $T = 5$. Displayed in each column of the last row is the average of the classification errors of an algorithm normalized by those of kernel SVM.

Dataset	LR	SVM	RVM	AMM	CPM	softplus	sum-ς	stack-ς (T=5)	SS-ς (T=5)
banana	47.76 ±4.38	**10.85** ±0.57	11.08 ±0.69	18.76 ±4.09	21.39 ±1.72	47.87 ±4.36	30.78 ±8.68	33.21 ±5.76	11.89 ±0.61
breast cancer	28.05 ±3.68	28.44 ±4.52	31.56 ±4.66	31.82 ±4.47	32.08 ±4.29	28.70 ±4.76	30.13 ±4.23	**27.92** ±3.31	28.83 ±3.40
titanic	22.67 ±0.98	22.33 ±0.63	23.20 ±1.08	28.85 ±8.56	22.37 ±0.45	22.53 ±0.43	22.48 ±0.25	22.71 ±0.70	**22.29** ±0.80
waveform	13.33 ±0.59	**10.73** ±0.86	11.16 ±0.72	11.81 ±1.13	12.76 ±1.17	13.62 ±0.71	11.51 ±0.65	12.25 ±0.69	11.69 ±0.69
german	23.63 ±1.70	23.30 ±2.51	23.67 ±2.28	25.13 ±3.73	25.03 ±2.49	24.07 ±2.11	23.60 ±2.39	**22.97** ±2.22	24.23 ±2.46
image	17.53 ±1.05	2.84 ±0.52	3.82 ±0.59	3.82 ±0.87	3.25 ±0.41	17.55 ±0.75	3.50 ±0.73	7.97 ±0.52	**2.73** ±0.53
Mean of SVM normalized errors	2.472	1	1.095	1.277	1.251	2.485	1.370	1.665	1.033

Table 2: Analogous table to Tab. 1 for comparing the number of experts (times the number of hyperplanes per expert), where an expert contains T hyperplanes for both stack- and SS-softplus regressions and contains a single hyperplane/support vector for all the others. The computational complexity for out-of-sample prediction is about linear in the number of hyperplanes/support vectors. Displayed in each column of the last row is the average of the number of experts (times the number of hyperplanes per expert) of an algorithm normalized by those of RBF kernel SVM.

Dataset	LR	SVM	RVM	AMM	CPM	softplus	sum-ς	stack-ς (T=5)	SS-ς (T=5)
banana	1	129.20 ±32.76	22.30 ±26.02	9.50 ±2.80	14.60 ±7.49	2	3.70 ±0.95	2 (×5)	7.60 (×5) ±1.17 (×5)
breast cancer	1	115.10 ±11.16	24.80 ±28.32	13.40 ±0.84	12.00 ±8.43	2	3.10 ±0.74	2 (×5)	6.40 (×5) ±1.43 (×5)
titanic	1	83.40 ±13.28	5.10 ±3.03	14.90 ±3.14	5.20 ±2.53	2	2.30 ±0.48	2 (×5)	4.00 (×5) ±0.94 (×5)
waveform	1	147.00 ±38.49	21.10 ±10.98	9.50 ±1.18	6.40 ±2.27	2	4.40 ±0.84	2 (×5)	8.90 (×5) ±2.33 (×5)
german	1	423.60 ±55.02	11.00 ±3.20	18.80 ±1.81	8.80 ±7.79	2	6.70 ±0.95	2 (×5)	14.70 (×5) ±1.77 (×5)
image	1	211.60 ±47.51	35.80 ±9.19	10.50 ±1.08	23.00 ±6.75	2	11.20 ±1.32	2 (×5)	17.60 (×5) ±1.90 (×5)
Mean of SVM normalized K	0.007	1	0.131	0.088	0.075	0.014	0.030	0.014 (×5)	0.057 (×5)

creases linearly in the number of used hyperplanes or support vectors. We summarize the number of experts (times the number of hyperplanes per expert if that number is not one) in Tab. 2, which indicates that in comparison to SVM that consistently requires the most number of experts (each expert corresponds to a support vector for SVM), the RVM, AMM, CPM, and the three proposed multi-hyperplane softplus regressions all require significantly less time for predicting the class label of a new data sample. It is also interesting to notice that the number of hyperplanes automatically inferred from the data by sum-softplus regression is generally smaller than the ones of AMM and CPM, both of which are selected through cross validations. Note that the number of active experts, defined as the value of $\sum_k \delta(\sum_i m_{ik}^{(1)} > 0)$, inferred by both sum- and SS-softplus regressions shown in Tab. 2 will be further reduced if we only take into consideration the experts whose weights are larger than a certain threshold, such as those with $r_k > 0.001$ for $k \in \{1, \dots, K_{\max}\}$.

Except for banana, a two-dimensional dataset, sum-softplus regression performs similarly to both AMM and CPM; and except for banana and image, stack-softplus regression performs similarly to both AMM and CPM. These results are not surprising as CPM, closely related to AMM, uses a convex polytope, defined as the intersection of multiple hyperplanes, to enclose one class, whereas the classification decision boundaries of sum-softplus regression, defined by the interactions of multiple hyperplanes via the sum-softplus function, can be bounded within a convex polytope that encloses negative examples, and that of stack-softplus regression can be related to a convex-polytope-like confined space that encloses positive examples. Note that while both sum- and stack-softplus regressions can partially remedy their sensitivity to how the data are labeled by combining the results obtained under two opposite labeling settings, the decision boundaries of them and those of both AMM and CPM are still restricted to a confined space related to a single convex polytope, which may be used to explain why on both banana and image, as well as on the XOR and double-moon datasets shown in Appendix E, they all clearly underperform SS-softplus regression, which separates two classes using the union of convex-polytope-like confined spaces.

For breast cancer, titanic, and german, all classifiers have comparable classification errors, suggesting minor or no advantages of using a nonlinear classifier on them. For these three datasets, it is interesting to notice that, as shown in Figs. 9-10, sum- and SS-softplus regressions infer no more than two and three experts, respectively, with non-negligible weights under both labeling settings. These interesting connections imply that for two linearly separable classes, while providing no obvious benefits but also no clear harms, both sum- and SS-softplus regressions tend to infer a few active experts, and both stack- and SS-softplus regressions exhibit no clear sign of overfitting as the number of expert criteria T increases.

Whereas for banana, waveform, and image, all nonlinear classifiers clearly outperform lo-

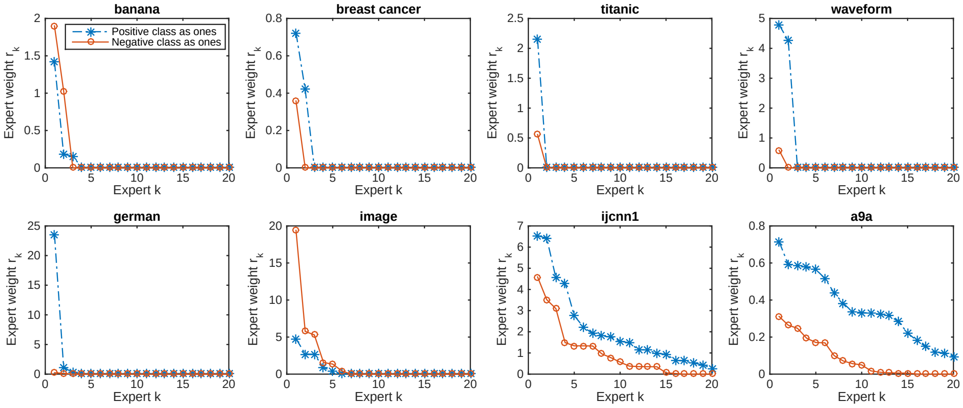

Figure 9: The inferred weights of the $K_{\max} = 20$ experts of sum-softplus regression, ordered from left to right according to their weights, on all datasets shown in Tab. 3, based on the maximum likelihood sample of a single random trial.

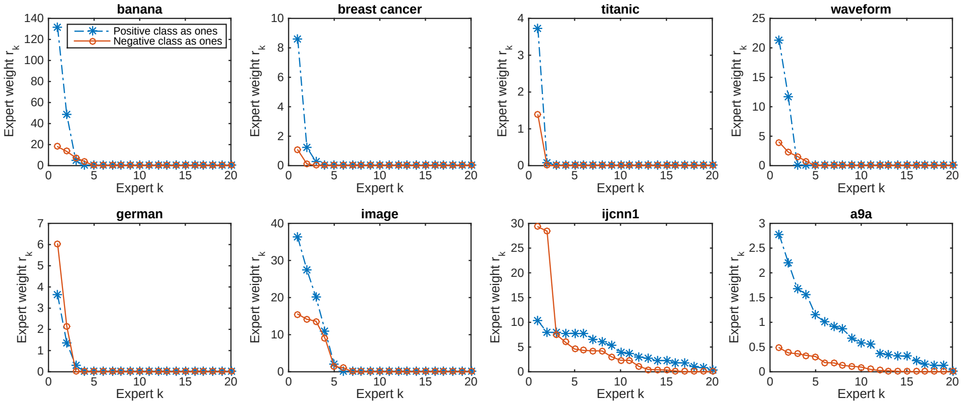

Figure 10: Analogous to Fig. 9 for SS-softplus regression with $K_{\max} = 20$ and $T = 5$.

gistic regression, and as shown in Figs. 9-10, sum- and SS-softplus regressions infer at least two and four experts, respectively, with non-negligible weights under at least one of the two labeling settings. These interesting connections imply that for two classes not linearly separable, both sum- and SS-softplus regressions may significantly outperform logistic regression by inferring a sufficiently large number of active experts, and both stack- and SS-softplus regressions may significantly outperform logistic regression by setting the number of expert criteria as $T \geq 2$, exhibiting no clear sign of overfitting as T further increases.

For both stack- and SS-softplus regressions, the computational complexity in both training and out-of-sample prediction increases linearly in T, the depth of the stack. To understand how increasing T affects the performance, we show in Tabs. 4-5 of Appendix E the classification errors of stack- and SS-softplus regressions, respectively, for $T \in \{1, 2, 3, 5, 10\}$. It is clear that increasing T from 1 to 2 generally leads to the most significant improvement if there is a clear advantage of increasing T, and once T is sufficiently large, further increasing T leads to small fluctuations of the performance but does not appear to lead to clear overfitting. It is also interesting to examine the number of active experts inferred by SS-softplus regression, where each expert is equipped with T hyperplanes, as T increases. As shown in Tab. 6 of Appendix E, this number has a clear increasing trend as T increases. This is not surprising as each expert is able to fit more complex geometric structure as T increases, and hence SS-softplus regression can employ more of them to more detailedly describe the decision boundaries. This phenomenon is also clearly visualized in comparing the inferred experts and decision boundaries for SS-softplus regression, as shown in Fig. 5, with those for sum-softplus regression, as shown in Fig. 1.

In addition to comparing softplus regressions with related algorithms on the six benchmark datasets used in Tipping [2001], we also consider ijcnn1 and a9a, two larger-scale benchmark datasets that have also been used in Chang et al. [2010], Wang et al. [2011] and Kantchelian et al. [2014]. In Appendix E, we report results on both datasets, whose training/testing partition is predefined, based on a single random trial for logistic regression, SVM, and RVM, and five independent random trials for AMM, CPM, and all softplus regressions. As shown in Tabs. 7-10 of Appendix E, we observe similar relationships between the classification errors and the number of expert criteria for both stack- and SS-softplus regressions, and both sum- and SS-softplus regressions provide a good comprise between the classification accuracies and amount of computation required for out-of-sample predictions.

As shown in Figs. 9-10, with the upper-bound of the number of experts set as $K_{\max} = 20$, for each of the first six datasets, both sum- and SS-softplus regressions shrink the weights of most of the 20 experts to be close to zero, clearly inferring the number of experts with non-negligible weights under both labeling settings. For both ijcnn1 and a9a, at one of the two labeling setting for both sum- and SS-softplus regressions, $K_{\max} = 20$ does not seem to be large enough to accommodate all experts with non-negligible weights. Thus we have also tried setting $K_{\max} = 50$, which is found to more clearly show the ability of the model to shrink the weights of unnecessary experts for both ijcnn1 and a9a, but at the expense of clearly increased computational complexity in both training and testing. The automatic shrinkage mechanism of the gamma process based sum- and SS-softplus regressions is attractive for both computation and implementation, as it allows setting $K_{\max}$ as large as permitted by the computational budget, without the need to worry about overfitting. Having the ability to support countably infinite experts in the prior and inferring a finite number of experts with non-negligible weights in the posterior is an attractive property of the proposed nonparametric Bayesian softplus regression models.

We comment that while we choose a fixed truncation to approximate a countably infinite nonparametric Bayesian model, it is possible to adaptive truncate the number of experts for the proposed gamma process based models, using strategies such as marginalizing out the underlying stochastic processes [Lijoi et al., 2007, Teh et al., 2006], performing reversible-jump MCMC [Green, 1995, Wolpert et al., 2011], and using slice sampling [Neal, 2003, Walker, 2007], which would be interesting topics for future research.

5 Conclusions

To regress a binary response variable on its covariates, we propose sum-, stack-, and sum-stack-softplus regressions that use, respectively, a convex-polytope-bounded confined space to enclose the negative class, a convex-polytope-like confined space to enclose the positive class, and a union of convex-polytope-like confined spaces to enclose the positive class. Sum-stack-softplus regression, including logistic regression and all the other softplus regressions as special examples, constructs a highly flexible nonparametric Bayesian predictive distribution by mixing the convolved and stacked covariate-dependent gamma distributions with the Bernoulli-Poisson distribution. The predictive distribution is deconvolved and demixed by inferring the parameters of the underlying nonparametric Bayesian hierarchical model using a series of data augmentation and marginalization techniques. In the proposed Gibbs sampler that has closed-form update equations, the parameters of different stacked gamma distributions can be updated in parallel within each iteration. Example results demonstrate that the proposed softplus regressions can achieve classification accuracies comparable to those of kernel support vector machine, but consume significant less computation for out-of-sample predictions, provide probability estimates, quantify uncertainties, and place interpretable geometric constraints on its classification decision boundaries directly in the original covariate space. It is of great interest to investigate how to generalize the proposed softplus regressions to model count, categorical, ordinal, and continuous response variables, and to model observed or latent multivariate discrete vectors. For example, to introduce covariate-dependence into a stick-breaking process mixture model [Ishwaran and James, 2001], one may consider replacing the normal cumulative distribution function used in the probit stick-breaking process of Chung and Dunson [2009] or the logistic function used in the logistic stick-breaking process of Ren et al. [2011] with the proposed softplus functions.

References

F. Aiolli and A. Sperduti. Multiclass classification with multi-prototype support vector machines. 6:817–850, 2005.

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *J. Amer. Statist. Assoc.*, 88(422):669–679, 1993.

C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Statist.*, 2(6):1152–1174, 1974.

Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *NIPS*, pages 153–160, 2007.

Y. Bengio, I. J. Goodfellow, and A. Courville. Deep Learning. Book in preparation for MIT Press, 2015.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

A. C. Cameron and P. K. Trivedi. *Regression Analysis of Count Data.* Cambridge, UK, 1998.

F. Caron and E. B. Fox. Sparse graphs using exchangeable random measures. *arXiv:1401.1137v3*, 2015.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin. Training and testing low-degree polynomial data mappings via linear SVM. *J. Mach. Learn. Res.*, 11:1471–1490, 2010.

Y. Chung and D. B. Dunson. Nonparametric bayes conditional distribution modeling with variable selection. *J. Amer. Statist. Assoc.*, 104:1646–1660, 2009.

R. T. Clemen and R. L. Winkler. Combining probability distributions from experts in risk analysis. *Risk analysis*, 19(2):187–203, 1999.

D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

D. R. Cox and E. J. Snell. *Analysis of binary data*, volume 32. CRC Press, 1989.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, 2:265–292, 2002.

T. Diethe. 13 benchmark datasets derived from the UCI, DELVE and STATLOG repositories. `https://github.com/tdiethe/gunnar_raetsch_benchmark_datasets/`, 2015.

N. Djuric, L. Lan, S. Vucetic, and Z. Wang. Budgetedsvm: A toolbox for scalable SVM approximations. *J. Mach. Learn. Res.*, 14:3813–3817, 2013.

C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia. Incorporating second-order functional knowledge for better option pricing. In *NIPS*, pages 472–478, 2001.

D. B. Dunson and A. H. Herring. Bayesian latent variable models for mixed discrete outcomes. *Biostatistics*, 6(1):11–25, 2005.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, pages 1871–1874, 2008.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Ann. Statist.*, 1(2):209–230, 1973.

R. A. Fisher, A. S. Corbet, and C. B. Williams. The relation between the number of species and the number of individuals in a random sample of an animal population. *Journal of Animal Ecology*, 12(1):42–58, 1943.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

B. E. Fristedt and L. F. Gray. *A modern approach to probability theory.* Birkhäuser, 1997.

C. Genest and J. V. Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135, 1986.

M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. Roy. Statist. Soc.: B*, 73(2):123–214, 2011.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *AISTATS*, pages 315–323, 2011.

C. Glynn, S. T. Tokdar, D. L. Banks, and B. Howard. Bayesian analysis of dynamic linear topic models. *arXiv:1511.03947*, 2015.

P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

M. Greenwood and G. U. Yule. An inquiry into the nature of frequency distributions representative of multiple happenings with particular reference to the occurrence of multiple attacks of disease or of repeated accidents. *J. R. Stat. Soc.*, 1920.

B. Grünbaum. *Convex Polytopes.* Springer New York, 2013.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer New York, 2001.

S. Haykin. *Neural networks and learning machines*, volume 3. Pearson Education Upper Saddle River, 2009.

R. Henao, Z. Gan, J. Lu, and L. Carin. Deep Poisson factor modeling. In *NIPS*, pages 2782–2790, 2015.

T. Heskes. Selecting weighting factors in logarithmic opinion pools. In *NIPS*, pages 266–272, 1998.

G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

G. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

M. D. Hoffman, D. M. Blei, C. Wang, and J. W. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

C. C. Holmes and L. Held. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1(1):145–168, 2006.

C. Hu, P. Rai, and L. Carin. Zero-truncated poisson tensor factorization for massive binary tensors. In *UAI*, 2015.

H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *J. Amer. Statist. Assoc.*, 96(453), 2001.

R. A. Jacobs. Methods for combining experts' probability assessments. *Neural Computation*, 7(5): 867–888, 1995.

O. Kallenberg. *Foundations of modern probability.* Springer, 2006.

A. Kantchelian, M. C. Tschantz, L. Huang, P. L. Bartlett, A. D. Joseph, and J. D. Tygar. Large-margin convex polytope machine. In *NIPS*, pages 3248–3256, 2014.

J. F. C. Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78, 1967.

J. F. C. Kingman. *Poisson Processes*. Oxford University Press, 1993.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

J. F. Lawless. Negative binomial and mixed Poisson regression. *Canadian Journal of Statistics*, 15 (3):209–225, 1987.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

A. Lijoi, R. H. Mena, and I. Prünster. Controlling the reinforcement in Bayesian non-parametric mixture models. *J. R. Stat. Soc.: Series B*, 69(4):715–740, 2007.

S. J. Long. *Regression Models for Categorical and Limited Dependent Variables*. SAGE, 1997.

Y. Ma, T. Chen, and E. Fox. A complete recipe for stochastic gradient MCMC. In *NIPS*, pages 2899–2907, 2015.

N. Manwani and P. S. Sastry. Learning polyhedral classifiers using logistic function. In *ACML*, pages 17–30, 2010.

N. Manwani and P. S. Sastry. Polyceptron: A polyhedral learning algorithm. *arXiv:1107.1564*, 2011.

P. McCullagh and J. A. Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *ICML*, pages 807–814, 2010.

R. M. Neal. Slice sampling. *Ann. Statist.*, pages 705–741, 2003.

S. Patterson and Y. W. Teh. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *NIPS*, pages 3102–3110, 2013.

N. G. Polson and J. G. Scott. Default Bayesian analysis for multi-way tables: a data-augmentation approach. *arXiv:1109.4180v1*, 2011.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using Pólya–Gamma latent variables. *J. Amer. Statist. Assoc.*, 108(504):1339–1349, 2013.

P. Rai, C. Hu, R. Henao, and L. Carin. Large-scale bayesian multi-label learning via topic-based label embeddings. In *NIPS*, 2015.

C. E. Rasmussen. The infinite Gaussian mixture model. In *NIPS*, pages 554–560, 2000.

G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine learning*, 42(3): 287–320, 2001.

L. Ren, L. Du, L. Carin, and D. Dunson. Logistic stick-breaking process. *J. Mach. Learn. Res.*, 12:203–239, 2011.

B. Schölkopf, C. J. C. Burges, and A. J. Smola. *Advances in kernel methods: support vector learning*. MIT Press, 1999.

B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

W. Shang, K. Sohn, D. Almeida, and H. Lee. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *ICML*, 2016.

A. Smith, T. Cohn, and M. Osborne. Logarithmic opinion pools for conditional random fields. In *ACL*, pages 18–25, 2005.

I. Steinwart. Sparseness of support vector machines. *J. Mach. Learn. Res.*, 4:1071–1105, 2003.

M. Stone. The opinion pool. *Annals of Mathematical Statistics*, 32(4):1339–1342, 1961.

D. M. J. Tax, M. Van Breukelen, R. P. W. Duin, and J. Kittler. Combining multiple classifiers by averaging or by multiplying? *Pattern recognition*, 33(9):1475–1485, 2000.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *J. Amer. Statist. Assoc.*, 2006.

M. Tipping. Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1: 211–244, June 2001.

A. Todeschini and F. Caron. Exchangeable random measures for sparse and modular graphs with overlapping communities. *arXiv:1602.02114*, 2016.

V. Vapnik. *Statistical learning theory*. Wiley New York, 1998.

S. G. Walker. Sampling the Dirichlet mixture model with slices. *Communications in Statistics Simulation and Computation*, 2007.

Z. Wang, N. Djuric, K. Crammer, and S. Vucetic. Trading representability for scalability: adaptive multi-hyperplane machine for nonlinear classification. In *KDD*, pages 24–32, 2011.

M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *ICML*, pages 681–688, 2011.

M. Welling, M. Rosen-Zvi, and G. E. Hinton. Exponential family harmoniums with an application to information retrieval. In *NIPS*, pages 1481–1488, 2004.

R. Winkelmann. *Econometric Analysis of Count Data*. Springer, Berlin, 5th edition, 2008.

R. L. Wolpert, M. A. Clyde, and C. Tu. Stochastic expansions using continuous dictionaries: Lévy Adaptive Regression Kernels. *Ann. Statist.*, 39(4):1916–1962, 2011.

E. P. Xing, R. Yan, and A. G. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *UAI*, 2005.

M. Zhou. Infinite edge partition models for overlapping community detection and link prediction. In *AISTATS*, pages 1135–1143, 2015.

M. Zhou and L. Carin. Negative binomial process count and mixture modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(2):307–320, 2015.

M. Zhou, L. Hannah, D. Dunson, and L. Carin. Beta-negative binomial process and Poisson factor analysis. In *AISTATS*, pages 1462–1471, 2012a.

M. Zhou, L. Li, D. Dunson, and L. Carin. Lognormal and gamma mixed negative binomial regression. In *ICML*, pages 1343–1350, 2012b.

M. Zhou, Y. Cong, and B. Chen. The Poisson gamma belief network. In *NIPS*, 2015a.

M. Zhou, O. H. M. Padilla, and J. G. Scott. Priors for random count matrices derived from a family of negative binomial processes. *To appear in J. Amer. Statist. Assoc.*, 2015b.

Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. CRC Press, 2012.

Softplus Regressions and Convex Polytopes: Supplementary Materials

A Polya-Gamma distribution

To infer the regression coefficient vector for each NB regression, we use the Polya-Gamma random variable $X \sim \mathrm{PG}(a, c)$, defined in Polson and Scott [2011] as the weighted sum of infinite independent, and identically distributed (*i.i.d.*) gamma random variables as

$$X = \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k-1/2)^2 + c^2/(4\pi^2)}, \quad g_k \sim \mathrm{Gamma}(a, 1). \tag{29}$$

As in Polson et al. [2013], the moment generating function of the Polya-Gamma random variable $X \sim \mathrm{PG}(a, c)$ can be expressed as

$$\mathbb{E}[e^{sX}] = \cosh^{a}\left(\frac{c}{2}\right)\cosh^{-a}\left(\sqrt{\frac{c^2/2 - s}{2}}\right).$$

Let us denote $f(s) = \sqrt{\frac{c^2/2-s}{2}}$ and hence $f'(s) = -1/[4f(s)]$. Since

$$\frac{d\mathbb{E}[e^{sX}]}{ds} = \frac{a}{4}\mathbb{E}[e^{sX}]\frac{\tanh[f(s)]}{f(s)},$$

$$\frac{d^2\mathbb{E}[e^{sX}]}{ds^2} = \frac{1}{\mathbb{E}[e^{sX}]}\left(\frac{d\mathbb{E}[e^{sX}]}{ds}\right)^2 + \frac{a}{4}\mathbb{E}[e^{sX}]\cosh^{-2}[f(s)]\frac{\sinh[2f(s)] - 2f[s]}{[2f(s)]^3},$$

the mean can be expressed as

$$\mathbb{E}[X \mid a, c] = \frac{d\mathbb{E}[e^{sX}]}{ds}(0) = \frac{a}{2|c|}\tanh\left(\frac{|c|}{2}\right), \tag{30}$$

and the variance can be expressed as

$$\begin{aligned}
\mathrm{var}[X \mid a, c] &= \frac{d^2\mathbb{E}[e^{sX}]}{ds^2}(0) - (\mathbb{E}[X])^2 = \frac{a\cosh^{-2}\left(|c|/2\right)[\sinh(|c|) - |c|]}{4|c|^3} = \frac{a}{2|c|^3}\frac{\sinh(|c|) - |c|}{\cosh(|c|) + 1} \\
&= \frac{a}{2|c|^3}\frac{1 - e^{-2|c|} - 2|c|e^{-|c|}}{1 + e^{-2|c|} + 2e^{-|c|}} = \frac{a\cosh^{-2}\left(\frac{|c|}{2}\right)}{4}\left(\frac{1}{6} + \sum_{n=1}^{\infty}\frac{|c|^{2n}}{(2n+3)!}\right),
\end{aligned} \tag{31}$$

which matches the variance shown in Glynn et al. [2015] but with a much simpler derivation.

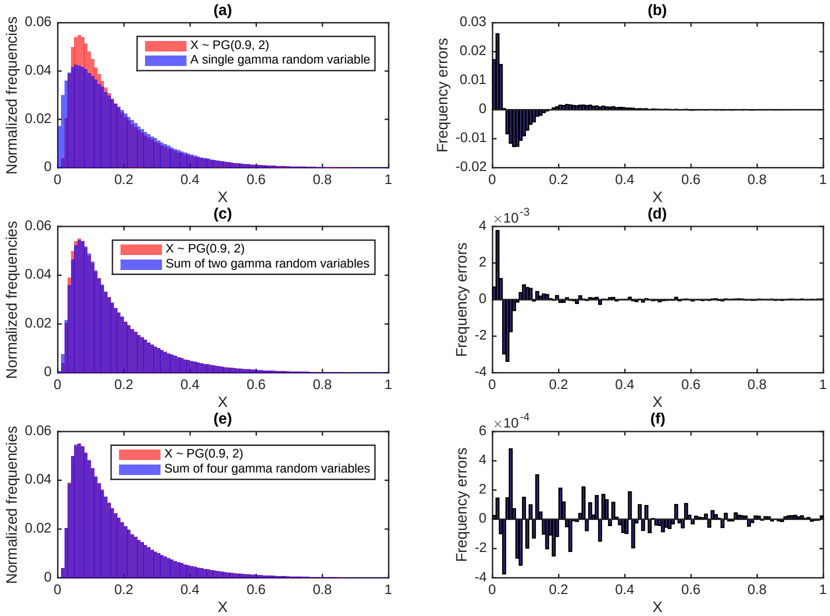

Figure 11: (a) Comparison of the normalized histogram of 10^6 independent random samples following $X \sim \mathrm{PG}(0.9, 2)$ (simulated using the truncated sampler at a truncation level of 1000) and that of 10^6 ones simulated with the approximate sampler truncated at one, *i.e.*, simulated from $X \sim \mathrm{Gamma}\left(\mu_\triangle^2/\sigma_\triangle^2, \sigma_\triangle^2/\mu_\triangle\right)$, where $\mu_\triangle = 0.1714$ and $\sigma_\triangle^2 = 0.0192$ are chosen to match the mean and variance of $X \sim \mathrm{PG}(0.9, 2)$; (b) the differences between the normalized frequencies of these two histograms. (c)-(d): analogous plots to (a)-(b), with the truncated sampler truncated at two. (e)-(f): analogous plots to (a)-(b), with the truncated sampler truncated at four.

As in (29), a PG distributed random variable can be generated from an infinite sum of weighted *i.i.d.* gamma random variables. In Polson et al. [2013], when a is an integer, $X \sim \mathrm{PG}(a, c)$ is sampled exactly using a rejection sampler, but when a is a positive real number, it is sampled approximately by truncating the infinite sum in (29). However, the mean of X approximately generated in this manner is guaranteed to be left biased. An improved approximate sampler is proposed in Zhou et al. [2012b] to compensate the bias of the mean, but not the bias of the variance.

We present in Proposition 10 an approximate sampler that is unbiased in both the mean and variance, using the summation of a finite number of gamma random variables. As shown in Fig. 11, the approximate sampler is quite accurate even only using two gamma random variables. We also provide some additional propositions, whose proofs are deferred to Appendix B, to describe some important properties that will be used in inference.

Proposition 10. *Denoting $K \in \{1, 2, \ldots\}$ as a truncation level, if*

$$\hat{X} = \frac{1}{2\pi^2} \sum_{k=1}^{K-1} \frac{g_k}{(k - 1/2)^2 + c^2/(4\pi^2)}, \quad g_k \sim \mathrm{Gamma}(a, 1) \tag{32}$$

and $X_\triangle \sim \mathrm{Gamma}\left(\mu_\triangle^2\sigma_\triangle^{-2}, \mu_\triangle^{-1}\sigma_\triangle^2\right)$, *where*

$$\mu_\triangle = \frac{a}{2|c|}\tanh\left(\frac{|c|}{2}\right) - \mathbb{E}[\hat{X}], \quad \sigma_\triangle^2 = \frac{a}{2|c|^3}\frac{\sinh(|c|)-|c|}{\cosh(|c|)+1} - \mathrm{var}[\hat{X}],$$

then $\hat{X} + X_\triangle$ *has the same mean and variance as those of* $X \sim \mathrm{PG}(a,c)$*, and the difference between their cumulant generating functions can be expressed as*

$$\ln \mathbb{E}[e^{sX}] - \ln \mathbb{E}[e^{s(\hat{X}+X_\triangle)}] = \sum_{n=3}^{\infty}\frac{s^n}{n}\left[\left(a\sum_{k=K}^{\infty} d_k^{-n}\right) - \frac{\mu_\triangle^2}{\sigma_\triangle^2}\left(\frac{\sigma_\triangle^2}{\mu_\triangle}\right)^n\right],$$

where $d_k = 2\pi^2(k-1/2)^2 + c^2/2$.

Proposition 11. *If* $X \sim \mathrm{PG}(a,c)$*, then* $\lim_{|c|\to\infty} X = 0$ *and* $\lim_{|c|\to\infty} |c|X = a/2$.

Proposition 12. *If* $X \sim \mathrm{PG}(a,0)$*, then* $\mathbb{E}[X\,|\,a,0] = a/4$ *and* $\mathrm{var}[X\,|\,a,0] = a/24$.

Proposition 13. *If* $X \sim \mathrm{PG}(a,c)$*, then* $\mathrm{var}[X\,|\,a,c] \geq \dfrac{a}{24}\cosh^{-2}\left(\dfrac{|c|}{2}\right)$*, where the equality holds if and only if* $c = 0$.

Proposition 14. $X \sim \mathrm{PG}(a,c)$ *has a variance-to-mean ratio as*

$$\frac{\mathrm{var}[X\,|\,a,c]}{\mathbb{E}[X\,|\,a,c]} = \frac{1}{|c|^2} - \frac{1}{|c|\sinh(|c|)}$$

and is always under-dispersed, since $\mathrm{var}[X\,|\,a,c] \leq \mathbb{E}[X\,|\,a,c]/6$ *almost surely, where the equality holds if and only if* $c = 0$.

B Proofs

Proof for Definition 1. For the hierarchical model in (7), we have $P(y_i = 0\,|\,\theta_i) = e^{-\theta_i}$. Further using the moment generating function of the gamma distribution, we have

$$P(y_i = 0\,|\,\boldsymbol{x}_i, \boldsymbol{\beta}) = \mathbb{E}_{\theta_i}[e^{-\theta_i}] = (1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}})^{-1}.$$

As $\lambda(\boldsymbol{x}_i) = -\ln[P(y_i = 0\,|\,\boldsymbol{x}_i, \boldsymbol{\beta})]$ by definition, we have $\lambda(\boldsymbol{x}_i) = \ln(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}})$. □

Proof for Definition 2. For the hierarchical model in (9), we have $P(y_i = 0\,|\,\theta_i) = e^{-\theta_i} = \prod_{k=1}^{\infty} e^{-\theta_{ik}}$. Using the moment generating function of the gamma distribution, we have

$$P(y_i = 0\,|\,\boldsymbol{x}_i, \{\boldsymbol{\beta}_k\}_k) = \mathbb{E}_{\theta_i}\left[\prod_{k=1}^{\infty} e^{-\theta_{ik}}\right] = \prod_{k=1}^{\infty}\mathbb{E}_{\theta_{ik}}[e^{-\theta_{ik}}] = \prod_{k=1}^{\infty}(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k})^{-r_k}.$$

As $\lambda(\boldsymbol{x}_i) = -\ln[P(y_i = 0 \,|\, \boldsymbol{x}_i, \{\boldsymbol{\beta}_k\}_k)]$ by definition, we obtain (8). □

Proof for Definition 3. For the hierarchical model in (11), we have $P(y_i = 0 \,|\, \theta_i^{(1)}) = e^{-\theta_i^{(1)}}$. Using the moment generating function of the gamma distribution, we have

$$P(y_i = 0 \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}^{(2)}, \theta_i^{(2)}) = \mathbb{E}_{\theta_i^{(1)}}\left[e^{-\theta_i^{(1)}}\right] = (1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(2)}})^{-\theta_i^{(2)}} = e^{-\theta_i^{(2)}\ln(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(2)}})}.$$

Marginalizing out $\theta_i^{(2)}$ leads to

$$P(y_i = 0 \,|\, \boldsymbol{x}_i, \boldsymbol{\beta}^{(2:3)}, \theta_i^{(3)}) = \mathbb{E}_{\theta_i^{(2)}}\left[e^{-\theta_i^{(2)}\ln(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(2)}})}\right] = e^{-\theta_i^{(3)}\ln[1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(3)}}\ln(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(2)}})]}.$$

Further marginalizing out $\theta_i^{(3)}, \ldots, \theta_i^{(T)}$ and with $\lambda(\boldsymbol{x}_i) = -\ln[P(y_i = 0 \,|\, \boldsymbol{x}_i, r, \boldsymbol{\beta}^{(2:T+1)})]$ by definition, we obtain (10). □

Proof for Definition 4. For the hierarchical model in (13), we have $P(y_i = 0 \,|\, \{\theta_{ik}^{(1)}\}_k) = e^{-\sum_{k=1}^{\infty}\theta_{ik}^{(1)}}$. Using the moment generating function of the gamma distribution, we have

$$P(y_i = 0 \,|\, \boldsymbol{x}_i, \{\boldsymbol{\beta}_k^{(2)}, \theta_{ik}^{(2)}\}_k) = \prod_{k=1}^{\infty} e^{-\theta_{ik}^{(2)}\ln\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(2)}}\right)}.$$

Further marginalizing out $\{\theta_{ik}^{(2)}\}_k, \ldots, \{\theta_{ik}^{(T)}\}_k$ and by definition with $\lambda(\boldsymbol{x}_i) = -\ln[P(y_i = 0 \,|\, \boldsymbol{x}_i, \{r_k, \boldsymbol{\beta}_k^{(2:T+1)}\}_k)]$, we obtain (12). □

Proof of Proposition 1. By construction, the infinite product would be equal or small than one. We need to further make sure that the infinite product would not degenerate to zero. Using the Lévy-Khintchine theorem [Kallenberg, 2006], we have

$$-\ln\left\{\mathbb{E}_G\left[e^{-\sum_{k=1}^{\infty} r_k \ln[1+\exp(\boldsymbol{x}_i'\boldsymbol{\beta}_k)]}\right]\right\} = \int_{\mathbb{R}_+\times\Omega}\left[1 - \left(\frac{1}{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}\right)^r\right]\nu(drd\boldsymbol{\beta}).$$

where $\nu(drd\boldsymbol{\beta}) = r^{-1}e^{-cr}drG_0(d\boldsymbol{\beta})$ is the Lévy measure of the gamma process. Since if $c \geq 0$, then $1 - e^{-cx} \leq cx$ for all $x \geq 0$, the right-hand-side term of the above equation would be bounded below

$$\int_{\mathbb{R}_+\times\Omega} r\ln[1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}]\nu(drd\boldsymbol{\beta}) = \frac{\gamma_0}{c}\int_{\Omega}\ln[1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}]g_0(d\boldsymbol{\beta}). \tag{33}$$

Since $e^{e^x} = 1 + e^x + \sum_{n=2}^{\infty}\frac{e^{nx}}{n!} \geq 1 + e^x$, we have

$$\ln(1+e^x) \leq e^x, \tag{34}$$

where the equality is true if and only if $x = -\infty$. Assuming $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, we have

$$\int_{\Omega} \ln[1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}}] g_0(d\boldsymbol{\beta}) \leq \int_{\Omega} e^{\boldsymbol{x}_i' \boldsymbol{\beta}} \mathcal{N}(\boldsymbol{\beta}; 0, \boldsymbol{\Sigma}) d\boldsymbol{\beta} = e^{\frac{1}{2} \boldsymbol{x}_i' \boldsymbol{\Sigma} \boldsymbol{x}_i}$$

Thus the integral in the right-hand-side of (33) is finite and hence the infinite product $\prod_{k=1}^{\infty} \left[1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_k}\right]^{-r_k}$ has a finite expectation that is greater than zero. □

Proof of Theorem 2. Since $\sum_{k' \neq k} r_{k'} \ln(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_{k'}}) \geq 0$ a.s., if (16) is true, then $r_k \ln(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_k}) \leq -\ln(1 - p_0)$ a.s. for all $k \in \{1, 2, \ldots\}$. Thus if (16) is true, then (17) is true a.s., which means the set of solutions to (16) is included in the set of solutions to (17). □

Proof of Proposition 3. Assuming $\boldsymbol{x}_i$ violates at least the kth inequality, which means $\boldsymbol{x}_i' \boldsymbol{\beta}_k > \ln\left[(1 - p_0)^{-\frac{1}{r_k}} - 1\right]$, then we have

$$\lambda(\boldsymbol{x}_i) = r_k \ln(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_k}) + \sum_{k' \neq k} r_{k'} \ln(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_{k'}}) \geq r_k \ln(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_k}) > -\ln(1 - p_0)$$

and hence $P(y_i = 1 \,|\, \boldsymbol{x}_i) = 1 - e^{-\lambda(\boldsymbol{x}_i)} > p_0$ and $P(y_i = 0 \,|\, \boldsymbol{x}_i) \leq 1 - p_0$. □

Proof of Proposition 4. By construction, the infinite product would be equal or small than one. We need to further make sure that the infinite product would not degenerate to zero. Using the Lévy-Khintchine theorem [Kallenberg, 2006] and $1 - e^{-cx} \leq cx$ for all $x \geq 0$ if $c \geq 0$, we have

$$\begin{aligned}
&- \ln \left\{ \mathbb{E}_G \exp \left[- \sum_{k=1}^{\infty} r_k \ln \left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_k^{(T+1)}} \ln \left\{ 1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_k^{(T)}} \ln \left[1 + \ldots \ln \left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}_k^{(2)}} \right) \right] \right\} \right) \right] \right\} \\
&= \int \left[1 - \left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(T+1)}} \ln \left\{ 1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(T)}} \ln \left[1 + \ldots e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(3)}} \ln \left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(2)}} \right) \right] \right\} \right)^{-r} \right] \nu(dr d\boldsymbol{\beta}^{(T+1:2)}) \\
&\leq \int r \ln \left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(T+1)}} \ln \left\{ 1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(T)}} \ln \left[1 + \ldots e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(3)}} \ln \left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(2)}} \right) \right] \right\} \right) \nu(dr d\boldsymbol{\beta}^{(T+1:2)}). \qquad (35)
\end{aligned}$$

Since $\ln(1 + e^x) \leq e^x$, we have

$$\begin{aligned}
& \ln \left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(T+1)}} \ln \left\{ 1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(T)}} \ln \left[1 + \ldots e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(3)}} \ln \left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(2)}} \right) \right] \right\} \right) \\
\leq \; & \ln \left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(T+1)}} \ln \left\{ 1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(T)}} \ln \left[1 + \ldots e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(4)}} \ln \left(1 + e^{\boldsymbol{x}_i' (\boldsymbol{\beta}^{(3)} + \boldsymbol{\beta}^{(2)})} \right) \right] \right\} \right) \\
\leq \; & \ln \left(1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(\star)}} \right) \leq e^{\boldsymbol{x}_i' \boldsymbol{\beta}^{(\star)}}
\end{aligned}$$

where $\boldsymbol{\beta}^{(\star)} := \boldsymbol{\beta}^{(T+1)} + \boldsymbol{\beta}^{(T)} + \ldots + \boldsymbol{\beta}^{(2)}$. Assuming $\boldsymbol{\beta}^{(t)} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_t)$, the right hand side of (35) would be bound below $\int r e^{\boldsymbol{x}_i'\boldsymbol{\beta}^{(\star)}} \nu(dr d\boldsymbol{\beta}^{(\star)}) = \gamma_0 c^{-1} e^{\frac{1}{2}\boldsymbol{x}_i'(\sum_{t=2}^{T+1}\boldsymbol{\Sigma}_t)\boldsymbol{x}_i}$. Therefore, the integral in the right-hand-side of (35) is finite and hence the infinite product in Proposition 4 has a finite expectation that is greater than zero under the gamma process. □

Proof of Theorem 5. Since

$$\sum_{k'\neq k} r_{k'} \ln\left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{k'}^{(T+1)}} \ln\left\{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{k'}^{(T)}} \ln\left[1 + \ldots \ln\left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_{k'}^{(2)}}\right)\right]\right\}\right) \geq 0$$

a.s., if (23) is true for at least one $k \in \{1, 2, \ldots\}$, then (22) is true a.s., which means the set of solutions to (22) encompass $\mathcal{D}_\star$. □

Proof of Proposition 7. Assume $\boldsymbol{x}_i$ satisfies at least the kth inequality, which means (23) is true, then

$$\lambda(\boldsymbol{x}_i) \geq r_k \ln\left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T+1)}} \ln\left\{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(T)}} \ln\left[1 + \ldots \ln\left(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(2)}}\right)\right]\right\}\right) > -\ln(1-p_0)$$

and hence $P(y_i = 1 \,|\, \boldsymbol{x}_i) = 1 - e^{-\lambda(\boldsymbol{x}_i)} > p$ and $P(y_i = 0 \,|\, \boldsymbol{x}_i) \leq 1 - p_0$. □

Proof of Theorem 8. By construction (27) is true for $t = 1$. Suppose (27) is also true for $t \geq 2$, then we can augment each $m_{ik}^{(t)}$ under its compound Poisson representation as

$$m_{ik}^{(t)} \,|\, m_{ik}^{(t+1)} \sim \text{SumLog}(m_{ik}^{(t+1)},\ p_{ik}^{(t+1)}),\ \ m_{ik}^{(t+1)} \sim \text{Pois}\left(\theta_{ik}^{(t+1)} q_{ik}^{(t+1)}\right), \tag{36}$$

where the joint distribution of $m_{ik}^{(t)}$ and $m_{ik}^{(t+1)}$, according to Theorem 1 of Zhou and Carin [2015], is the same as that in

$$m_{ik}^{(t+1)} \,|\, m_{ik}^{(t)} \sim \text{CRT}(m_{ik}^{(t)},\ \theta_{ik}^{(t+1)}),\ \ m_{ik}^{(t)} \sim \text{NB}(\theta_{ik}^{(t+1)},\ p_{ik}^{(t+1)}),$$

where CRT refers to the Chinese restaurant table distribution described in Zhou and Carin [2015]. Marginalizing $\theta_{ik}^{(t+1)}$ from the Poisson distribution in (36) leads to $m_{ik}^{(t+1)} \sim \text{NB}(\theta_{ik}^{(t+2)}, p_{ik}^{(t+2)})$. Thus if (27) is true for layer t, then it is also true for layer $t+1$. □

Proof of Proposition 10. Since $\hat{X}$ and $X_\triangle$ are independent to each other, with (30) and (31), we have and $\mathbb{E}[X] = \mathbb{E}[\hat{X}] + \mu_\triangle = \mathbb{E}[\hat{X} + X_\triangle]$ and $\text{var}[X] = \text{var}[\hat{X}] + \sigma_\triangle^2 = \text{var}[\hat{X} + X_\triangle]$. Using Taylor series expansion, we have

$$\ln \mathbb{E}[e^{sX}] = -\sum_{k=1}^{\infty} a \ln(1 - s d_k^{-1}) = a \sum_{k=1}^{\infty}\sum_{n=1}^{\infty} \frac{s^n d_k^{-n}}{n} = s\mathbb{E}[X] + s^2 \frac{\text{var}[X]}{2} + a \sum_{n=3}^{\infty}\sum_{k=1}^{\infty} \frac{s^n d_k^{-n}}{n},$$

$$\ln \mathbb{E}[e^{s\hat{X}}] = -\sum_{k=1}^{K-1} a \ln(1 - sd_k^{-1}) = s\mathbb{E}[\hat{X}] + s^2 \frac{\mathrm{var}[\hat{X}]}{2} + a \sum_{n=3}^{\infty} \sum_{k=1}^{K-1} \frac{s^n d_k^{-n}}{n},$$

$$\ln \mathbb{E}[e^{sX_\triangle}] = -\frac{\mu_\triangle^2}{\sigma_\triangle^2} \ln\left(1 - s\frac{\sigma_\triangle^2}{\mu_\triangle}\right) = \frac{\mu_\triangle^2}{\sigma_\triangle^2} \sum_{n=1}^{\infty} \frac{s^n (\frac{\sigma_\triangle^2}{\mu_\triangle})^n}{n} = s\mu_\triangle + s^2 \frac{\sigma_\triangle^2}{2} + \frac{\mu_\triangle^2}{\sigma_\triangle^2} \sum_{n=3}^{\infty} \frac{s^n (\frac{\sigma_\triangle^2}{\mu_\triangle})^n}{n}.$$

The proof is completed with $\ln \mathbb{E}[e^{sX}] - \ln \mathbb{E}[e^{s(\hat{X}+X_\triangle)}] = \ln \mathbb{E}[e^{sX}] - \ln \mathbb{E}[e^{s(\hat{X})}] - \ln \mathbb{E}[e^{X_\triangle}]$. □

Proof of Proposition 11. For the Polya-Gamma random variable $X \sim \mathrm{PG}(a, c)$, since $\mathbb{E}[X] = \frac{a}{2|c|} \tanh\left(\frac{|c|}{2}\right)$ and $\lim_{|c|\to\infty} \tanh\left(\frac{|c|}{2}\right) = 1$, we have $\lim_{|c|\to\infty} \mathbb{E}[X] = 0$. With the expression of $\mathrm{var}[X]$ shown in (31), we have $\lim_{|c|\to\infty} \mathrm{var}[X] = 0$. Therefore, we have $X \to 0$ as $|c| \to \infty$.

Since $\lim_{|c|\to\infty} \mathbb{E}[|c|X] = a/2$ and

$$\mathrm{var}[|c|X] = (|c|)^2 \mathrm{var}[X] = \frac{a}{2|c|} \frac{\sinh(|c|) - |c|}{\cosh(|c|) + 1} = \frac{a}{2} \frac{\frac{\tanh(|c|)}{|c|} - \cosh^{-1}(|c|)}{1 + \cosh^{-1}(|c|)}.$$

we have $\lim_{|c|\to\infty} \mathrm{var}[|c|X] = 0$ and hence $\lim_{|c|\to\infty} |c|X = a/2$. □

Proof of Proposition 12. Using (30), we have

$$\mathbb{E}[X \mid a, 0] = \lim_{c\to 0} \mathbb{E}[X \mid a, c] = \lim_{c\to 0} \frac{a}{2|c|} \tanh\left(\frac{|c|}{2}\right) = \lim_{c\to 0} \frac{a}{2} \frac{e^{|c|} - 1}{|c|} \frac{1}{e^{|c|} + 1} = \frac{a}{4}.$$

Using (31), we have

$$\mathrm{var}[X \mid a, 0] = \lim_{c\to 0} \mathrm{var}[X \mid a, c] = \lim_{c\to 0} \frac{a \cosh^{-2}\left(\frac{|c|}{2}\right)}{4} \left(\frac{1}{6} + \sum_{n=1}^{\infty} \frac{|c|^{2n}}{(2n+3)!}\right) = \frac{a}{24}.$$ □

Proof of Proposition 13. Using (31), we have $\mathrm{var}[X \mid a, c] = \frac{a \cosh^{-2}\left(\frac{|c|}{2}\right)}{4} \left(\frac{1}{6} + \sum_{n=1}^{\infty} \frac{|c|^{2n}}{(2n+3)!}\right) \geq \frac{a}{24} \cosh^{-2}\left(\frac{|c|}{2}\right)$, with the equality holds if and only if $c = 0$. □

Proof of Proposition 14. With (30) and (31), the variance-to-mean ratio is

$$\frac{\mathrm{var}[X \mid a, c]}{\mathbb{E}[X \mid a, c]} = \frac{1}{|c|^2} - \frac{1}{|c| \sinh(|c|)} = \frac{\sinh(|c|) - |c|}{|c|^2 \sinh(|c|)} = \left[\frac{1}{6} \sum_{n=0}^{\infty} \frac{|c|^{2n+3}\, 3!}{(2n+3)!}\right] \Bigg/ \left[\sum_{n=0}^{\infty} \frac{|c|^{2n+3}}{(2n+1)!}\right].$$

When $c = 0$, with Proposition 12, we have $\mathrm{var}[X \mid a, c] / \mathbb{E}[X \mid a, c] = 1/6$. When $c \neq 0$, since $3!/(2n+3)! < 1/(2n+1)!$ a.s. for all $n \in \{1, \ldots\}$, we have

$$\sum_{n=0}^{\infty} \frac{|c|^{2n+3}\, 3!}{(2n+3)!} < \sum_{n=0}^{\infty} \frac{|c|^{2n+3}}{(2n+1)!}$$

a.s. and hence $\mathrm{var}[X \mid a, c] / \mathbb{E}[X \mid a, c] < 1/6$ a.s. □

C Gibbs sampling for sum-stack-softplus regression

For SS-softplus regression, Gibbs sampling via data augmentation and marginalization proceeds as follows.

Sample m_i***.*** Denote $\theta_{i\cdot} = \sum_{k=1}^{K} \theta_{ik}^{(1)}$. Since $m_i = 0$ a.s. given $y_i = 0$ and $m_i \geq 1$ given $y_i = 1$, and in the prior we have $m_i \sim \text{Pois}(\theta_{i\cdot})$, following the inference for the Bernoulli-Poisson link in Zhou [2015], we can sample m_i as

$$(m_i \,|\, -) \sim y_i \text{Pois}_+ \left(\theta_{i\cdot}\right), \tag{37}$$

where $m \sim \text{Pois}_+(\theta)$ denotes a draw from the truncated Poisson distribution, with PMF $f_M(m \,|\, y_i = 1, \theta) = (1 - e^{-\theta})^{-1} \theta^m e^{-\theta}/m!$, where $m \in \{1, 2, \ldots\}$. To draw truncated Poisson random variables, we use an efficient rejection sampler described in Zhou [2015], whose smallest acceptance rate, which happens when the Poisson rate is one, is 63.2%.

Sample $m_{ik}^{(1)}$***.*** Since letting $m_i = \sum_{k=1}^{K} m_{ik}^{(1)}$, $m_{ik}^{(1)} \sim \text{Pois}(\theta_{ik}^{(1)})$ is equivalent in distribution to letting $(m_{i1}^{(1)}, \ldots, m_{iK}^{(1)}) \,|\, m_i \sim \text{Mult}\left(m_i, \theta_{i1}^{(1)}/\theta_{i\cdot}, \ldots, \theta_{iK}^{(1)}/\theta_{i\cdot}\right)$, $m_i \sim \text{Pois}\left(\theta_{i\cdot}\right)$, similar to Dunson and Herring [2005] and Zhou et al. [2012a], we sample $m_{ik}^{(1)}$ as

$$(m_{i1}^{(1)}, \ldots, m_{iK}^{(1)} \,|\, -) \sim \text{Mult}\left(m_i, \theta_{i1}^{(1)}/\theta_{i\cdot}, \ldots, \theta_{iK}^{(1)}/\theta_{i\cdot}\right). \tag{38}$$

Sample $m_{ik}^{(t)}$ ***for*** $t \geq 2$***.*** As in Theorem 8's proof, we sample $m_{ik}^{(t)}$ for $t = 2, \ldots, T+1$ as

$$(m_{ik}^{(t)} \,|\, m_{ik}^{(t-1)}, \theta_{ik}^{(t)}) \sim \text{CRT}\big(m_{ik}^{(t-1)},\ \theta_{ik}^{(t)}\big). \tag{39}$$

Sample $\boldsymbol{\beta}_k^{(t)}$***.*** Using data augmentation for NB regression, as in Zhou et al. [2012b] and Polson et al. [2013], we denote $\omega_{ik}^{(t)}$ as a random variable drawn from the Polya-Gamma (PG) distribution [Polson and Scott, 2011] as $\omega_{ik}^{(t)} \sim \text{PG}\left(m_{ik}^{(t-1)} + \theta_{ik}^{(t)},\ 0\right)$, under which we have $\mathbb{E}_{\omega_{ik}^{(t)}}\left[\exp(-\omega_{ik}^{(t)}(\psi_{ik}^{(t)})^2/2)\right] = \cosh^{-(m_{ik}^{(t-1)}+\theta_{ik}^{(t)})}(\psi_{ik}^{(t)}/2)$. Thus the likelihood of $\psi_{ik}^{(t)} := \boldsymbol{x}_i \boldsymbol{\beta}_k^{(t)} + \ln q_{ik}^{(t-1)} = \ln\left(e^{q_{ik}^{(t)}} - 1\right)$ in (27) can be expressed as

$$\begin{aligned}
\mathcal{L}(\psi_{ik}^{(t)}) &\propto \frac{\left(e^{\psi_{ik}^{(t)}}\right)^{m_{ik}^{(t-1)}}}{\left(1 + e^{\psi_{ik}^{(t)}}\right)^{m_{ik}^{(t-1)}+\theta_{ik}^{(t)}}} = \frac{2^{-(m_{ik}^{(t-1)}+\theta_{ik}^{(t)})} \exp\left(\frac{m_{ik}^{(t-1)}-\theta_{ik}^{(t)}}{2}\psi_{ik}^{(t)}\right)}{\cosh^{m_{ik}^{(t-1)}+\theta_{ik}^{(t)}}(\psi_{ik}^{(t)}/2)} \\
&\propto \exp\left(\frac{m_{ik}^{(t-1)} - \theta_{ik}^{(t)}}{2}\psi_i\right) \mathbb{E}_{\omega_{ik}^{(t)}}\left[\exp[-\omega_{ik}^{(t)}(\psi_{ik}^{(t)})^2/2]\right].
\end{aligned}$$

Combining the likelihood $\mathcal{L}(\psi_{ik}^{(t)}, \omega_{ik}^{(t)}) \propto \exp\left(\frac{m_{ik}^{(t-1)}-\theta_{ik}^{(t)}}{2}\psi_i\right)\exp[-\omega_{ik}^{(t)}(\psi_{ik}^{(t)})^2/2]$ and the prior, we sample auxiliary Polya-Gamma random variables $\omega_{ik}^{(t)}$ as

$$(\omega_{ik}^{(t)} \,|\, -) \sim \mathrm{PG}\left(m_{ik}^{(t-1)} + \theta_{ik}^{(t)},\ \boldsymbol{x}_i'\boldsymbol{\beta}_k^{(t)} + \ln q_{ik}^{(t-1)}\right), \tag{40}$$

conditioning on which we sample $\boldsymbol{\beta}_k^{(t)}$ as

$$(\boldsymbol{\beta}_k^{(t)} \,|\, -) \sim \mathcal{N}(\boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}), \quad \boldsymbol{\Sigma}_k^{(t)} = \left(\mathrm{diag}(\alpha_{0tk}, \ldots, \alpha_{Vtk}) + \sum\nolimits_i \omega_{ik}^{(t)} \boldsymbol{x}_i \boldsymbol{x}_i'\right)^{-1},$$
$$\boldsymbol{\mu}_k^{(t)} = \boldsymbol{\Sigma}_k^{(t)}\left[\sum\nolimits_i \left(-\omega_{ik}^{(t)} \ln q_{ik}^{(t-1)} + \frac{m_{ik}^{(t-1)} - \theta_{ik}^{(t)}}{2}\right)\boldsymbol{x}_i\right]. \tag{41}$$

Once we update $\boldsymbol{\beta}_k^{(t)}$, we calculate $q_{ik}^{(t)}$ using (25). To draw Polya-Gamma random variables, we use the approximate sampler described in Proposition 10, which is unbiased in both its mean and its variance. The approximate sampler is found to be highly accurate even for a truncation level as small as one, for various combinations of the two Polya-Gamma parameters. Unless stated otherwise, we set the truncation level of drawing a Polya-Gamma random variable as six, which means the summation of six independent gamma random variables is used to approximate a Polya-Gamma random variable.

Sample $\theta_{ik}^{(t)}$**.** Using the gamma-Poisson conjugacy, we sample $\tau_{ik}^{(t)} := \theta_{ik}^{(t)} q_{ik}^{(t)}$ as

$$(\tau_{ik}^{(t)} \,|\, -) \sim \mathrm{Gamma}\left(\theta_{ik}^{(t+1)} + m_{ik}^{(t)},\, 1 - e^{-q_{ik}^{(t+1)}}\right). \tag{42}$$

Sample α_{vtk}**.** We sample α_{vtk} as

$$(\alpha_{vtk} \,|\, -) \sim \mathrm{Gamma}\left(e_0 + \frac{1}{2}, \frac{1}{f_0 + \frac{1}{2}(\beta_{vk}^{(t)})^2}\right). \tag{43}$$

Sample c_0**.** We sample c_0 as

$$(c_0 \,|\, -) \sim \mathrm{Gamma}\left(e_0 + \gamma_0, \frac{1}{f_0 + \sum_k r_k}\right). \tag{44}$$

Sample γ_0 ***and*** r_k**.** Let us denote

$$\tilde{p}_k := \sum\nolimits_i q_{ik}^{(T+1)} \Big/ \big(c_0 + \sum\nolimits_i q_{ik}^{(T+1)}\big).$$

Given $l_{\cdot k} = \sum_i m_{ik}^{(T+1)}$, we first sample

$$(\tilde{l}_k \,|\, -) \sim \mathrm{CRT}(l_{\cdot k}, \gamma_0/K). \tag{45}$$

With these latent counts, we then sample γ_0 and r_k as

$$\begin{aligned}
(\gamma_0 \,|\, -) &\sim \mathrm{Gamma}\left(a_0 + \tilde{l}_{\cdot}, \, \frac{1}{b_0 - \frac{1}{K}\sum_k \ln(1-\tilde{p}_k)}\right), \\
(r_k \,|\, -) &\sim \mathrm{Gamma}\left(\frac{\gamma_0}{K} + l_{\cdot k}, \, \frac{1}{c_0 + \sum_i q_{ik}^{(T+1)}}\right).
\end{aligned} \tag{46}$$

C.1 Numerical stability

For stack-softplus and SS-softplus regressions with $T > 1$, if for some data point $\boldsymbol{x}_i$, the inner product $\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(t)}$ takes such a large negative number that $e^{-\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(t)}} = 0$ under a finite numerical precision, then $q_{ik}^{(t)} = 0$ and $\ln q_{ik}^{(t)} = -\infty$. For example, in both 64 bit Matlab (version R2015a) and 64 bit R (version 3.0.2), if $\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(t)} \leq -745.2$, then $e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k^{(t)}} = 0$ and hence $q_{ik}^{(t)} = 0$, $p_{ik}^{(t)} = 0$, and $\ln q_{ik}^{(t)} = -\infty$.

If $q_{ik}^{(t)} = 0$, then with (40), we let $\omega_{ik}^{(t+1)} = 0$, and with Proposition 11, we let

$$-\omega_{ik}^{(t+1)} \ln q_{ik}^{(t)} + \frac{m_{ik}^{(t)} - \theta_{ik}^{(t+1)}}{2} = \frac{m_{ik}^{(t)} + \theta_{ik}^{(t+1)}}{2} + \frac{m_{ik}^{(t)} - \theta_{ik}^{(t+1)}}{2} = m_{ik}^{(t)},$$

and with (25), we let $q_{ik}^{(\tilde{t})} = 0$ for all $\tilde{t} \geq t$. Note that if $q_{ik}^{(t)} = 0$, drawing $\omega_{ik}^{(\tilde{t})}$ for $\tilde{t} \in \{t+1, \ldots, T+1\}$ becomes unnecessary. To avoid the numerical issue of calculating $\theta_{ik}^{(t)}$ with $\tau_{ik}^{(t)}/q_{ik}^{(t)}$ when $q_{ik}^{(t)} = 0$, we let

$$\theta_{ik}^{(t)} = \tau_{ik}^{(t)} / \max\{\epsilon, q_{ik}^{(t)}\}, \tag{47}$$

where we set $\epsilon = 10^{-10}$ to for illustrations and $\epsilon = 10^{-6}$ to produce the results in the tables. To ensure that the covariance matrix for $\boldsymbol{\beta}_k^{(t)}$ is positive definite, we bound α_{vtk} above 10^{-3}.

C.2 The propagation of latent counts across layers

As the number of tables occupied by the customers is in the same order as the logarithm of the number of customers in a Chinese restaurant process, $m_{ik}^{(t+1)}$ in (28) is in the same order as $\ln\left(m_{ik}^{(t)}\right)$ and hence often quickly decreases as t increases, especially when t is small. In addition, since $m_{ik}^{(t+1)} \leq m_{ik}^{(t)}$ almost surely (a.s.), $m_{ik}^{(t)} = 0$ a.s. if $m_{ik}^{(1)} = 0$, $m_{ik}^{(t)} \geq 1$ a.s. if $m_{ik}^{(1)} \geq 1$, and $m_i \geq 1$ a.s. if $y_i = 1$, we have the following two corollaries.

Corollary 15. *The latent count $m_{\cdot k}^{(t)} = \sum_i m_{ik}^{(t)}$ monotonically decreases as t increases and $m_{\cdot k}^{(t)} \geq \sum_i \delta(m_{ik}^{(1)} \geq 1)$.*

Corollary 16. *The latent count* $m_{\cdot\cdot}^{(t)} = \sum_k m_{\cdot k}^{(t)}$ *monotonically decreases as* t *increases and* $m_{\cdot\cdot}^{(t)} \geq \sum_i \delta(y_i = 1)$.

With Corollary 16, one may consider using the values of $m_{\cdot\cdot}^{(t)} / \sum_i \delta(y_i = 1)$ to decide whether T, the depth of the gamma belief network used in SS-softplus regression, need to be increased to increase the model capacity, or whether T could be decreased to reduce the computational complexity. Moreover, with Corollary 15, one may consider using the values of $m_{\cdot k}^{(t)} / \sum_i \delta(y_i = 1)$ to decide how many criteria would be sufficient to equip each individual expert. For simplicity, we consider the number of criteria for each expert as a parameter that determines the model capacity and we fix it as T for all experts in this paper.

D Related Methods and Discussions

While we introduce a novel nonlinear regression framework for binary response variables, we recognize some interesting connections with previous work, including the gamma belief network, several binary classification algorithms that use multiple hyperplanes, and the ideas of using the mixture or product of multiple probability distributions to construct a more complex predictive distribution, as discussed below.

D.1 Gamma belief network

The Poisson gamma belief network is proposed in Zhou et al. [2015a] to construct a deep Poisson factor model, in which the shape parameters of the gamma distributed factor score matrix at layer t are factorized under the gamma likelihood into the product of a factor loading matrix and a gamma distributed factor score matrix at layer $t + 1$. While the scale parameters of the gamma distributed factor scores depend on the indices of data samples, they are constructed to be independent of the indices of the latent factors, making it convenient to derive closed-form Gibbs sampling update equations via data augmentation and marginalization. The gamma belief networks in both stack- and SS-softplus regressions, on the other hand, do not factorize the gamma shape parameters but parameterize the logarithm of each gamma scale parameter using the inner product of the corresponding covariate and regression coefficient vectors. Hence a gamma scale parameter in softplus regressions depends on both the index of the data sample and that of the corresponding latent expert. On a related note, while the gamma distribution function is the building unit for both gamma belief networks, the one in Zhou et al. [2015a] is used to factorize the Poisson rates of the observed or latent high-dimensional count vectors, extracting multilayer deep representations in an unsupervised manner, whereas the one used in (11) is designed for supervised learning to establish a direct functional relationship to predict a label given its covariates, without introducing factorization within the gamma belief network.

D.2 Multi-hyperplane regression models

Generalizing the construction of multiclass support vector machines in Crammer and Singer [2002], the idea of combining multiple hyperplanes to define nonlinear binary classification decision boundary has been discussed in Aiolli and Sperduti [2005], Wang et al. [2011], Manwani and Sastry [2010, 2011], and Kantchelian et al. [2014]. In particular, Kantchelian et al. [2014] clearly connects the idea of combining multiple hyperplanes for nonlinear classification with the learning of a convex polytope, defined by the intersection of multiple hyperplanes, to separate one class from the other, and shows that a convex polytope classifier can provide larger margins than a linear classifier equipped with a single hyperplane.

From this point of view, the proposed sum-softplus regression is closely related to the convex polytope machine (CPM) of Kantchelian et al. [2014] as its decision boundary can be explicitly bounded by a convex polytope that encloses negative examples, as described in Theorem 2 and illustrated in Fig. 1. Distinct from the CPM that uses a convex polytope as its decision boundary, and provides no probability estimates for class labels and no principled ways to set its number of equally-weighted hyperplanes, sum-softplus regression makes its decision boundary smoother than the corresponding bounding convex polytope, as shown in Figs. 1 (c)-(d), using more complex interactions between hyperplanes than simple intersection, provides probability estimates for its labels, and supports countably infinite differently-weighted hyperplanes with the gamma-negative binomial process.

In addition, to solve the objective function that is non-convex, the CPM relies on heuristics to hard assign a positively labeled data point to one and only one of the hyperplanes, making the learning of the parameters for each hyperplane become a convex optimization problem, whereas all softplus regressions use Bayesian inference with closed-form Gibbs sampling update equations, in which each data point is assigned to one or multiple hyperplanes to learn their parameters. Moreover, distinct from the CPM and sum-softplus regression that use either a single convex polytope or a single convex-polytope bounded space to enclose *negative* examples, the proposed stack-softplus regression defines a single convex-polytope-like confined space to enclose *positive* examples, and the proposed SS-softplus regression further generalizes all of them in that its decision boundary is related to the union of multiple convex-polytope-like confined spaces.

D.3 Mixture, product, convolution, and stack of experts

With each regression coefficient vector analogized as an expert, the proposed softplus regressions can also be related to the idea of combining multiple experts' beliefs to improve a model's predictive performance. Conventionally, if an expert's belief is expressed as a probability density/mass function, then one may consider using the linear opinion pool [Stone, 1961] or logarithmic opinion pool [Genest and Zidek, 1986, Heskes, 1998] to aggregate multiple experts' probability distributions into a single one. To reach a single aggregated distribution of some unknown quantity y, the linear opinion pool, also known as mixture of experts (MoE), aggregates experts additively by taking a weighted average of their distributions on y, while the logarithmic opinion pool, including the product of experts (PoE) of Hinton [2002] as a special case, aggregates them multiplicatively by taking a weighted geometric mean of these distributions [Clemen and Winkler, 1999, Jacobs, 1995].

Opinion pools with separately trained experts can also be related to ensemble methods [Hastie et al., 2001, Zhou, 2012], including both bagging [Breiman, 1996] and boosting [Freund and Schapire, 1997]. Another common strategy is to jointly train the experts using the same set of features and data. For example, the PoE of Hinton [2002] trains its equal-weighted experts jointly on the same data. The proposed softplus regressions follow the second strategy to jointly train on the same data not only its experts but also their weights.

Assume there are K experts and the kth expert's belief on y is expressed as a probability density/mass function $f_k(y \,|\, \theta_k)$, where θ_k represents the distribution parameters. The linear opinion pool aggregates the K expert distributions into a single one using

$$f_Y(y \,|\, \{\theta_k\}_k) = \sum_{k=1}^{K} \pi_k f_k(y \,|\, \theta_k), \tag{48}$$

where π_k are nonnegative weights and sum to one. The logarithmic opinion pool aggregates the K expert distributions using

$$f_Y(y \,|\, \{\theta_k\}_k) = \frac{\prod_{k=1}^{K} [f_k(y \,|\, \theta_k)]^{\pi_k}}{\sum_y \prod_{k=1}^{K} [f_k(y \,|\, \theta_k)]^{\pi_k}}, \tag{49}$$

where $\pi_k \geq 0$ and the constraint $\sum_{k=1}^{K} \pi_k = 1$ is also commonly imposed. If $\pi_k = 1$ for all k, then a logarithmic opinion pool becomes a product of experts (PoE).

In decision and risk analysis, the functions $f_k(y)$ usually represent independent experts' subjective probabilities, which are often assumed to be known *a priori*, and the focus is to optimize the expert weights [Clemen and Winkler, 1999]. Whereas in statistics and machine learning, both the functions $f_k(y)$ and the expert weights are typically learned from the data.

One common strategy is to first train different experts separately, such as using different feature sets, different data, and different learning algorithms, and subsequently aggregate their distributions into a single one. For example, in Tax et al. [2000], a set of classifiers are first separately trained on different independent data sources and then aggregated additively or multiplicatively to construct a single classifier.

D.3.1 Mixture of experts

A linear opinion pool is also commonly known as a mixture of experts (MoE). Not only are there efficient algorithms, using expectation-maximization (EM) or MCMC, to jointly learn the experts (mixture components) and their mixture weights in a MoE, there are nonparametric Bayesian algorithms, such as Dirichlet process mixture models [Antoniak, 1974, Rasmussen, 2000], that support a potentially infinite number of experts. We note it is possible to combine the proposed softplus regression models with linear opinion pool to further improve their performance. We leave that extension for future study.

D.3.2 Product of experts

In contrast to MoEs, the logarithmic opinion pool could produce a probability distribution with sharper boundaries, but at the same time is usually much more challenging to train due to a normalization constant that is typically intractable to compute. Hinton's product of experts (PoE) is one of the most well-know logarithmic opinion pools; since jointly training the experts and their weights in the logarithmic opinion pool makes the inference much more difficult, all the experts in a PoE are weighted equally [Hinton, 2002]. A PoE is distinct from previously proposed logarithmic opinion pools in that its experts are trained jointly rather than separately. Even with the restriction of equal weights, the exact gradients of model parameters in a PoE are often intractable to compute, and hence contrastive divergence that approximately computes these gradients is commonly employed for approximate maximum likelihood inference. Moreover, no nonparametric Bayesian prior is available to allow the number of experts in a PoE to be automatically inferred. PoEs have been successfully applied to binary image modeling [Hinton, 2002] and one of its special forms, the restricted Boltzmann machine, was widely used as a basic building block in constructing a deep neural network [Bengio et al., 2007, Hinton et al., 2006]. PoEs for non-binary data and several other logarithmic opinion pools inspired by PoEs have also been proposed, with applications to image analysis, information retrieval, and computational linguistics [Smith et al., 2005, Welling et al., 2004, Xing et al., 2005]. We note one may apply the proposed softplus regressions to regress the binary response variables on the covariates transformed by PoEs, such as restricted Boltzmann machine and its deep constructions, to further improve the classification performance. We leave that extension for future study.

D.3.3 Stack of experts

Different from both MoE and PoE, we propose stack of experts (SoE) that repeatedly mixes the same distribution with respect to the same distribution parameter as

$$x \sim f(\,r_1, w_1), \ldots,\; r_{k-1} \sim f(r_k, w_k), \ldots,\; r_{K-1} \sim f(r_K, w_K).$$

In a SoE, the marginal distribution of x given r and $\{w_k\}_{1,K}$ can be expressed as

$$f(x \,|\, r_K, \{w_k\}_{1,K}) = \int \ldots \int f(x \,|\, r_1, w_1) f(r_1 \,|\, r_2, w_2) \ldots f(r_{K-1} \,|\, r_K, w_K) dr_{K-1} \ldots dr_1,$$

where the parameter w_k that is pushed into the stack after w_{k-1} will pop out before w_{k-1} to parameterize the marginal distribution. In both stack- and SS-softplus regressions, we obtain a stacked gamma distribution $x \sim f(r_K, \{w_k\}_{1,K})$ by letting $f(x_k \,|\, r_k, w_k) = \text{Gamma}(x_k; r_k, e^{w_k})$, and as shown in (10) and (12), the regression coefficient vectors that are pushed into the stack later appear earlier in both the stack- and SS-softplus functions.

D.3.4 Convolution of experts

Distinct from both MoE and PoE, we may consider that both sum- and SS-softplus regressions use a convolution of experts (CoE) strategy to aggregate multiple experts' beliefs by convolving their probability distributions into a single one. A CoEs is based on a fundamental law in probability and statistics: the probability distribution of independent random variables' summation is equal to the convolution of their probability distributions [*e.g.*, Fristedt and Gray, 1997]. Thus even though it is possible that the convolution is extremely difficult to solve and hence the explicit form of the aggregated probability density/mass function might not be available, simply adding together the random samples independently drawn from a CoE's experts would lead to a random sample drawn from the aggregated distribution that is smoother than any of the distributions used in convolution.

In a general setting, denoting $G = \sum_{k=1}^{\infty} r_k \delta_{\omega_k}$ as a draw from a completely random measure [Kingman, 1993] that consists of countably infinite atoms, one may construct an infinite CoE model to generate random variables from a convolved distribution as $x = \sum_{k=1}^{\infty} x_k,\; x_k \sim f_k$, where $f_k = f(r_k, \omega_k)$ are independent experts parameterized by both the weights r_k and atoms ω_k of G. Denoting $(f_i * f_j)(x) := \int f_i(\tau) f_j(x-\tau) d\tau$ as the convolution operation, under the infinite CoE model, we have

$$f_X(x) = (f_1 * f_2 * \ldots * f_\infty)(x),$$

where the same distribution function is repeatedly convolved to increase the representation power to better fit complex data.

Under this general framework, we may consider both sum- and SS-softplus regressions as infinite CoEs, with the gamma process used as the underlying completely random measure Kingman [1967, 1993] to support countably infinite differently weighted probability distributions for convolution. For sum-softplus regression, as shown in (9) of Theorem 2, each expert can be considered as a gamma distribution whose scale parameter is parameterized by the inner product of the covariate vector and an expert-specific regression coefficient vector, and the convolution of countably infinite experts' gamma distributions is used as the distribution of the BerPo rate of the response variable; and alternatively, as shown in (14) of Theorem 2, each expert can be considered as a NB regression, and the convolution of countably infinite experts' NB distributions is used as the distribution of the latent count response variable. For SS-softplus regression, each expert can be considered as a stacked gamma distribution, and the convolution of countably infinite experts' stacked gamma distributions is used as the distribution of the BerPo rate of a response variable.

Related to the PoE of Hinton [2002], a CoE trains its experts jointly on the same data. Distinct from that, a CoE does not have an intractable normalization constant in the aggregated distribution, its experts can be weighted differently, and its number of experts could be automatically inferred from the data in a nonparametric Bayesian manner. The training for a CoE is also unique, as inferring the parameters of each expert essentially corresponds to deconvolving the aggregated distribution. Moreover, the convolution operation ensures that the aggregated distribution is smoother than every expert distribution. For inference, while it is often challenging to analytically deconvolve the convolved distribution function, we consider first constructing a hierarchical Bayesian model that can generate random variables from the convolved distribution, and then developing a MCMC algorithm to decompose the total sum x into the x_k of individual experts, which are then used to infer the model parameters r_k and ω_k for each expert.

E Experimental settings and additional results

We use the L_2 regularized logistic regression provided by the LIBLINEAR package [Fan et al., 2008] to train a linear classifier, where a bias term is included and the regularization parameter C is five-fold cross-validated on the training set from $(2^{-10}, 2^{-9}, \ldots, 2^{15})$.

For kernel SVM, a Gaussian RBF kernel is used and three-fold cross validation is used to tune both the regularization parameter C and kernel width on the training set. We use the LIBSVM package [Chang and Lin, 2011], where we three-fold cross-validate both the regularization parameter C and kernel-width parameter γ on the training set from $(2^{-5}, 2^{-4}, \ldots, 2^{5})$, and choose the default settings for all the other parameters. Following Chang et al. [2010], for the ijcnn1 dataset, we choose $C = 32$ and $\gamma = 2$, and for the a9a dataset, we choose $C = 8$ and $\gamma = 0.03125$.

For RVM, instead of directly quoting the results from Tipping [2001], which only reported the mean but not standard deviation of the classification errors for each of the first six datasets in Tab. 3, we use the matlab code[1] provided by the author, using a Gaussian RBF kernel whose kernel width is three-fold cross-validated on the training set from $(2^{-5}, 2^{-4.5}, \ldots, 2^{5})$ for both ijcnn1 and a9a and from $(2^{-10}, 2^{-9.5}, \ldots, 2^{10})$ for all the others.

We consider adaptive multi-hyperplane machine (AMM) of Wang et al. [2011], as implemented in the BudgetSVM[2] (Version 1.1) software package [Djuric et al., 2013]. We consider the batch version of the algorithm. Important parameters of the AMM include both the regularization parameter λ and training epochs E. As also observed in Kantchelian et al. [2014], we do not observe the testing errors of AMM to strictly decrease as E increases. Thus, in addition to cross validating the regularization parameter λ on the training set from $\{10^{-7}, 10^{-6}, \ldots, 10^{-2}\}$, as done in Wang et al. [2011], for each λ, we try $E \in \{5, 10, 20, 50, 100\}$ sequentially until the cross-validation error begins to decrease, *i.e.*, under the same λ, we choose $E = 20$ if the cross-validation error of $E = 50$ is greater than that of $E = 20$. We use the default settings for all the other parameters.

We consider the convex polytope machine (CPM) of [Kantchelian et al., 2014], using the python code[3] provided by the authors. Important parameters of the CPM include the entropy parameter h, regularization factor C, and number of hyperplanes K for each side of the CPM ($2K$ hyperplanes in total). Similar to the setting of [Kantchelian et al., 2014], we first fix $h = 0$ and select the best regularization factor C from $\{10^{-4}, 10^{-3}, \ldots, 10^{0}\}$ using three-fold cross validation on the training set. For each C, we try $K \in \{1, 3, 5, 10, 20, 40, 60, 80, 100\}$ sequentially until the cross-validation error begins to decrease. With both λ and K selected, we then select h from $\{0, \ln(K/10), \ln(2K/10), \ldots, \ln(9K/10)\}$. For each trial, we consider 10 million iterations in cross-validation and 32 million iterations in training with the cross-validated parameters. Note different from Kantchelian et al. [2014], which suggests that the error rate decreases as K increases, we cross-validate K as we have found that the testing errors of the CPM may increase once it increases over certain limits.

[1] `http://www.miketipping.com/downloads/SB2_Release_200.zip`

[2] `http://www.dabi.temple.edu/budgetedsvm/`

[3] `https://github.com/alkant/cpm`

Algorithm 1 Upward-downward Gibbs sampling for sum-stack-softplus (SS-softplus) regression.
Inputs: y_i: the observed labels, $\boldsymbol{x}_i$: covariate vectors, $K_{\max}$: the upper-bound of the number of experts, T: the number of criteria of each expert, I_{Prune}: the set of iterations at which the operation of deactivating experts is performed, and the model hyper-parameters.
Outputs: KT regression coefficient vectors $\boldsymbol{\beta}_k^{(t)}$ and K weights r_k, where $K \leq K_{\max}$ is the total number of active experts that are associated with nonzero latent counts.

```
Initialize the model parameters with $\boldsymbol{\beta}_k^{(t)} = 0$ and $r_k = 1/K_{\max}$.
for $iter = 1 : maxIter$ do Gibbs sampling
    parfor $k = 1, \dots, K_{\max}$ do Downward sampling
        for $t = T, T-1 \dots, 1$ do
            Sample $\theta_{ik}^{(t)}$ if Expert $k$ is active ;
        end for
    end parfor
    Sample $m_i$ ; Sample $\{m_{ik}^{(1)}\}_k$ ;
    parfor $k = 1, \dots, K_{\max}$ do Upward sampling
        if Expert $k$ is active then
            for $t = 2, 3, \dots, T+1$ do
                Sample $m_{ik}^{(t)}$ ; Sample $\omega_{ik}^{(t)}$ ; Sample $\boldsymbol{\beta}_k^{(t)}$ and Calculate $p_{ik}^{(t)}$ and $q_{ik}^{(t)}$ ;
            end for
        end if
        Deactivate Expert $k$ if $iter \in I_{Prune}$ and $m_{\cdot k}^{(1)} = 0$ ;
    end parfor
    Sample $\gamma_0$ and $c_0$ ;
    Sample $r_1, \dots, r_{K_{\max}}$ ;
end for
```

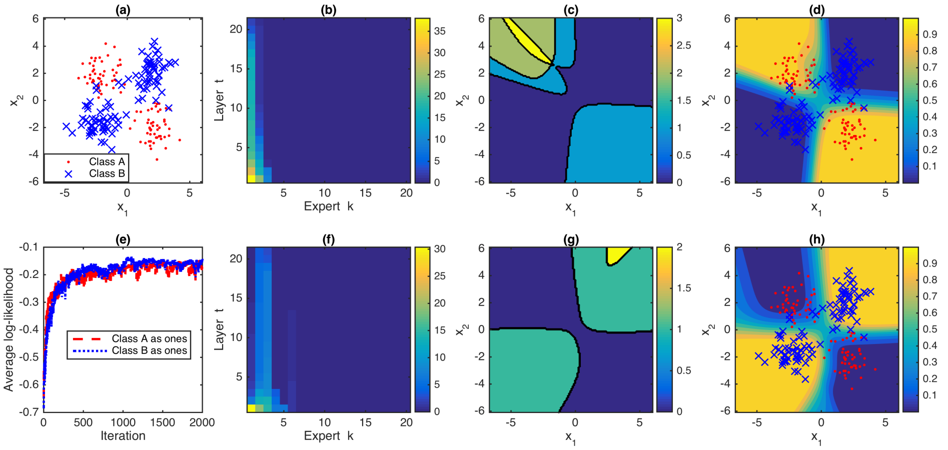

Figure 12: Analogous figure to to Fig. 7 for SS-softplus regression for a different dataset with two classes, where Class A consists of 50 data points (x_i, y_i) centering around $(-2, 2)$, where $x_i \sim \mathcal{N}(-2, 1)$ and $y_i \sim \mathcal{N}(2, 1)$, and another 50 such kind of data points centering around $(2, -2)$, and Class B consists of 50 such kind of data points centering around $(2, 2)$ and another 50 such kind of data points centering around $(-2, -2)$.

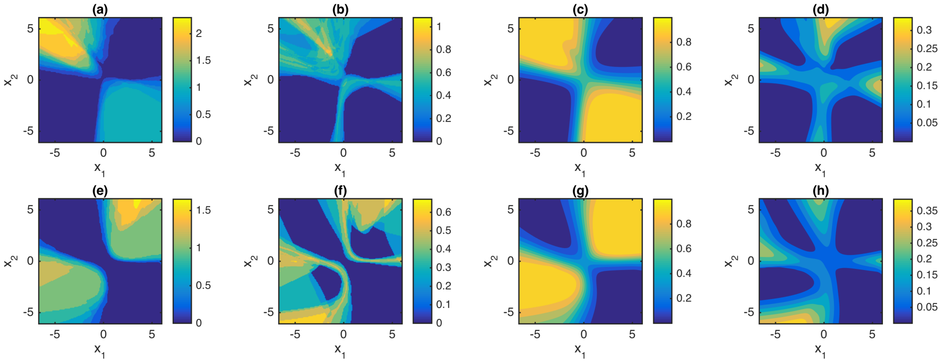

Figure 13: Analogous figure to Fig. 7, with the same experimental setting used for Fig. 12.

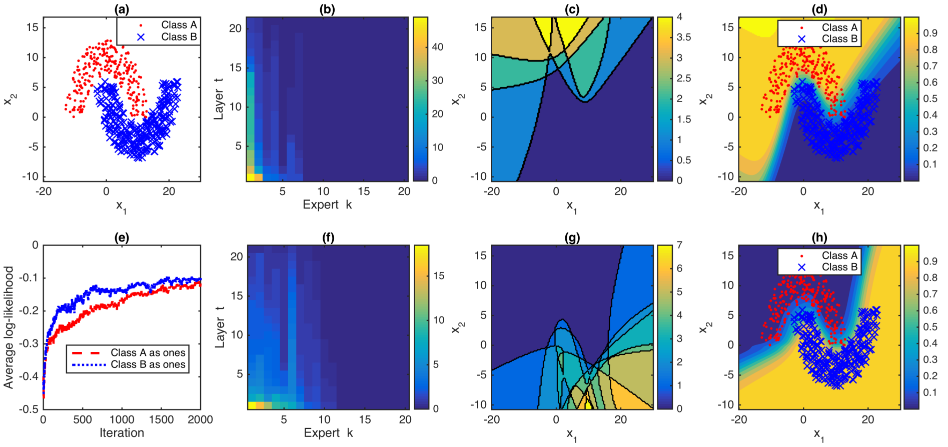

Figure 14: Analogous figure to Fig. 7 for SS-softplus regression for a double moon dataset, where both Classes A and B consist of 250 data points (x_i, y_i).

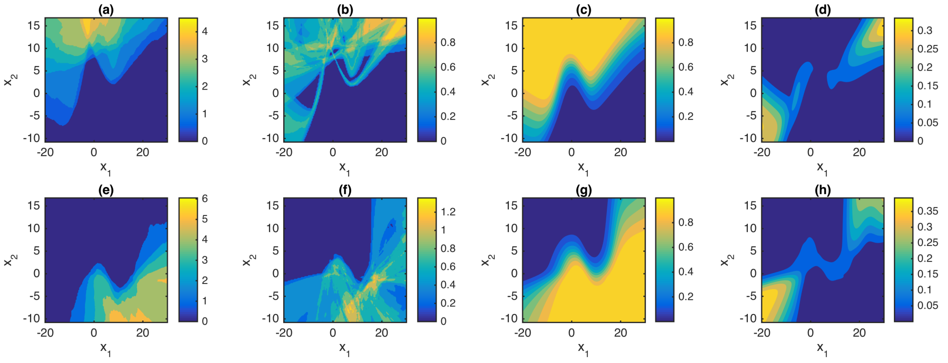

Figure 15: Analogous figure to Fig. 8, with the same experimental setting used for Fig. 14.

Table 3: Binary classification datasets used in experiments, where V is the feature dimension.

Dataset	banana	breast cancer	titanic	waveform	german	image	ijcnn1	a9a
Train size	400	200	150	400	700	1300	49,990	32,561
Test size	4900	77	2051	4600	300	1010	91,701	16,281
V	2	9	3	21	20	18	22	123

Table 4: Performance of stack-softplus regression with the depth set as $T \in \{1, 2, 3, 5, 10\}$, where stack-softplus regression with $T = 1$ reduces to softplus regression.

Dataset	softplus	stack-ς (T=2)	stack-ς (T=3)	stack-ς (T=5)	stack-ς (T=10)
banana	47.87 ± 4.36	34.66 ± 5.58	32.19 ± 4.76	33.21 ± 5.76	$\mathbf{30.67} \pm 4.23$
breast cancer	28.70 ± 4.76	29.35 ± 2.31	29.48 ± 4.94	$\mathbf{27.92} \pm 3.31$	28.31 ± 4.36
titanic	22.53 ± 0.43	22.80 ± 0.59	22.48 ± 0.55	22.71 ± 0.70	22.84 ± 0.54
waveform	13.62 ± 0.71	12.52 ± 1.14	$\mathbf{12.23} \pm 0.79$	12.25 ± 0.69	12.33 ± 0.65
german	24.07 ± 2.11	23.73 ± 1.99	23.67 ± 1.89	$\mathbf{22.97} \pm 2.22$	23.80 ± 1.64
image	17.55 ± 0.75	9.11 ± 0.99	8.39 ± 1.05	7.97 ± 0.52	$\mathbf{7.50} \pm 1.17$
Mean of SVM normalized errors	2.485	1.773	1.686	1.665	1.609

Table 5: Performance of SS-softplus regression with $K_{\max} = 20$ and the depth set as $T \in \{1, 2, 3, 5, 10\}$, where SS-softplus regression with $T = 1$ reduces to sum-softplus regression.

Dataset	sum-ς	SS-ς (T=2)	SS-ς (T=3)	SS-ς (T=5)	SS-ς (T=10)
banana	30.78 ± 8.68	15.00 ± 5.31	12.54 ± 1.18	$\mathbf{11.89} \pm 0.61$	11.93 ± 0.59
breast cancer	30.13 ± 4.23	29.74 ± 3.89	30.39 ± 4.94	28.83 ± 3.40	$\mathbf{28.44} \pm 4.60$
titanic	22.48 ± 0.25	22.56 ± 0.65	22.42 ± 0.45	22.29 ± 0.80	$\mathbf{22.20} \pm 0.48$
waveform	11.51 ± 0.65	11.41 ± 0.96	$\mathbf{11.34} \pm 0.70$	11.69 ± 0.69	12.92 ± 1.00
german	23.60 ± 2.39	$\mathbf{23.30} \pm 2.54$	$\mathbf{23.30} \pm 2.20$	24.23 ± 2.46	23.90 ± 1.50
image	3.50 ± 0.73	2.76 ± 0.47	$\mathbf{2.59} \pm 0.47$	2.73 ± 0.53	2.93 ± 0.46
Mean of SVM normalized errors	1.370	1.079	1.033	1.033	1.059

Table 6: Analogous table to Tab. 5 for the number of inferred experts (hyperplanes).

Dataset	sum-ς	SS-ς (T=2)	SS-ς (T=3)	SS-ς (T=5)	SS-ς (T=10)
banana	3.70 ± 0.95	5.70 ± 0.67	6.80 ± 0.79	7.60 ± 1.17	9.80 ± 2.39
breast cancer	3.10 ± 0.74	4.10 ± 0.88	5.70 ± 1.70	6.40 ± 1.43	9.50 ± 1.51
titanic	2.30 ± 0.48	3.30 ± 0.82	3.80 ± 0.92	4.00 ± 0.94	6.20 ± 1.23
waveform	4.40 ± 0.84	6.20 ± 1.62	7.00 ± 2.21	8.90 ± 2.33	11.50 ± 2.72
german	6.70 ± 0.95	9.80 ± 1.48	11.10 ± 2.64	14.70 ± 1.77	20.00 ± 2.40
image	11.20 ± 1.32	13.20 ± 2.30	14.60 ± 2.07	17.60 ± 1.90	21.40 ± 2.22
Mean of SVM normalized K	0.030	0.041 (×2)	0.048 (×3)	0.057 (×5)	0.077 (×10)

Table 7: Comparison of classification errors of logistic regression (LR), support vector machine (SVM), adaptive multi-hyperplane machine (AMM), convex polytope machine (CPM), softplus regression, sum-softplus (sum-ς) regression with $K_{\max} = 20$, stack-softplus (stack-ς) regression with $T = 5$, and SS-softplus regression with $K_{\max} = 20$ and $T = 5$.

Dataset	LR	SVM	RVM	AMM	CPM	softplus	sum-ς	stack-ς (T=5)	SS-ς (T=5)
ijcnn1	8.00	1.30	**1.29**	2.06 ±0.27	2.57 ±0.17	8.41 ±0.03	3.39 ±0.17	6.43 ±0.15	2.24 ±0.12
a9a	15.00	**14.88**	14.95	15.03 ±0.17	15.08 ±0.07	15.02 ±0.06	**14.88** ±0.05	15.00 ±0.06	15.02 ±0.11

Table 8: Analogous table to Tab. 7 for the number of inferred experts (hyperplanes).

	LR	SVM	RVM	AMM	CPM	softplus	sum-ς	stack-ς (T=5)	SS-ς (T=5)
ijcnn1	1	2477	296	8.20 ±0.84	58.00 ±13.04	2	37.60 ±1.52	2 (×5)	38.80 (×5) ±0.84 (×5)
a9a	1	11506	109	28.00 ±4.12	7.60 ±2.19	2	37.60 ±0.55	2 (×5)	40.00 (×5) ±0.00 (×5)

Table 9: Performance of stack-softplus regression with the depth set as $T \in \{1, 2, 3, 5, 10\}$, where stack-softplus regression with $T = 1$ reduces to softplus regression.

Dataset	softplus	stack-ς (T=2)	stack-ς (T=3)	stack-ς (T=5)	stack-ς (T=10)
ijcnn1	8.41 ± 0.03	6.73 ± 0.13	6.44 ± 0.21	6.43 ± 0.15	$\mathbf{6.39} \pm 0.08$
a9a	15.02 ± 0.06	14.96 ± 0.04	$\mathbf{14.93} \pm 0.06$	15.00 ± 0.06	14.97 ± 0.08

Table 10: Performance of SS-softplus regression with $K_{\max} = 20$ and the depth set as $T \in \{1, 2, 3, 5, 10\}$, where SS-softplus regression with $T = 1$ reduces to sum-softplus regression.

Dataset	sum-ς	SS-ς (T=2)	SS-ς (T=3)	SS-ς (T=5)	SS-ς (T=10)
ijcnn1	3.39 ± 0.17	2.32 ± 0.18	2.31 ± 0.17	2.24 ± 0.12	$\mathbf{2.19} \pm 0.11$
a9a	$\mathbf{14.88} \pm 0.05$	14.98 ± 0.03	15.07 ± 0.20	15.02 ± 0.11	15.09 ± 0.06