Distributed Generative Modelling with Sub-Linear Communication Overhead

Nico Piatkowski

TU Dortmund, AI Group
nico.piatkowski@tu-dortmund.de

Abstract. Pushing machine learning towards the edge, often implies the restriction to ultra-low-power (ULP) devices with rather limited compute capabilities. Generative models estimate the data generating probability mass $\mathbb{P}^*$ which can in turn be used for various tasks, including simulation, prediction/forecasting, and novelty detection. Whenever the actual learning task is unknown at learning time or the task is allowed to change over time, learning a generative model is the only viable option. However, learning such models on resource constrained systems raises several challenges. Recent advances in exponential family learning allow us to estimate sophisticated models on highly resource-constrained systems. Nevertheless, the setting in which the training data is distributed among several devices in a network with presumably high communication costs has not yet been investigated. We close this gap by deriving and exploiting a new property of integer models. More precisely, we present a model averaging scheme whose communication complexity is sub-linear w.r.t. the parameter dimension d, and provide an upper bound on the global loss. Experimental results on benchmark data show, that the aggregated model is often on par with the non-distributed global model.

Keywords: distributed learning · undirected models · integer models · model averaging

1 Introduction

When data is collected at various physical locations, we are faced with several opportunities regarding the subsequent data processing. Data might be partitioned in several different ways. Two prototypical scenarios are depicted in Fig. 1. In the horizontal scenario, the full data is distributed instance-wise over multiple devices. This happens due to storage or privacy restrictions. For vertically distributed data, different devices measure different features of the same instance. This does not imply that those features are independent. An example are large industrial processes, where measurement hardware itself is distributed. Here, we consider case a), i.e., the same data generating process can be observed at multiple locations. Approaches for case b) can be found in [14].

The most obvious option to address horizontal data distribution is to send the data to a central server. This comes of course with a huge communication

overhead and a loss of privacy. To address these issues, we may aggregate the collected data to reduce the communication. Moreover, data points can be perturbed [4] to increase the privacy of the data source. However, we still have to send a significant amount of data over the network. Instead of sending the raw, aggregated, or perturbed data, an alternative is to learn the model directly at the edge, i.e., where the data is actually generated. In this scenario, models are updated whenever new data arrives at the device. If a convergence criterion is met, e.g., based on distributed convex thresholding [16], the models of all devices are collected and aggregated, to arrive at a global solution that can benefit from the complete data. Such an aggregation can be carried out by Radon machines [8]. However, we will resort to a simple averaging operation that is reminiscent of Bayesian model averaging [5].

A huge set of machine learning techniques can potentially be applied in this setting. Here, we restrict ourselves to generative probabilistic models for discrete data, which can used for various tasks, including simulation, prediction/forecasting, and novelty detection. Moreover, these models are statistically sound, in the sense that they allow for consistent estimation of the data generating probability mass. This is especially interesting when the actual learning task is unknown at learning time or the task is allowed to change over time.

Learning a model close to where the data is actually generated, often implies the restriction to ultra-low-power (ULP) devices with rather limited compute capabilities. Especially in the distributed or federated learning settings, edge devices are subject to strong resource constraints. Communication efficiency [9] and computational burden [13, 2] must be reduced, in order to get along with the available hardware. Here, we will resort to integer undirected models [13, 11] which provide a complete framework for learning and inference under heavy resource constraints. Nevertheless, the setting in which the training data is distributed among several devices in a network with presumably high communication costs has not yet been investigated in the context of integer undirected models. We close this gap by deriving and exploiting a new property of integer models. More precisely, we show that integer models have a high intrinsic sparsity. Based on this observation, we present a model averaging scheme whose communication complexity is sub-linear w.r.t. the parameter dimension d.

2 Notation and Background

Let us summarize the notation and background necessary for the subsequent development. The Kullback-Leibler divergence between two probability mass functions $\mathbb{P}$ and $\mathbb{Q}$ is defined by $\mathrm{KL}[\mathbb{Q}\|\mathbb{P}] = \sum_{\boldsymbol{x}\in\mathcal{X}} \mathbb{Q}(\boldsymbol{x})(\log \mathbb{Q}(\boldsymbol{x}) - \log \mathbb{P}(\boldsymbol{x}))$, which is never negative and zero if and only if $\mathbb{P} = \mathbb{Q}$. The set $\mathbb{N}$ contains all non-negative integers.

2.1 Undirected Models

An undirected graph $G = (V, E)$ consists of $n = |V|$ vertices, connected via edges $(v, w) \in E$. A clique C is a fully-connected subset of vertices, i.e., $\forall v, w \in$

a)

	id	$\boldsymbol{X}_1$	$\boldsymbol{X}_2$	$\boldsymbol{X}_3$	$\boldsymbol{X}_4$
I	1	$\boldsymbol{x}_1^1$	$\boldsymbol{x}_2^1$	$\boldsymbol{x}_3^1$	$\boldsymbol{x}_4^1$
	2	$\boldsymbol{x}_1^2$	$\boldsymbol{x}_2^2$	$\boldsymbol{x}_3^2$	$\boldsymbol{x}_4^2$
II	3	$\boldsymbol{x}_1^3$	$\boldsymbol{x}_2^3$	$\boldsymbol{x}_3^3$	$\boldsymbol{x}_4^3$
	4	$\boldsymbol{x}_1^4$	$\boldsymbol{x}_2^4$	$\boldsymbol{x}_3^4$	$\boldsymbol{x}_4^4$

b)

		III		IV	
id	$\boldsymbol{X}_1$	$\boldsymbol{X}_2$	$\boldsymbol{X}_3$	$\boldsymbol{X}_4$
1	$\boldsymbol{x}_1^1$	$\boldsymbol{x}_2^1$	$\boldsymbol{x}_3^1$	$\boldsymbol{x}_4^1$
2	$\boldsymbol{x}_1^2$	$\boldsymbol{x}_2^2$	$\boldsymbol{x}_3^2$	$\boldsymbol{x}_4^2$
3	$\boldsymbol{x}_1^3$	$\boldsymbol{x}_2^3$	$\boldsymbol{x}_3^3$	$\boldsymbol{x}_4^3$
4	$\boldsymbol{x}_1^4$	$\boldsymbol{x}_2^4$	$\boldsymbol{x}_3^4$	$\boldsymbol{x}_4^4$

Fig. 1. Two prototypical scenarios for data distribution of an exemplary data set with $n = 4$ features (columns) and $N = 4$ data points (rows). Left: Horizontal distribution. The data points are distributed among devices I and II. Right: Vertical distribution. The features are distributed among devices III and IV.

$C : (v, w) \in E$. The set of all maximal cliques of G is denoted by $\mathcal{C}$. Here, any undirected graph represents the conditional independence structure of some n-dimensional random variable $\boldsymbol{X}$ [15]. To this end, we identify each vertex $v \in V$ with a random variable $\boldsymbol{X}_v$ taking values in the state space $\mathcal{X}_v$. The random vector $\boldsymbol{X} = (\boldsymbol{X}_v : v \in V)$, with probability mass function (pmf) $\mathbb{P}$, represents the random joint state of all vertices in some arbitrary but fixed order, taking values $\boldsymbol{x}$ in the Cartesian product space $\mathcal{X} = \bigotimes_{v \in V} \boldsymbol{X}_v$. If not stated otherwise, $\mathcal{X}$ is a discrete set. Moreover, we allow to access these quantities for any proper subset of variables $S \subset V$, i.e., $\boldsymbol{X}_S = (\boldsymbol{X}_v : v \in S)$, $\boldsymbol{x}_S$, and $\mathcal{X}_S$, respectively. According to the Hammersley-Clifford theorem [6], the probability mass of $\mathcal{X}$ factorizes over positive functions $\psi_C : \mathcal{X} \to \mathbb{R}_+$, one for each maximal clique of the underlying graph,

$$\mathbb{P}(\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C) \,, \tag{1}$$

normalized via $Z = \sum_{\boldsymbol{x} \in \mathcal{X}} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C)$. Due to positivity of ψ_C, it can be written as an exponential, i.e., $\psi_C(\boldsymbol{x}_C) = \exp(\langle \boldsymbol{\theta}_C, \phi_C(\boldsymbol{x}_C) \rangle)$ with sufficient statistic $\phi_C : \mathcal{X}_C \to \mathbb{R}^{|\mathcal{X}_C|}$. Here, we assume the use of overcomplete sufficient statistic, i.e., for discrete data, $\phi_C(\boldsymbol{x}_C)$ is a $|\mathcal{X}_C|$-dimensional "one-hot" vector, where the single 1 entry indicates the specific state $\boldsymbol{x}_C$ of the clique C. Thus, $\psi_C(\boldsymbol{x}_C) = \exp(\langle \boldsymbol{\theta}_C, \phi_C(\boldsymbol{x}_C) \rangle) = \exp(\boldsymbol{\theta}_{C=\boldsymbol{x}_C})$. The full joint pmf can then be written in the famous exponential family form $\mathbb{P}(\boldsymbol{X} = \boldsymbol{x}) = \exp(\langle \boldsymbol{\theta}, \phi(\boldsymbol{x}) \rangle - A)$ with $\boldsymbol{\theta} = (\boldsymbol{\theta}_C : C \in \mathcal{C})$, $\phi(\boldsymbol{x}) = (\phi_C(\boldsymbol{x}_C) : C \in \mathcal{C})$, and log-partition function $A = \log Z = \log \sum_{\boldsymbol{x}} \exp(\langle \boldsymbol{\theta}, \phi(\boldsymbol{x}) \rangle)$.

The parameters of exponential family members are estimated by minimizing the negative average log-likelihood $\ell(\boldsymbol{\theta}; \mathcal{D}) = -(1/|\mathcal{D}|) \sum_{\boldsymbol{x} \in \mathcal{D}} \log \mathbb{P}_{\boldsymbol{\theta}}(\boldsymbol{x})$ for some data set $\mathcal{D}$ via first-order numeric optimization methods. $\mathcal{D}$ contains samples from $\boldsymbol{X}$, and it can be shown that the estimated probability mass converges to the data generating distribution $\mathbb{P}^*$ as the size of $\mathcal{D}$ increases.

In the context of horizontally distributed data (Fig. 1a), we assume the existence of k data sets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$, each generated by $\mathbb{P}^*$, and collected by one of k distributed devices.

2.2 Integer Undirected Models

Pushing machine learning towards the edge, i.e., towards the data generating devices, often translates to pushing machine learning to devices with heavily restricted resources. To facilitate the application of undirected models on such devices, we consider an integer version of $\psi_C(\boldsymbol{x}_C)$ [13]:

$$\bar{\psi}_C(\boldsymbol{x}_C) = 2^{\langle \bar{\boldsymbol{\theta}}_C, \phi_C(\boldsymbol{x}_C)\rangle} \tag{2}$$

with $\bar{\boldsymbol{\theta}} \in \mathbb{N}^d$. Let us shortly recap the different layers of approximation that are involved in integer undirected model: 1) The mere base change from exp to 2 is not an approximation at all—the exponential family of densities can be formulated equivalently with any arbitrary base. 2) The restriction to parameter vectors in $\mathbb{N}^d$ is indeed an approximation. However, it can be shown that the error w.r.t. the model likelihood is bounded—the best integer model is not arbitrarily far away from the best real-valued solution. 3) Probabilistic inference is per se possible in the integer domain, e.g., via belief propagation [10] or Gibbs sampling. To circumvent issues with numerical stability, we use an approximate message passing scheme, called bit-length propagation [11]. In general, we assume that the underlying conditional independence structure G is a tree. If not, we employ the junction tree algorithm [15].

Using integer models has several convenient implications for ultra-low-power systems. First of all, it can be shown that approximate maximum likelihood estimation can be carried out without the need for floating point co-processing units [11]. This reduces both, the required chip-size as well as the power consumption of the underlying hardware: Evaluating (2) reduces to a mere bit-shift instead of a rather costly (in terms of clock-cycles) evaluation of the transcendental function exp. Indeed, having $\langle \bar{\boldsymbol{\theta}}_C, \phi_C(\boldsymbol{x}_C)\rangle > \omega$ results in an overflow during bit-shifting on systems with word-size ω. However, it was shown in [11] that overflows can be prevented by using specialized inference algorithms and data structures. The actual parameter learning is carried out by an integer gradient descent technique that is guaranteed to output a (locally) optimal integer solution.

Second, empirical results show [13, 11], that only a few (≈ 3) bits for each model parameter suffice to achieve practical results in terms of prediction accuracy and approximate marginals probabilities. Technically, learning is carried out over $\{0, 1, \ldots, 2^b - 1\}^d$ instead of the full $\mathbb{N}$—a fact that is always true on practical hardware—where b is a hyper-parameter. Hence, storing and communicating the learned model $\bar{\boldsymbol{\theta}}$ requires less than 10% of bits compared to an ordinary model with 64 bit encoding.

Third, it can be shown that the parameter vector of exponential family models with overcomplete sufficient statistic is *never* dense, i.e., at least $|\mathcal{C}|$ model

parameters are guaranteed to be zero. We exploit and improve this fact in the sequel and use it to devise a distributed learning scheme with sub-linear communication overhead.

3 Distributed Integer Undirected Models

We will now go through a series of theoretical insights which will eventually lead to a new distributed learning scheme for integer undirected models. But before we start, let us stress the meaning of "sub-linear" in the context of exponential family models. As said in the introduction, the easiest solution is to send the raw data to a central server and perform learning there. Having observed N_i n-dimensional data points at device i, this amounts to $nN_i\omega$ transmitted bits, assuming word-size ω. In this extreme case, no computational resources are required at the data source but at the cost of maximum communication complexity. One could tend to say that communicating less than $\mathcal{O}(nN_i\omega)$ is "sub-linear". However, in case of exponential family models, neither sending nor storing the full amount of data is required at all—the model parameters can be learned from an aggregated version of the data set. To see this, consider the objective function of the integer model:

$$\begin{aligned}\ell(\boldsymbol{\theta};\mathcal{D}) &= -\frac{1}{|\mathcal{D}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\log_2 \mathbb{P}_{\boldsymbol{\theta}}(\boldsymbol{x}) = -\frac{1}{|\mathcal{D}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\log_2 2^{\langle\boldsymbol{\theta},\phi(\boldsymbol{x})\rangle - A_2(\boldsymbol{\theta})} \\ &= -\frac{1}{|\mathcal{D}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\langle\boldsymbol{\theta},\phi(\boldsymbol{x})\rangle - A_2(\boldsymbol{\theta}) = A_2(\boldsymbol{\theta}) - \langle\boldsymbol{\theta},\frac{1}{|\mathcal{D}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\phi(\boldsymbol{x})\rangle \qquad (3)\end{aligned}$$

Setting $\boldsymbol{\mu} = {}^1\!/_{|\mathcal{D}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\phi(\boldsymbol{x})$, we see that $\ell(\boldsymbol{\theta};\mathcal{D}) = A_2(\boldsymbol{\theta}) - \langle\boldsymbol{\theta},\boldsymbol{\mu}\rangle = \ell(\boldsymbol{\theta};\boldsymbol{\mu})$, i.e., $\boldsymbol{\theta}$ can be learned from the average sufficient statistic $\boldsymbol{\mu}$—access to the raw data set $\mathcal{D}$ is not required. Assuming a process that generates new data points in some fixed time intervals, it is straightforward to update $\boldsymbol{\mu}$ as a running average. Thus, in case of exponential family models, transmitting $\boldsymbol{\mu}$ and $|\mathcal{D}|$ to some central server is equivalent to transmitting the full data set. This implies that sub-linearity requires a communication complexity that is strictly less than d—the dimension of $\phi(\boldsymbol{x})$ and $\boldsymbol{\theta}$. To achieve this, we first exploit a property of the one-hot encoding that underlies $\phi(\boldsymbol{x})$.

Theorem 1 (Overcomplete models are always sparse). *Denote the number of non-zeros by* $\|\boldsymbol{\theta}\|_0 = \sum_{i=1}^{d}|\boldsymbol{\theta}_i|^0$ *with* $0^0 = 0$*, and let* $\boldsymbol{\theta}\in\mathbb{R}^d$ *with* $\|\boldsymbol{\theta}\|_0 = d$ *be the dense parameter vector of some exponential family member with overcomplete sufficient statistic* ϕ*. Then, there is* $\boldsymbol{\theta}'\in\mathbb{R}^d$ *such that* $\mathbb{P}_{\boldsymbol{\theta}} = \mathbb{P}_{\boldsymbol{\theta}'}$ *and*

$$\|\boldsymbol{\theta}'\|_0 \leq \|\boldsymbol{\theta}\|_0 - |\mathcal{C}| < \|\boldsymbol{\theta}\|_0 = d$$

Proof. Exponential family models with overcomplete sufficient statistics are shift-invariant w.r.t. each clique parameter vector [11]. Recall that $\boldsymbol{\theta}$ is defined

as the concatenation of the parameter vectors of all cliques. Consider the d-dimensional vector $S_C(\alpha)$ which is zero everywhere, except for the positions of the parameters that belong to the clique C—at these positions, we put the value α. For the proof, explicit knowledge about these positions it not required. The only important fact is that some arbitrary subset of the d dimensions contains the parameters for clique C. Now, consider the vector $\boldsymbol{\theta}' = \boldsymbol{\theta} + S_C(\alpha)$. We have

$$\mathbb{P}_{\boldsymbol{\theta}'}(\boldsymbol{x}) = \frac{\exp(\langle \boldsymbol{\theta}', \phi(\boldsymbol{x})\rangle)}{\sum_{\boldsymbol{x}'} \exp(\langle \boldsymbol{\theta}', \phi(\boldsymbol{x}')\rangle)} = \frac{\exp(\langle \boldsymbol{\theta}, \phi(\boldsymbol{x})\rangle + \alpha)}{\sum_{\boldsymbol{x}'} \exp(\langle \boldsymbol{\theta}, \phi(\boldsymbol{x}')\rangle + \alpha)} = \mathbb{P}_{\boldsymbol{\theta}}(\boldsymbol{x})$$

The equality in the middle holds because $\langle S_C(\alpha), \phi(\boldsymbol{x})\rangle = \alpha$ for all $\boldsymbol{x} \in \mathcal{X}$. This is, in fact, a direct implication of ϕ's overcompleteness. The above result is called *shift-invariance*.

Let $\boldsymbol{\theta}_{C,1}$ be the first parameter of each clique's parameter vector. We construct the vector $\boldsymbol{\theta}' = \boldsymbol{\theta} + \sum_{C\in\mathcal{C}} S_C(-\boldsymbol{\theta}_{C,1})$. Shift-invariance holds for each C, and thus $\mathbb{P}_{\boldsymbol{\theta}'}(\boldsymbol{x}) = \mathbb{P}_{\boldsymbol{\theta}'}(\boldsymbol{x})$. By assumption, $\boldsymbol{\theta}$ is dense, i.e., $\|\boldsymbol{\theta}\|_0 = d$. By construction, $\boldsymbol{\theta}'$ must have at least $|\mathcal{C}|$ zero-values. $\|\boldsymbol{\theta}'\|_0 \leq \|\boldsymbol{\theta}\|_0 - |\mathcal{C}|$ holds with equality if all d dimensions of $\boldsymbol{\theta}$ have a different value. □

The above theorem guarantees that any exponential family member with overcomplete sufficient statistic has an optimal parameter with at least $|\mathcal{C}|$ zero-entries. This result arises from overcompleteness and has not yet any specific connection to our integer models. Based on this result, we provide the following genuine new insight:

Theorem 2 (Integer models are sparser). *Let $\boldsymbol{\theta} \in \{1, \ldots, 2^b - 1\}^d$ be the dense parameter vector of some integer exponential family member with overcomplete sufficient statistic ϕ. Let further $|\mathcal{X}_{C_{\min}}|$ be the smallest clique state space. Then, if b is chosen such that $2^b - 1 < |\mathcal{X}_{C_{\min}}|$, there exists $\boldsymbol{\theta}'$ such that $\mathbb{P}_{\boldsymbol{\theta}} = \mathbb{P}_{\boldsymbol{\theta}'}$ and*

$$\|\boldsymbol{\theta}'\|_0 \leq \|\boldsymbol{\theta}\|_0 - 2|\mathcal{C}| < \|\boldsymbol{\theta}\|_0 = d$$

Proof. By assumption, there are less parameter values than clique states. Thus, each clique parameter vector $\boldsymbol{\theta}_C$ must contain one parameter value z at least twice. Again, we exploit shift-invariance to subtract z from each parameter in $\boldsymbol{\theta}_C$ which generates at least 2 zero values. This procedure is applied to all cliques $C \in \mathcal{C}$ to end up with a parameter vector $\boldsymbol{\theta}'$ that contains at least $2|\mathcal{C}|$ zeros. □

Thus the number of zero-parameters is increased by at least a factor of 2. The same idea cannot be applied to ordinary (non-integer) exponential families—there, all real-valued parameters will be different with probability 1.

Our new result tells us that integer models have not only computational benefits but also non-trivial implications when the learned integer model has to be transmitted. The final step is the aggregation of independent local models. For simplicity, we restrict ourselves to plain model averaging. Assuming that the underlying network allows for broadcasts, this operation can be carried out locally in each device. If no broadcast is available, we can send each model to a designated central node that carries out the aggregation. Clearly, model averaging involves some error, due to the non-linearity of the exponential family.

However, the following corollary raises some hope that the averaged model is not too bad.

Lemma 1 (Upper bound on the loss). *Let $\boldsymbol{\theta}^i \in \mathbb{R}^d$ for $i = 1, 2, \ldots, k$ be the parameter vector of the model learned on the i-th device. Let further $\hat{\boldsymbol{\theta}} = \sum_{i=1}^k \alpha^i \boldsymbol{\theta}^i$ with $\alpha_i > 0$ and $\sum_{i=1}^k \alpha^i = 1$ be the corresponding model average. Then, for any arbitrary data set $\mathcal{D}$, we have*

$$\ell(\hat{\boldsymbol{\theta}}; \mathcal{D}) \leq \sum_{i=1}^k \alpha_i \ell(\boldsymbol{\theta}^i; \mathcal{D})$$

Proof. The negative log-likelihood is convex. The result is thus a direct corollary of Jensen's inequality. □

At a first glace, this result seems odd, since it suggests that the global model average is *better* than the local models. It is important to understand that the negative log-likelihoods on the right-hand-side are computed w.r.t. the (global) data set $\mathcal{D}$, and not w.r.t. the local sets $\mathcal{D}_i$. The local model's loss can be arbitrarily large on $\mathcal{D}$ which explains why this inequality holds. However, in the joint limit of $|\mathcal{D}_i| \to \infty$ for $i = 1, \ldots, k$, all local data sets are equivalent and the inequality will turn into an equality. In practice, we want to explore the space "in between", where a finite amount of data has been observed at each device, but the individual local models are still similar to each other. The pairwise distances between local average sufficient statistics can be bounded by a function of the available data.

Lemma 2 (Distance between expected statistics). *Let $\boldsymbol{X}$ be a random variable with state space $\mathcal{X}$, $\mathcal{D}_i$ and $\mathcal{D}_j$ two pairwise independent data sets with samples from $\boldsymbol{X}$, and $\phi : \mathcal{X} \to \mathbb{R}^d$ some function. Denote the estimated expectation of $\phi(\boldsymbol{X})$ w.r.t. $\mathcal{D}_i$ by $\boldsymbol{\mu}^i = \frac{1}{|\mathcal{D}_i|} \sum_{\boldsymbol{x} \in \mathcal{D}_i} \phi(\boldsymbol{x})$ and likewise for $\boldsymbol{\mu}^j$. Then,*

$$\|\boldsymbol{\mu}^i - \boldsymbol{\mu}^j\|_\infty \leq 2\sqrt{\frac{(c+1)\log d}{2|\mathcal{D}'|}} = \epsilon$$

with probability of at least $\delta = (1 - 2\exp(-c \log d))^2$ for any $c > 0$. D' is the smaller of the two data sets $\mathcal{D}_i$ and $\mathcal{D}_j$.

Proof. $\boldsymbol{\mu}^i$ is unbiased due to $\mathbb{E}[\boldsymbol{\mu}^i] = \frac{1}{|\mathcal{D}_i|} \sum_{\boldsymbol{x} \in \mathcal{D}_i} \mathbb{E}[\phi(\boldsymbol{X})] = \boldsymbol{\mu}^*$. According to Hoeffding's inequality [7],

$$\mathbb{P}(|{\boldsymbol{\mu}^i}_l - \mathbb{E}[\boldsymbol{\mu}^i]_l| > t) \leq 2\exp(-2|\mathcal{D}|t^2)$$

for all $t > 0$. Since this holds for any dimension l, we can apply the union bound to get

$$\mathbb{P}(\exists l \in [d] : |{\boldsymbol{\mu}^i}_l - \boldsymbol{\mu}^*_l| > t) \leq 2\exp(-2|\mathcal{D}|t^2 + \log d) \; .$$

We set $t = \sqrt{(c+1)\log d/(2|\mathcal{D}|)}$. Thus $\|\boldsymbol{\mu}^i - \boldsymbol{\mu}^*\|_\infty \leq \sqrt{\frac{(c+1)\log d}{2|\mathcal{D}_i|}}$ with probability at least $1 - 2\exp(-c \log d)$. Indeed, the same holds for $\boldsymbol{\mu}^j$. Finally, we

Algorithm 1: Distributed ULP-Learning of Generative Models

input k local data sets, one per device; desired (ϵ, δ)-pair; parameter width b
output Global model $\hat{\boldsymbol{\theta}} = (1/k) \sum_{i=1}^{k} \boldsymbol{\theta}^i$ (Lemma 1)

for all devices $i = 1, 2, \ldots, k$ in parallel **do**
  if New data arrives **then**
    Update($\boldsymbol{\mu}^i$) (Eq. 3)
    $\boldsymbol{\theta}^i \leftarrow \arg\min_{\boldsymbol{\theta} \in \{0,1,\ldots,2^b-1\}^d} \ell(\boldsymbol{\theta}; \boldsymbol{\mu}^i)$
  end if
  if $|\mathcal{D}_i|$ is large enough to satisfy (ϵ, δ) (Lemma 2) **then**
    Sparsify($\boldsymbol{\theta}^i$) (Theorem 2)
    Broadcast($\boldsymbol{\theta}^i$)
    return
  end if
end for

apply the triangle inequality to derive $\|\boldsymbol{\mu}^i - \boldsymbol{\mu}^j\|_\infty \leq \|\boldsymbol{\mu}^i - \boldsymbol{\mu}^*\|_\infty + \|\boldsymbol{\mu}^j - \boldsymbol{\mu}^*\|_\infty$. Since both events are independent, the final inequality has probability of at least $\delta = (1 - 2\exp(-c \log d))^2$. □

Increasing c makes the probability δ larger, at the cost of an increased distance ϵ. The lemma can help us to decide when local models are "good enough": Informally, $\boldsymbol{\theta}^i$ and $\boldsymbol{\theta}^j$ will approach each other when $\boldsymbol{\mu}^i$ and $\boldsymbol{\mu}^j$ are approaching each other. We will make use of this intuition without providing a proof. However, the relation between $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ can be made explicitly by proof techniques provided in [1]. Here, we choose ϵ and δ to determine the number of samples that is required at each device for all local models being similar with high probability.

The final distributed learning procedure is provided in Alg. 1. There, evaluating the stopping criterion requires knowledge about the amount of data that has been collected by each device—this number could be transmitted in a recurring manner. Here, for simplicity, we assume that data arrives synchronously at the devices and that all devices are started at the same point in time. Hence, all models will collect the same number of data points.

Note that the global model $\hat{\boldsymbol{\theta}}$ is likely to be non-integer. The resulting model average can be rounded to recover an integer solution. This, however, involves an additional approximation error [11]. Instead, we scale local models by $\log 2$, which results in a base-change back to exp. The scaled output $(\log 2)\hat{\boldsymbol{\theta}}$ is thus the parameter of an ordinary (non-integer) exponential family member. Alg. 1 can hence be re-interpreted as a method that recovers an ordinary exponential family from a set of integer models.

4 Experimental Demonstration

We perform numerical experiments to assess the proposed method. More precisely, we to answer the following questions:

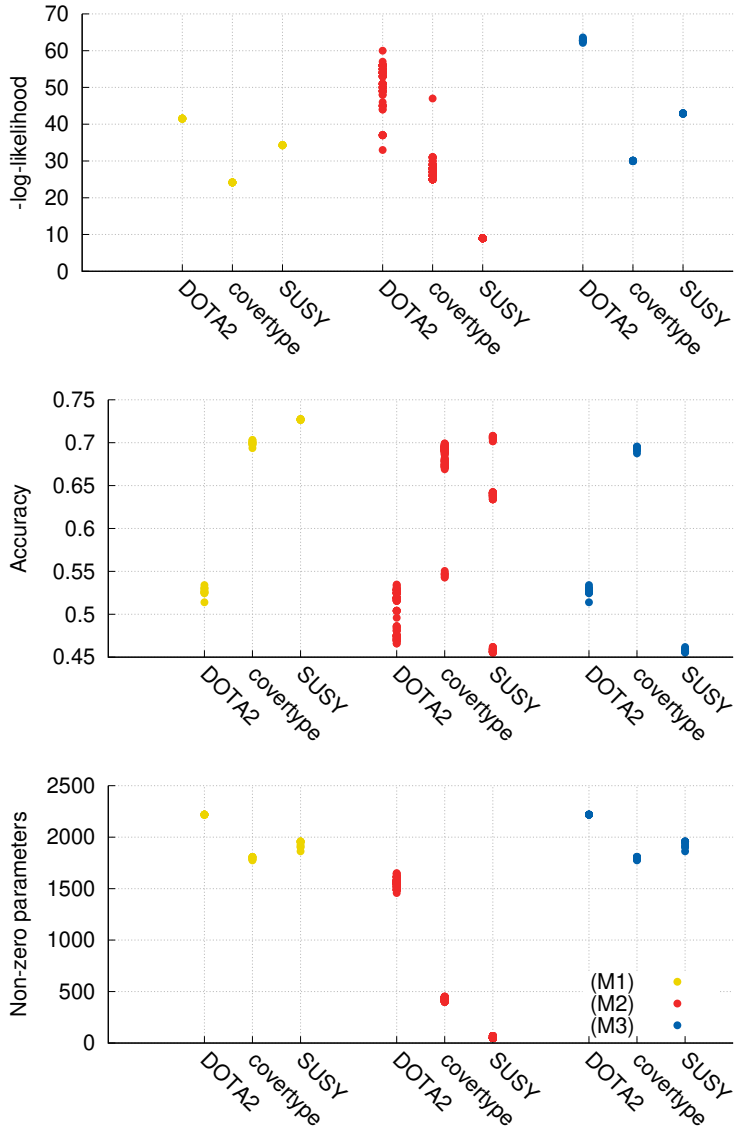

Fig. 2. Experimental results on three benchmark data sets. Each point represents the outcome of a single cross-validation fold. Top: Negative average log-likelihood $\ell(\boldsymbol{\theta}; \mathcal{D})$. Mid: Classification accuracy. Bottom: Number of non-zero values for each learned parameter vector $\boldsymbol{\theta}$, i.e., $\|\boldsymbol{\theta}\|_0$. All three plots share the same key. Best viewed in color.

(Q1) What is the improvement w.r.t. communication complexity on real data sets?

(Q2) How does the averaged model perform, compared to the global model and the individual local models?

All experiments are carried out with the `PX` framework[1]. For (**Q1**), we measure the number of bits that must be transmitted for each model. For (**Q2**), we measure the negative log-likelihood of each model, as well as the classification accuracy. We record these measurements for three different models:

(M1) Ordinary undirected model with access to the full training data.
(M2) $k = 10$ local integer undirected models with $b = 3$.
(M3) Scaled model average $(\log 2)\hat{\boldsymbol{\theta}}$ of all $k = 10$ local (**M2**) models.

All models are trained on three benchmark data sets from the UCI machine learning repository, namely `SUSY` ($n = 19$, $N = 5 \times 10^6$), `covertype` ($n = 55$, $N = 581012$), and `DOTA2` ($n = 117$, $N = 102944$)—representing normal, small, and tiny data sets, respectively. The conditional independence structure G is approximated by the Chow-Liu algorithm [3], computed on a hold-out set of size 10^4. Each numeric variable is discretized into its 10-quantiles. All results are 10-fold cross-validated. For (**M2**), we split the training set of each cross-validation fold further into $k = 10$ separate data sets which are then used as local data sets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$. In total, we have 10 global models, 100 local modes, and 10 averaged models, where each model averaging is performed over 10 local models.

The results are summarized in Fig. 2. Let us first investigate (**Q1**). The third plot in Fig. 2 shows the individual model sizes for each cross-validation run. As asserted by Theorem 2, the local integer models exhibit a superior sparsity while the global model (**M1**) and the averaged model (**M3**) are mostly dense. Moreover, recall that we learned the integer models with $b = 3$, i.e., each model parameter can be encoded with 3 bits. Combining the higher sparsity and the lower representation complexity, we see that the number of bits required to transmit each local model is reduced by a factor of almost 40 on `SUSY` compared to a dense 64 bit floating point model.

Regarding (**Q2**), results for likelihood and accuracy are shown in the first two plots of Fig. 2. Please note that the likelihood-value of the integer models is an approximation, the likelihood-values of the other model types are exact. We observe that the accuracy of all models on `DOTA2` and `covertype` is qualitatively the same, where (**M1**) achieves the best accuracy, followed by (**M2**) and (**M3**). This alone is interesting, since the amount of data available to each local model is $10\times$ lower compared to the global model. On `SUSY`, the accuracy degrades dramatically on the integer models and hence also on the combined model. Moreover, the local models exhibit a much larger variance compared to the other model types. On all data sets, we see that various local models show a much higher classification error than the global model. Indeed, the accuracy of the aggregated model depends strongly on the local model's quality. The classification results for the (**M3**) model on `DOTA2` and `covertype` are almost indistinguishable from the global (**M1**) model, while the accuracy on `SUSY` breaks down. The results for

[1] http://randomfields.org/px

the likelihood show similar effects. However, we see that accuracy and likelihood are not strongly coupled. The likelihood of (**M2**) and (**M3**) is much worse than those of (**M1**) models, the corresponding classification results are yet similar.

5 Conclusion

Based on new theoretical findings about the sparsity of integer undirected models, we proposed a new scheme for the distributed learning of generative exponential family models. Theoretical and experimental results certify that our method has a sub-linear communication complexity—a fraction of bits which are required to transmit the dense models is sufficient to reconstruct a full-fledged exponential family model. In many cases, the reconstructed models exhibit a similar classification performance as non-distributed (global) models. Our scheme can thus serve as the basis for many practical distributed solutions.

Moreover, our results provide several new research opportunities: First, our scheme can be easily combined with recent latent variable models [12] and hence, opens the path for distributed probabilistic deep learning. Second, the stopping criterion and the averaging scheme suggest some room for improvement. Our Hoeffding-bound-based stopping criterion is very pessimistic and requires a very large number of samples to guarantee that all local models are similar with high probability. It shall be investigated if convex thresholding [16] delivers any benefit over the stopping criterion that was derived from Lem. 2. Finally, the results presented in [8] suggest, that the model aggregation based on Radon points delivers a higher quality compared to plain model averaging. We should hence employ radon machines instead of plain model averaging to aggregate the local models.

Acknowledgments This research has been funded by the Federal Ministry of Education and Research of Germany as part of the competence center for machine learning ML2R (01S18038A).

References

1. Bradley, J.K., Guestrin, C.: Sample complexity of composite likelihood. In: International Conference on Artificial Intelligence and Statistics (AISTATS). pp. 136–160 (2012), http://proceedings.mlr.press/v22/bradley12.html
2. Caldas, S., Konecný, J., McMahan, H.B., Talwalkar, A.: Expanding the reach of federated learning by reducing client resource requirements. CoRR **abs/1812.07210** (2018), http://arxiv.org/abs/1812.07210
3. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory **14**(3), 462–467 (1968). https://doi.org/10.1109/TIT.1968.1054142
4. Duchi, J.C., Jordan, M.I., Wainwright, M.J.: Privacy aware learning. J. ACM **61**(6), 38:1–38:57 (2014). https://doi.org/10.1145/2666468

5. Fragoso, T.M., Bertoli, W., Louzada, F.: Bayesian model averaging: A systematic review and conceptual classification. International Statistical Review **86**(1), 1–28 (2018). https://doi.org/10.1111/insr.12243
6. Hammersley, J.M., Clifford, P.: Markov fields on finite graphs and lattices. Unpublished manuscript (1971)
7. Hoeffding, W.: Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association **58**(301), 13–30 (1963)
8. Kamp, M., Boley, M., Missura, O., Gärtner, T.: Effective parallelisation for machine learning. In: Advances in Neural Information Processing Systems (NIPS). pp. 6480–6491 (2017), http://papers.nips.cc/paper/7226-effective-parallelisation-for-machine-learning
9. Konecný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. CoRR **abs/1610.05492** (2016), http://arxiv.org/abs/1610.05492
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Burlington, MA, USA (1988)
11. Piatkowski, N.: Exponential families on resource-constrained systems. Ph.D. thesis, TU Dortmund, Germany (2018), http://hdl.handle.net/2003/36877
12. Piatkowski, N.: Hyper-parameter-free generative modelling with deep boltzmann trees. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD) (2019)
13. Piatkowski, N., Lee, S., Morik, K.: Integer undirected graphical models for resource-constrained systems. Neurocomputing **173**, 9–23 (2016). https://doi.org/10.1016/j.neucom.2015.01.091
14. Stolpe, M.: Distributed analysis of vertically partitioned sensor measurements under communication constraints. Ph.D. thesis, TU Dortmund, Germany (2017), http://hdl.handle.net/2003/35815
15. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning **1**(1–2), 1–305 (2008). https://doi.org/10.1561/2200000001
16. Wolff, R.: Distributed convex thresholding. In: Symposium on Principles of Distributed Computing (PODC). pp. 325–334 (2015). https://doi.org/10.1145/2767386.2767387