Model Fusion with Kullback–Leibler Divergence

Sebastian Claici [* 1 2]   Mikhail Yurochkin [* 2 3]   Soumya Ghosh [2 3]   Justin Solomon [1 2]

Abstract

We propose a method to fuse posterior distributions learned from heterogeneous datasets. Our algorithm relies on a mean field assumption for both the fused model and the individual dataset posteriors and proceeds using a simple assign-and-average approach. The components of the dataset posteriors are assigned to the proposed global model components by solving a regularized variant of the assignment problem. The global components are then updated based on these assignments by their mean under a KL divergence. For exponential family variational distributions, our formulation leads to an efficient non-parametric algorithm for computing the fused model. Our algorithm is easy to describe and implement, efficient, and competitive with state-of-the-art on motion capture analysis, topic modeling, and federated learning of Bayesian neural networks.[1]

1. Introduction

In this paper, we study *model fusion*, the problem of learning a unified global model from a collection of pre-trained local models. Model fusion provides a straightforward and efficient approach to *federated learning* (FL), in which a model is learned from siloed data without direct access.

As a motivating example, any one hospital may be able to use its patient data to train a model aiding diagnosis or treatment, but due to limited data and skew the resulting model may not be effective. To overcome this issue, a group of hospitals could in principle collaborate to produce a stronger model by pooling their data, but it is typically not permissible to share individual patient information between

[1]Code link: https://github.com/IBM/KL-fusion

(a) Bayesian neural network trained on biased data

(b) Fused Bayesian neural network

Figure 1. Bayesian neural networks can estimate how certain they are of each prediction they make. To illustrate this, we train a network on a subset of the MNIST dataset that mostly contains the digit 2 and ask the network to predict on all digits; confidence is predictably low on digits other than 2. However, if we are given several pre-trained networks that each exhibit certain biases, but which in aggregate have seen examples of all 10 digits, can we merge them together into a network that has high accuracy and is confident on all of MNIST? We propose a *model fusion* approach to solve this and other related problems. To illustrate our approach, we train Bayesian neural networks on subsets of MNIST that are skewed toward a few digits and fuse the trained networks. To show that the fused network is confident on all 10 digits, we order samples by the entropy of the network's prediction for the fused model (b), and compare to a local model trained on mostly 2s (a). These images sort examples about which the networks are most confident (left) to least confident (right). The fused network is still unconfident on out-of-sample data not used to train any of the fused networks—in this case, digits from fonts rather than handwriting—while the biased Bayesian network is unconfident on digits different than those seen in its training set.

institutions. Federated learning—including model fusion—provides a means of generating a stronger or more widely-

applicable model than what can be obtained from any one hospital's data, while only sharing aggregate information in the form of model parameters. As a second example, when learning from edge devices (e.g., next word prediction on smart phones), users often do not want their personal data to leave the device. Federated learning algorithms only require parties (i.e., data owners) to share only local model parameters, rather than providing direct access to user-specific local data. Some key aspects that distinguish federated learning from classical distributed learning are (1) constraints on the frequency of communication and (2) heterogeneity of the local datasets.

To overcome these challenges, we fuse heterogeneous models in a *single* communication round that aggregates locally-trained models into a global model. This "one-shot" approach, which distinguishes model fusion from other FL methods, is crucial for certain applications yet is largely overlooked with the exception of Yurochkin et al. (2019a;b). In particular, the one shot approach allows parties to erase data in favor of storing only their local models.

Many common examples motivating FL require well-calibrated uncertainty measurements for the fused model; our examples of medical decision-making and word prediction on smart phones provide applications in which uncertainty quantification can be used to avoid making an inaccurate and potentially unsafe intervention. This context motivates a *Bayesian* approach to model fusion and FL more broadly. The Bayesian nature of our fusion algorithm bolsters trust in systems that use this machinery in applications that encounter out-of-distribution samples when deployed in practice. We demonstrate the value of a Bayesian approach to model fusion through applications from topic models to neural networks.

Contributions. We present a Bayesian approach to model fusion, in which *local* models trained on individual datasets are combined to learn a single *global* model. Our approach fuses models represented using the mean field approximation, common for lightweight and robust local estimation. It is nonparametric in the sense that it does not require the dimensionality of the fused mean field model to be fixed *a priori*. Specific contributions include:

- a non-parametric method to determine the posterior distribution of the fused global model;
- an easily-implemented assign-then-average fusion procedure that scales to large numbers of local posterior distributions and fused model components;
- a model for fusion that is flexible enough to handle posterior distributions in any exponential family; and
- comprehensive validation demonstrating effectiveness for applications including motion capture analysis, topic modeling, and Bayesian neural networks.

2. Related Work

This paper develops model fusion techniques for approximate Bayesian inference, combining parametric approximations to an intractable posterior distribution. While we primarily focus on mean-field variational inference (VI), owing to its popularity, our methods are equally applicable to Laplace approximations (Bishop, 2006), assumed density filtering (Opper, 1998), and expectation propagation (Minka, 2001) methods that learn a parametric approximation to the posterior. Variational inference seeks to approximate the true posterior distribution by a tractable approximate distribution by minimizing the KL divergence between the variational approximation and the true posterior. In contrast with Markov chain Monte Carlo methods, VI relies on optimization and is thus able to exploit advances in stochastic gradient methods allowing for VI based inference algorithms to scale to large data and models with a large number of parameters, such as Bayesian Neural Networks (BNNs) (Neal, 1995) considered in this work.

Distributed posterior inference has been actively studied in the literature (Hasenclever et al., 2017; Broderick et al., 2013; Bui et al., 2018; Srivastava et al., 2015; Bardenet et al., 2017). As with distributed optimization, however, the goal is typically to achieve computational speedups, leading to approaches ill suited for model fusion due to high number of communication rounds required for convergence and assumption on the homogeneity of the datasets. Moreover, the inherent permutation invariance structure of many high-utility models (e.g., topic models, mixture models, HMMs, and BNNs) is ignored by prior distributed Bayesian learning methods as it is of minor importance when many communication rounds are permissible. On the contrary, our model fusion formulation requires careful consideration of the permutation structure as we show in the subsequent section. Aggregation of Bayesian posteriors respecting permutation structure was considered in Campbell & How (2014), but their method is limited to homogeneous data and requires combinatorial optimization except few special cases. Subsequent work relaxes the homogeneity constraint and propose a greedy streaming approach for Dirichlet process mixture models (Campbell et al., 2015).

Yurochkin et al. (2019a;b) studied fusion of parameters of permutation invariant models learned from heterogeneous data, however their approach is not suitable when a global posterior, rather than a point estimate, is desired. This limitation makes their methods ill-suited for fusion of Bayesian neural networks where posterior is required to assess prediction uncertainty—the key utility of BNNs. Their method also precludes using full information provided by the local posteriors, e.g. covariances, that may be necessary to efficiently identify global model with fusion as we demonstrate in the experimental studies.

3. Homogeneous Fusion

Before introducing our non-parametric model for heterogeneous fusion, we consider the simpler *homogeneous* fusion problem. The purpose of this section is to define notation and to introduce the building blocks for the algorithm of §4.

Assume we have D datasets on which we run some inference procedure to recover a mean-field approximation to the posterior distribution. For dataset j, we are thus given

$$p_j(z_1, \ldots, z_L) = \prod_{l=1}^{L} q(z_l | \theta_l^j)$$

where $q(z_l | \theta_l^j)$ is the approximate posterior of component z_l parameterized by θ_l^j. An example to keep in mind is topic modeling, where the z_l's represent topics and the θ_l's are the posterior variational Dirichlet parameters for topic l. Our goal is to recover a single global distribution over the z_l's without returning to the data to learn this global distribution. Hence, all we can use for inference are the parameters θ_l^j extracted from each dataset j.

We can pose the problem of recovering a global posterior as that of minimizing a divergence $\mathcal{D}(\cdot \parallel \cdot)$ to the local posteriors, but the ordering of posterior parameters is different from dataset to dataset. This phenomenon is called *label switching*, and is caused by the permutation invariance of the posterior (Monteiller et al., 2019).

Because we are in the *homogeneous* case, we can assume that the components of each local posterior can be put into correspondence across datasets. This allows us to assume that the global model admits the same product factorization:

$$\bar{p}(z_1, \ldots, z_L) = \prod_{g=1}^{L} q(z_g | \bar{\theta}_g).$$

Our goal is to find an effective choice of the $\bar{\theta}_g$'s, but we must be careful in how we define the objective. In particular, the ordering of components for each dataset can be arbitrary as the posteriors are invariant to permutations in the parameters, and our objective function must account for this permutation invariance.

To this end, we introduce auxiliary optimization variables P^j for each dataset posterior. The P^j's are permutation matrices that allow for parameter reorderings across local models. The problem we wish to solve is then:

$$\min_{\{\bar{\theta}_g\}, \{P^j\}} \quad \sum_{j=1}^{D} \mathcal{D}\left(\prod_{l=1}^{L} q\left(z_g \middle| \sum_{g=1}^{L} P_{lg}^j \bar{\theta}_g \right) \middle\| \prod_{l=1}^{L} q(z_l | \theta_l^j) \right)$$

$$\text{subject to} \quad \sum_{l=1}^{L} P_{lg}^j = 1, \sum_{g=1}^{L} P_{lg}^j = 1, P_{lg}^j \in \{0, 1\}.$$

$$(1)$$

The notation for indices here and later follows the convention that g indexes global parameters, l indexes local parameters, and j indexes datasets; $\mathcal{D}$ is a divergence. To read this equation, notice that the inner sum $\sum_{g=1}^{L} P_{lg}^j \bar{\theta}_g$ selects the global parameter that best explains the z_l. Thus, we can think of (1) as asking for the choice of global parameters $\bar{\theta}_g$ as well as a permutation for each dataset telling how to put the $\bar{\theta}_g$ and θ_l^j in correspondence such that the total divergence over all datasets is minimized.

The tractability of this problem depends on the divergence $\mathcal{D}(\cdot \parallel \cdot)$. One choice that greatly simplifies the problem is the Kullback–Leibler (KL) divergence, which decomposes over product distributions and allows us to write (1) as

$$\min_{\{\bar{\theta}_g\}, \{P^j\}} \quad \sum_{j=1}^{D} \sum_{l=1}^{L} \text{KL}\left(q\left(z_g \middle| \sum_{g=1}^{L} P_{lg}^j \bar{\theta}_g \right) \middle\| q(z_l | \theta_l^j) \right)$$

$$\text{subject to} \quad \sum_{l=1}^{L} P_{lg}^j = 1, \sum_{g=1}^{L} P_{lg}^j = 1, P_{lg}^j \in \{0, 1\}.$$

$$(2)$$

To further simplify, we can exploit the binary constraints in our problems. The P^j's are binary matrices with a single 1 in each row and column. Because all other entries of $P_{\cdot g}^j$ are 0, we can move the sum outside the KL term, as $P_{lg}^j \cdot \text{KL}(\cdot \parallel \cdot)$ will not contribute to the objective function if $P_{lg}^j = 0$. The final form of our objective becomes

$$\min_{\{\bar{\theta}_g\}, \{P^j\}} \quad \sum_{j=1}^{D} \sum_{l,g=1}^{L} P_{lg}^j \text{KL}\left(q(z_g | \bar{\theta}_g) \middle\| q(z_l | \theta_l^j) \right)$$

$$(3)$$

$$\text{subject to} \quad \sum_{l=1}^{L} P_{lg}^j = 1, \sum_{g=1}^{L} P_{lg}^j = 1, P_{lg}^j \in \{0, 1\}.$$

Problem (3) is easier to solve than what we started with in (1). With fixed $\{P^j\}$, the problem is a barycenter or clustering problem under the KL divergence, which is known in closed form for exponential family distributions (§3.1), while with fixed $\{\bar{\theta}_g\}$ it reduces to a stable marriage assignment that can be solved efficiently using the Hungarian algorithm (Kuhn, 1955); this step has worst-case $\tilde{O}(L^3)$ complexity. Since the local parameters θ_l^j are independent across datasets, the P^j's can be computed independently.

3.1. Averaging parametric distributions

For posterior distributions that are in the same exponential family, computing their barycenter under the KL divergence amounts to averaging their natural parameters (Banerjee et al., 2005). In particular, given distributions q_i in the same exponential family Q with natural parameters η_i as well as

a set of weights $\lambda_i \geq 0$ with $\sum_i \lambda_i = 1$, the solution to

$$\min_{q \in Q} \sum_{i=1}^{n} \lambda_i \mathrm{KL}(q \parallel q_i)$$

is a distribution $q^* \in Q$ with natural parameter $\eta^* = \sum_{i=1}^{n} \lambda_i \eta_i$. In our case, given a assignments $\{P^j\}$, we can solve for $\bar{\theta}_g$ by minimizing

$$\min_{q_g \in Q} \sum_{j=1}^{D} \sum_{l=1}^{L} P_{lg}^j \mathrm{KL}\Big(q_g \parallel q(z_l|\theta_l^j)\Big) \tag{4}$$

and converting from natural parameters to $\bar{\theta}_g$.

4. Heterogeneous Fusion

The homogeneous approach requires two assumptions that often do not hold: (1) that the local posterior distributions contain the same number of components and (2) that these components can be matched to one another.

As an extreme example, if we run an inference procedure on data gathered from multiple hospitals, each specializing in a particular set of diseases, we ideally expect the combined model to incorporate information about *all* diseases treated across *all* hospitals. If we run the procedure on the data from each hospital individually, however, then each hospital's model only contains information about the diseases it treats; the mismatch between maladies at different hospitals prevents us from matching their parameters bijectively.

Motivated by this example, we observe that in practical settings the global fused model likely needs *more* components than the local models to capture the fact that local data can be skewed or missing. We call this setting the *heterogeneous* case. In the heterogeneous model, the inference procedure on each dataset may find a different number of components. Some components present in one dataset may not be present in another, and the total number of components is unknown—demanding a nonparametric solution.

4.1. Heterogeneous Model

To describe our model for heterogeneous fusion, we make a few notational changes. Let G be an estimate of the number of global components (we will see how G can be inferred in §4.2), and L_j be the number of components in dataset j. Instead of permutation matrices, the P^j are singly-stochastic, since there may be unmatched global components. We can

modify (1) to take these changes into account:

$$\min_{\{\bar{\theta}_g\},\{P^j\}} \quad \sum_{j=1}^{D} \mathcal{D}\left(\prod_{l=1}^{L_j} q\left(z_g \Bigg| \sum_{g=1}^{G} P_{lg}^j \bar{\theta}_g\right) \Bigg\| \prod_{l=1}^{L_j} q(z_l|\theta_l^j)\right)$$

$$\text{subject to} \quad \sum_{l=1}^{L_j} P_{lg}^j \leq 1, \sum_{g=1}^{G} P_{lg}^j = 1, P_{lg}^j \in \{0,1\}. \tag{5}$$

If we think of P^j as an $L_j \times G$ matrix, the inequality constraint in (5) forces columns of P^j to zero if global component g is not used. The same simplifications we used to derive (3) in §3 apply here, leading to the following formulation of the heterogeneous problem when G is known:

$$\min_{\{\bar{\theta}_g\},\{P^j\}} \quad \sum_{j=1}^{D} \sum_{l=1}^{L_j} \sum_{g=1}^{G} P_{lg}^j \mathrm{KL}\Big(q(z_g|\bar{\theta}_g) \parallel q(z_l|\theta_l^j)\Big)$$

$$\text{subject to} \quad \sum_{l=1}^{L_j} P_{lg}^j \leq 1, \sum_{g=1}^{G} P_{lg}^j = 1, P_{lg}^j \in \{0,1\}. \tag{6}$$

This model can cope with mismatched components among the local models (e.g., different diseases appearing at different hospitals), but it does not tell how to choose the number of global components G—a challenge we address next.

4.2. Estimating the number of global components

A key issue with (6) is that we do not know the true number of global components G. If we naïvely overestimate this parameter, there is neither a term in the objective nor a constraint in (6) that would reduce the number of components that are used; in the extreme case, the fused model would simply concatenate all the components of the local models without clustering any of them together.

To motivate our approach to choosing G, we first consider the simpler problem of matching a *single* local model to the global model. The optimization variables in this case are the $\bar{\theta}_g$'s as before, as well as a single $L \times G$ matching matrix P. We believe that the local model can be approximated best by a small number of components, and we wish to encode this explicitly in the objective.

Recall that a component of the global model that goes unused corresponds to a 0 column of P. For a binary matrix P with $G > L$, 0 columns occur for exactly $G - L$ of the columns when optimizing (6) with $L = 1$. But, if we relax the binary constraint, this may no longer be the case. Inspired by the approach of Carli et al. (2013) to clustering using optimal transport, to promote 0 columns in P for the relaxed problem where $P_{lg} \in [0,1]$, we regularize our problem using the $L_{2,1}$ matrix norm. This approach

can be understood as optimizing the L_1 norm of the vector of L_2 column norms of P, promoting sparsity in the vector of norms and hence existence of 0 columns in P. In mathematical notation, our relaxed objective with the sparsity-promoting regularizer is

$$\sum_{l,g} P_{lg} \mathrm{KL}\big(q(z_g|\bar{\theta}_g) \,\|\, q(z_l|\theta_l)\big) + \lambda \sum_{g=1}^{G} \left(\sum_{l=1}^{L} P_{lg}^2\right)^{1/2}. \tag{7}$$

The mixed-norm regularizer is needed instead of a simpler L_1 regularizer, since the constraints of (6) effectively prescribe the L_1 norm of P to a fixed constant.

A naïve extension to multiple local models might sum (7) over all datasets, but this approach only promotes sparsity within the individual global-to-local assignments. This can lead to a scenario wherein every global component is assigned to some local component at least once, again saturating the total number G of available global components. Instead, following the intuition above, we can view the set $\{P^j\}_{j=1}^{D}$ as a tensor and minimize an $L_{2,2,1}$ tensor norm:

$$\sum_{g=1}^{G} \left(\sum_{j=1}^{D} \left(\sum_{l=1}^{L_j} (P_{lg}^j)^2\right)\right)^{1/2}. \tag{8}$$

This quantity is the $L_{2,1}$ norm of the $G \times D$ matrix whose element at position (g,j) is the norm of column g in P^j.

4.3. The heterogeneous matching problem

Combining (5)–(8) yields the following problem:

$$\min_{\{\bar{\theta}_g\},\{P^j\}} \sum_{j=1}^{D}\sum_{l=1}^{L_j}\sum_{g=1}^{G} P_{lg}^j \mathrm{KL}\Big(q(z_g|\bar{\theta}_g) \,\Big\|\, q(z_l|\theta_l^j)\Big) +$$

$$\lambda \sum_{g=1}^{G} \left(\sum_{j=1}^{D} \left(\sum_{l=1}^{L_j} (P_{lg}^j)^2\right)\right)^{1/2}$$

$$\text{subject to } \sum_{l=1}^{L_j} P_{lg}^j \le 1, \ \sum_{g=1}^{G} P_{lg}^j = 1, \ P_{lg}^j \in \{0,1\}. \tag{9}$$

Choosing λ. To choose a regularization parameter λ, we ensure that the two terms in the objective have the same scale. Since the P^j's are positive matrices with entries less than 1, the scale of the problem is given by the $\mathrm{KL}(\cdot \,\|\, \cdot)$ terms. An empirically effective choice is to scale the divergences by their standard deviation and set $\lambda = 0.1$.

Alternation algorithm. Optimizing (9) is not straightforward. In the homogeneous case, we exploited the fact that

P^j was a permutation to relax the binary constraints and recover a binary matrix; thanks to the inequality in (6), however, we are no longer guaranteed to find a binary P^j if we relax the $P_{lg}^j \in \{0,1\}$ constraint. Relaxing the binary constraints, however, turns (9) into a convex problem in the $\{P^j\}$ and $\{\bar{\theta}_g\}$ individually. Our alternation approach from §3 can be modified to suit our new problem. Because we are no longer guaranteed that the P^j's are permutations, we take a weighted average when updating the $\bar{\theta}_g$'s using (4).

5. Experimental Results

5.1. Simulated experiments

We begin by verifying KL-fusion in a synthetic setting. We consider the problem of fusing Gaussian mixture models with arbitrary means and covariances. Our goal is to estimate true data-generating mixture components by fusing local posterior approximations.

To quantify estimation quality we compute Hausdorff distances between the polytopes spanned by true and estimated mixture component means, as suggested by Nguyen (2013; 2015). We also evaluate error in the fused model size G relative to the true value. Typical Gaussian mixtures assume a Gaussian–Wishart prior for means and covariances of the components (Bishop, 2006); in this case, the posterior can be estimated efficiently using mean-field variational inference with Gaussian–Wishart variational distributions (Attias, 2000). To simulate instances of the heterogeneous fusion problem, when generating local dataset, we sample a random subset of the global mixture components and add Gaussian noise to add heterogeneity in model size and parameters. We describe the generating process precisely in the supplemental document.

We consider three baselines:

- "Oracle" variational inference (VI) trained on a pooled dataset given the true number of global components (this reference baseline that is infeasible in model fusion);
- Dirichlet process (DP) based clustering of the mean components of the local posteriors (Ferguson, 1973; Blei & Jordan, 2006); and
- the SPAHM fusion method (Yurochkin et al., 2019a).

The fusion method of Campbell & How (2014) is too inefficient for this problem, since it requires combinatorial search; see §5.3 for a comparison to their method.

Our first experiment demonstrates failure of prior methods when the means of the data-generating mixture components are poorly-separated; in this case, we need covariances to disambiguate the components. KL-fusion utilizes full posterior, while SPAHM and DP are limited to only using point estimates of the local posterior means. In Figure 2, we
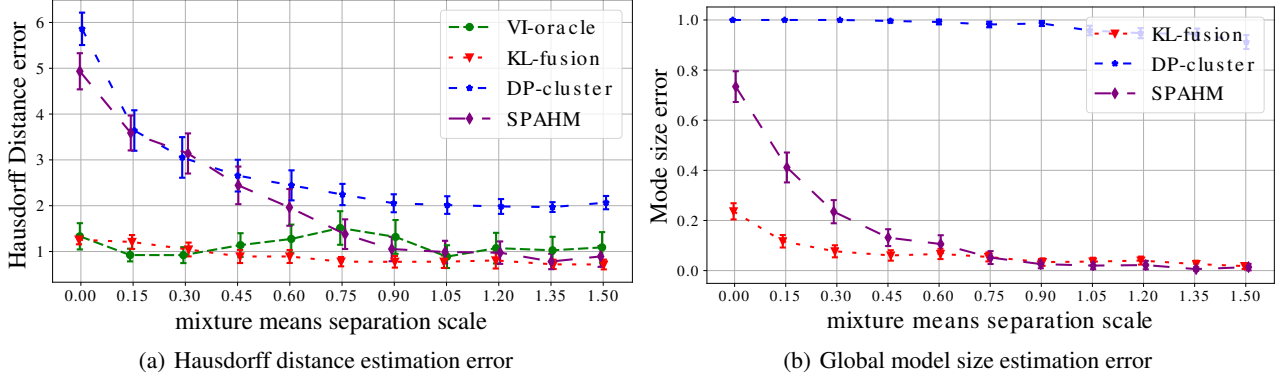
(a) Hausdorff distance estimation error

(b) Global model size estimation error

Figure 2. Fusion of Gaussian mixture posteriors under varying degree of separation between data generation mixture components. KL-fusion can identify true mean parameters under the low separation regime utilizing the covariance information.
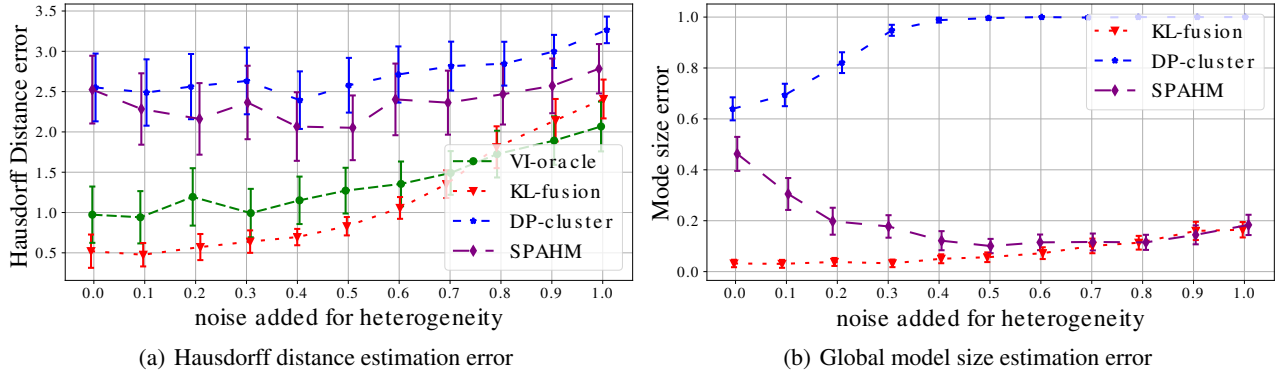

(a) Hausdorff distance estimation error

(b) Global model size estimation error

Figure 3. Fusion of Gaussian mixture posteriors under varying heterogeneity in local datasets enforced via noise standard deviation added to global means when simulating local means. KL-fusion outperforms baselines and degrades gracefully.

vary scale of the variance used to generate means of the true mixture components (higher x-axis value implies better separation): KL-fusion is the only fusion method capable of accurate estimation on par with the VI oracle in the low-separation regime. SPAHM performs well only when mixture means are well-separated.

In our second experiment, we fix the separation scale (at 0.5) and study the effect of noise on the generating process for local components. As before, to generate local components we sample from a random subset of the global mixture components and add Gaussian noise to introduce local heterogeneity. We vary this noise in Figure 3 and report results on estimating the true number of components as well as the Hausdorff distance between estimated and true global models (in the previous experiment we fixed the noise at 0.5). Intuitively, the problem becomes harder as this noise increases, eventually producing datasets that do not relate to the global structure in a meaningful way. In general, our results with KL-fusion and oracle VI support the intuition that performance degrades gracefully as local models vary more from the underlying global model. Fusion-based inference can even outperform pooled inference: local datasets have less structure and are potentially more amenable to mean-

field inference, yielding high-quality approximate posteriors for the subsets of the global model that are easily aggregated using KL-fusion. SPAHM model size estimation error decreases as a function of the noise variance that we add to simulate heterogeneity; this might be caused by the additional separation introduced with this noise.

5.2. Analyzing motion sequences through fused hidden Markov models

We consider the problem of discovering common structures among related time series. As a motivating application, we study data from motion capture sensors on joints of people performing exercise routines, collected from the CMU MoCap database.[2] Each sequence in this database consists of 64 measurements of human subjects performing various exercises over time. Following (Fox et al., 2014), we select 12 informative measurements for capturing gross motor behavior: body torso position, neck angle, two waist angles, and a symmetric pair of right and left angles at each subjects shoulders, wrists, knees, and feet. Each sequence thus is a 12-dimensional time series. We use a curated subset

---

[2] http://mocap.cs.cmu.edu

Table 1. MoCap labeling quality comparison

	Rand index	AMI
KL-fusion	**0.286**	**0.458**
SPAHM	0.254	0.445

collected by Fox et al. (2014) of two subjects each providing three sequences. In addition to having several exercises in common, this subset comes with human-annotated labels, facilitating quantitative comparisons between models.

We use mean-field inference with Gaussian–Wishart variational distributions to obtain the approximate posterior of a "sticky" HDP-HMM (Fox et al., 2008), similar to the analogous experiment by Yurochkin et al. (2019a). KL-fusion matches activities inferred from each subject. We use the Rand index (Rand, 1971) and Adjusted Mutual Information (AMI) (Vinh et al., 2010) to quantify quality of the fused model according the human-annotated labels. Table 5.2 compares performance of KL-fusion to SPAHM, demonstrating improved quality of the activity labelings corresponding to the fused models. This experiment gives a practical example where covariances in a Gaussian–Wishart mean-field approximation improve fusion, enabled by KL-fusion's ability to process exponential family distributions.

5.3. Fusion of topic models

Following Campbell & How (2014), we run decentralized variational inference on the latent Dirichlet allocation topic model (Blei et al., 2003). We verify our method against Approximate Merging of Posteriors with Symmetry (AMPS) algorithm from Campbell & How (2014) on the 20 newsgroups dataset, consisting of 18,689 documents with 1,000 held out for testing and a vocabulary of 12,497 words after stemming and stop word removal.

The full description of the model setup is given in the supplemental document. Briefly, the posterior Dirichlet variational parameters learned on the local datasets are fused using KL-fusion and AMPS, and the resulting models are evaluated by computing the predictive $\log$ likelihood of 10% of the words in each test document given the remaining 90%. Results are given in Figure 4 for 10 trials; we measure the test $\log$ likelihood and amount of time required to compute the fused model. The parametric model in (3) and the objective for AMPS are similar, and thus we expect similar performance from the two methods; our non-parametric version can infer richer models and performs better, but comes at an increased computational cost.

5.4. Fusion of Bayesian neural networks

In this experiment we demonstrate utility of KL-fusion applied to one-shot federated learning of Bayesian neural net-
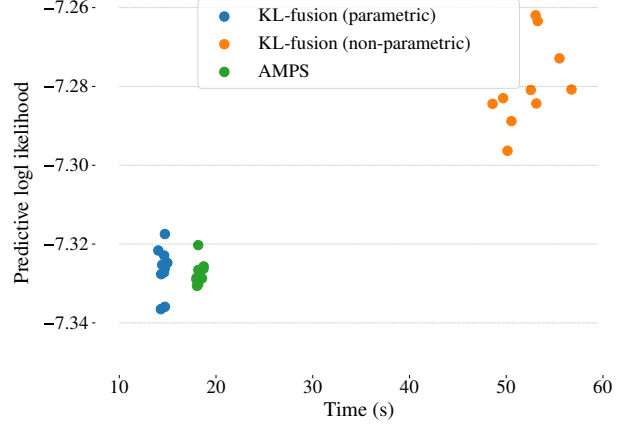

Figure 4. Scatter plot of test $\log$ likelihood and time for the experiment described in §5.3. Higher $\log$ likelihood is better. The parametric model minimizes a very similar objective to AMPS and we observe similar performance between the two. The richer expressiveness of the non-parametric model allows it to perform better, but comes with larger computational requirements.

works. Federated learning systems deployed in practice will inevitably face examples outside of the train data distribution leading to mistakes that might be costly for a business or diminish satisfaction of a user with an edge device. Bayesian neural networks are valued for their ability to quantify prediction uncertainty and raise an alarm when facing an out of distribution (OOD) example, but in a federated learning setting data of any individual client might be insufficient to obtain a good quality BNN.

We measure the effectiveness of our procedure for fusing BNNs locally trained on MNIST digits. For this experiment, we split the MNIST training data into five partitions at random. We simulate a heterogeneous partitioning of the data by sampling the proportion p_k of each class k from a five-dimensional symmetric Dirichlet distribution with a concentration parameter of 0.8, and allocating a $p_{k,j}$ proportion of the instances of class k to partition j. This process results in a non-uniform distribution of classes in each partition. For each dataset we train a single 150-node hidden layer BNN with a horseshoe prior (Ghosh et al., 2019; 2018). Horseshoe is a shrinkage prior allowing BNNs to automatically select the appropriate number of hidden units. We use Gaussian variational distribution with diagonal covariance for the weights of the neurons. Details are presented in the supplement.

We use KL-fusion to obtain a global BNN with 281 units, far smaller than a concatenation of all the local BNNs. Table 2 illustrates that the fused model significantly improves upon the predictive performance of the local models both in terms of accuracy and held-out test $\log$ likelihoods. We also examine the predictive uncertainties produced by the fused BNN. Figure 1 qualitatively compares the predictive entropy produced by the fused BNN against one of the local
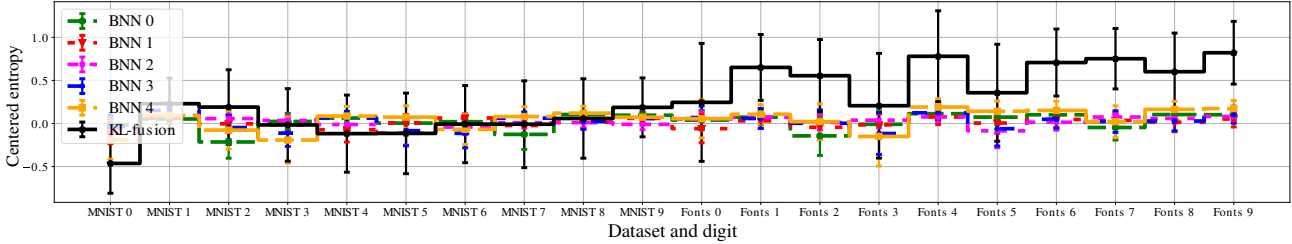
Figure 5. Entropy of local BNNs and KL-fused BNN across 10 digits of MNIST and Fonts centered at the corresponding mean MNIST entropies. Centered entropy above 0 indicates higher uncertainty comparative to that of the MNIST digits for a given BNN. Fused BNN has increased uncertainty over the Fonts digits, while local BNNs do not show such a trend.

Table 2. Comparison of local and fused BNNs

	Accuracy, %	Test ll	Entropy
KL-fusion	**95.8**	**-0.32**	**0.79**
BNN 0	90.1	-1.42	2.12
BNN 1	91.6	-1.44	2.13
BNN 2	82.7	-1.56	2.17
BNN 3	87.8	-1.37	2.08
BNN 4	91.9	-1.18	2.00

BNNs on a set of test images comprising of the standard MNIST test set as well as an out of distribution sample made up of computer generated digits in various fonts (Smirnov et al., 2020). We observe that while the local network's low entropy samples are heavily influenced by the local training data, the fused model is able to borrow statistical strength from all the local models and exhibits low entropy across all MNIST digits, reserving high entropy predictions for those digits from the OOD set whose fonts look very different from the hand drawn MNIST digits.

In Figure 5 we plot average entropies of the local BNNs and the KL-fused BNN across digits and datasets centering at the corresponding mean entropy on all MNIST digits. The Dirichlet based split of the train data resulted in dataset 0 receiving 2s (78% of all MNIST images of 2) and 7s (69%), dataset 4 received 62% of all 0s and 68% of 3s (other datasets are also skewed but to a lesser extent). As a result, we observe a lower entropy for 2s and 7s, in both MNIST and Fonts, displayed by BNN 0, and similar for BNN 4 on 0s and 3s. This observation suggests (as well as Figure 1) that these BNNs have learned to distinguish their dominant digits from the rest of the digits, rather than the desired BNN classifying all 10 digits and being uncertain on OOD examples. Our KL fusion method is able to produce a BNN with the desired properties from these biased local BNNs as entropy increases significantly on the OOD Fonts digits.

6. Conclusion

Federated learning techniques vary in complexity and communication overhead. On one extreme, some approaches hand information back and forth between different entities as they reach a consensus on the global model. On the other extreme, model fusion extracts a global model in a single shot: Local models are combined into the global model by solving a single optimization problem, and then the learning procedure is complete. Our technique and experiments show that model fusion can be effective despite its simplicity: In a single step, we extract a global model capturing relevant information from multiple local models.

The design of an effective fusion algorithm combines several key ideas. Working specifically with mean field approximations and exponential family distributions leads to a feasible algorithm while staying applicable to a wide array of practical scenarios, as illustrated by the examples in §5. This setup also allows our method to use information about the full local distributions, rather than point estimates as in previous work (Yurochkin et al., 2019a). Moreover, mixed norm regularization dynamically adjusts the dimensionality of our fused model. More broadly, model fusion in the Bayesian setting accompanies the fused model with uncertainty estimates, valuable for detecting out-of-distribution samples that are not captured by any of the individual local models as in Figure 1.

The success of KL-fusion suggests several avenues for future work. We likely can extend our algorithm to Bregman divergences other than KL using a similar formulation and algorithm; farther afield, optimal transport distances could improve the quality of our inference procedure but would likely require adjustment to the simplifications outlined in §3. We also could extend our method to handle *iterative* refinement, communicating the global model back to the local models as a means of improving the analysis of each component dataset.

KL-fusion seeks to find a global model to best approximate all of the local posteriors. In the distributed posterior inference literature the goal is often to approximate the posterior distribution of a pooled dataset assuming homogeneous data partition, e.g. Srivastava et al. (2015). Understading the connection between fused model and pooled data posterior is an interesting theoretical problem demanding new proof techniques to account for permutation invariance and data heterogeneity considered in our KL-fusion algorithm.

References

Attias, H. A variational Bayesian framework for graphical models. In *Advances in Neural Information Processing Systems*, pp. 209–215, 2000.

Banerjee, A., Dhillon, I. S., Ghosh, J., and Sra, S. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6: 1345–1382, September 2005.

Bardenet, R., Doucet, A., and Holmes, C. On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research*, 18(1):1515–1557, 2017.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Blei, D. M. and Jordan, M. I. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3: 993–1022, March 2003.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., and Jordan, M. I. Streaming variational Bayes. In *Advances in Neural Information Processing Systems*, pp. 1727–1735, 2013.

Bui, T. D., Nguyen, C. V., Swaroop, S., and Turner, R. E. Partitioned variational inference: A unified framework encompassing federated and continual learning. *arXiv preprint arXiv:1811.11206*, 2018.

Campbell, T. and How, J. P. Approximate decentralized Bayesian inference. *arXiv:1403.7471*, 2014.

Campbell, T., Straub, J., Fisher III, J. W., and How, J. P. Streaming, distributed variational inference for Bayesian nonparametrics. In *Advances in Neural Information Processing Systems*, pp. 280–288, 2015.

Carli, F. P., Ning, L., and Georgiou, T. T. Convex clustering via optimal mass transport. *arXiv:1307.5459*, 2013.

Ferguson, T. S. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, pp. 209–230, 1973.

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. An HDP-HMM for systems with state persistence. In *International Conference on Machine Learning*, pp. 312–319. ACM, 2008.

Fox, E. B., Hughes, M. C., Sudderth, E. B., and Jordan, M. I. Joint modeling of multiple time series via the beta process with application to motion capture segmentation. *The Annals of Applied Statistics*, pp. 1281–1313, 2014.

Ghosh, S., Yao, J., and Doshi-Velez, F. Structured variational learning of Bayesian neural networks with horseshoe priors. In *International Conference on Machine Learning*, pp. 1744–1753, 2018.

Ghosh, S., Yao, J., and Doshi-Velez, F. Model selection in Bayesian neural networks via horseshoe priors. *Journal of Machine Learning Research*, 20(182):1–46, 2019.

Hasenclever, L., Webb, S., Lienart, T., Vollmer, S., Lakshminarayanan, B., Blundell, C., and Teh, Y. W. Distributed bayesian learning with stochastic natural gradient expectation propagation and the posterior server. *The Journal of Machine Learning Research*, 18(1):3744–3780, 2017.

Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97, 1955.

Minka, T. P. Expectation propagation for approximate Bayesian inference. In *Conference on Uncertainty in Artificial Intelligence*, pp. 362–369, 2001.

Monteiller, P., Claici, S., Chien, E., Mirzazadeh, F., Solomon, J. M., and Yurochkin, M. Alleviating label switching with optimal transport. In *Advances in Neural Information Processing Systems*, pp. 13612–13622, 2019.

Neal, R. M. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1995.

Nguyen, X. Convergence of latent mixing measures in finite and infinite mixture models. *The Annals of Statistics*, 41 (1):370–400, 2013.

Nguyen, X. Posterior contraction of the population polytope in finite admixture models. *Bernoulli*, 21(1):618–646, 02 2015.

Opper, M. A Bayesian approach to on-line learning. *On-line Learning in Neural Networks*, pp. 363–378, 1998.

Rand, W. M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

Smirnov, D., Fisher, M., Kim, V. G., Zhang, R., and Solomon, J. Deep parametric shape predictions using

distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 561–570, 2020.

Srivastava, S., Cevher, V., Dinh, Q., and Dunson, D. Wasp: Scalable bayes via barycenters of subset posteriors. In *Artificial Intelligence and Statistics*, pp. 912–920, 2015.

Vinh, N. X., Epps, J., and Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., and Hoang, N. Statistical model aggregation via parameter matching. In *Advances in Neural Information Processing Systems*, pp. 10954–10964, 2019a.

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., and Khazaeni, Y. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pp. 7252–7261, 2019b.