Supplementary material for Model Fusion with Kullback–Leibler Divergence

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

1. Simulated experiments

In §5.1 of the main text we studied fusion of mixture model posteriors learned from heterogeneous datasets. Below we describe the data generating process used in these experiments.

First, we generate true global means $\mu_g \sim \mathcal{N}(0, I_D\sigma_0^2) \in \mathbb{R}^D$ for $g = 1, \dots, G$. We set true number of global components $G = 5$, data dimension $D = 10$, and entries of the diagonal covariance $\sigma_0^2 = sG$. Parameter s is the separation scale controlling the degree of separation between the true global means and corresponds to the x-axis in Figure 2 of the main text. To generate covariances $\{\Sigma_g\}_{g=1}^G$ of the global mixture components we used a slightly modified Scikit-learn (Pedregosa et al., 2011) function for random positive definite matrices (see code for details).[1]

To generate $J = 50$ heterogeneous datasets, we first assign a random probability to each global component. For each dataset j we then select a random subset of global means and covariances by drawing Bernoulli random variables with the corresponding probabilities. This corresponds to the Beta-Bernoulli process (Thibaux & Jordan, 2007). Next, to each selected mean we add Gaussian noise with standard deviation σ to enforce heterogeneity in the parameters. This σ is the x-axis in Figure 3 of the main text. We also slightly perturb covariances using Wishart distribution (see code for details). Finally, each dataset is generated from a Gaussian mixture with the corresponding means and covariances and mixture component probabilities drawn from a symmetric Dirichlet distribution. To illustrate the generative process for one dataset, in Figure 1 we give an example in $D = 2$ dimensions, with $G = 3$ global components, separation scale $s = 0.1$ and heterogeneity noise $\sigma = 0.1$. Our generative process resulted in a local mixture model with two components slightly perturbed from the corresponding global components.

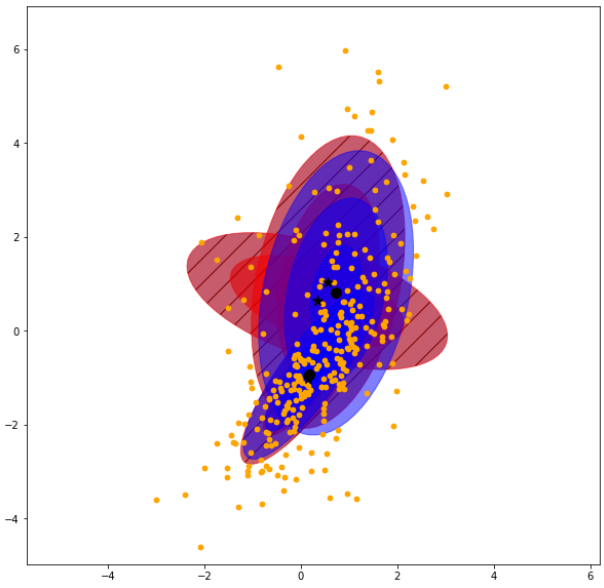


Figure 1. Illustration of the data simulation process in two dimensions: contours of the Gaussian densities corresponding to the global mixture components are in red with black stars as means. Components of the local mixture model are in blue with black circles marking means: our data generative process selected a subset of two global mixture components and perturbed means with some Gaussian noise to enforce heterogeneity. Corresponding local dataset is shown in orange.

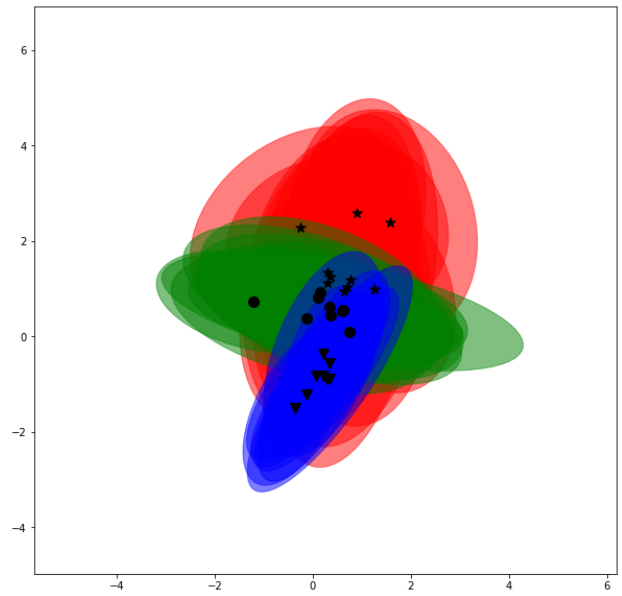
To obtain local posteriors for KL-fusion, on each of the datasets we ran variational inference with Gaussian-Wishart variational distributions.[2] Recall that our KL-fusion algorithm alternates between clustering of the components of the local posterior distributions and finding corresponding barycenters. In Figure 2(a) we show the result of the clustering step on one of the iterations of KL-fusion and in Figure 2(b) we present estimates of the fused mixture model means and covariances obtained from the corresponding barycenter.

[*]Equal contribution [1]CSAIL, MIT, Cambridge, Massachusetts, USA [2]MIT-IBM Watson AI Laboratory, Cambridge, Massachusetts, USA [3]IBM Research, Cambridge, Massachusetts, USA. Correspondence to: Sebastian Claici <sclaici@csail.mit.edu>, Mikhail Yurochkin <mikhail.yurochkin@ibm.com>.

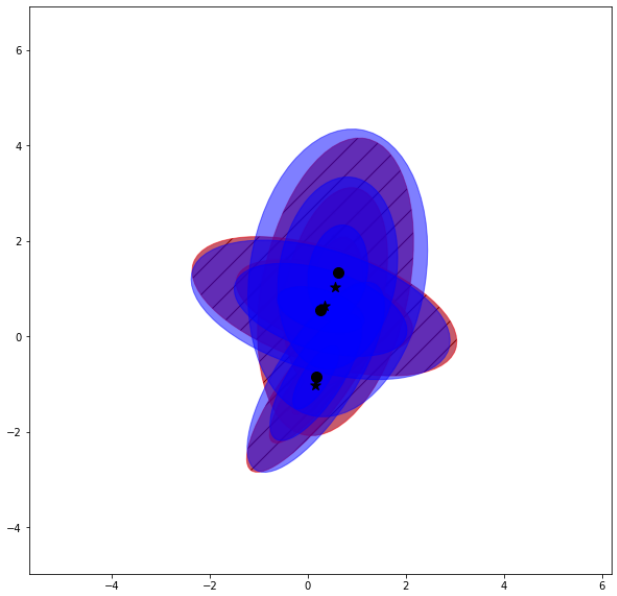
Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

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

[2]*sklearn.mixture.BayesianGaussianMixture*



(a) Clustering iteration of KL-fusion: after several iterations our algorithm learned meaningful clustering of the distributions corresponding to the posterior approximations of the local mixture components. Distributions in the same cluster are shown with the same color and mean marker. Our algorithm correctly identified that there should be 3 clusters.



(b) Fused posterior learned with KL-fusion: in blue we show estimates of the fused mixture model means and covariances obtained using our algorithm. This is the result of taking KL barycenter corresponding to the clustering on the left. In red we show true global means and covariances. KL-fusion produces an accurate estimate and estimates the size of the global model correctly.

Figure 2. Visualization of the KL-fusion algorithm in 2 dimensions

2. Topic modeling details

The problem setup for the topic modeling experiment was as follows: The 20news training dataset was split into 5 separate datasets based on topics of the news articles. For each local dataset, we fit a topic model with 10 topics. In this context, each component is a topic, the parameters for each component are the posterior Dirichlet variational parameters (one for each word), and the goal of fusion is to infer these posterior Dirichlet parameters for the global model. We then fused these posterior distributions using either AMPS (Campbell & How, 2014), the parametric version of our algorithm, or the non-parametric version of our algorithm.

For each document in the test set, we measure predictive log likelihood of the 10% of the words based on the remaining 90%. Predictive log likelihood is defined as in Wang et al. (2011), and approximated by

$$p(\mathbf{w}_{j2} \,|\, \mathbf{w}_{j1}, \mathcal{D}_{\text{train}}) = \prod_{w \in \mathbf{w}_{j2}} \sum_{k} \bar{\pi}_{jk} \bar{\phi}_{kw} \tag{1}$$

where $\bar{\pi}_{jk}$ is the proportion of topic k in document j, and $\bar{\phi}_{kw}$ is the posterior Dirichlet parameter of word w in topic k, $\mathbf{w}_{j2}$ are the held-out words. The log of this quantitity is summed over all documents in the test set.

3. HDP-HMM details

Our HMM models use multivariate Normal-Wishart observation models and Hierarchical Dirichlet process allocation models. The state specific transition probabilities π_k are drawn according to the following generative process. First, we draw $\beta \sim \text{GEM}(\gamma)$ from the stick breaking distribution. That is,

$$\beta_j = \nu_j \prod_{l=1}^{j-1} (1-\nu_l); \quad \nu_j \mid \gamma \sim \text{Beta}(1, \gamma); \quad j = 1, 2, \ldots, \tag{2}$$

We then draw π_k from a Dirichlet process with a discrete base measure shared across states,

$$\pi_k \mid \eta, \kappa, \beta \sim \text{DP}(\eta+\kappa, \frac{\eta\beta + \kappa\delta_k}{\eta + \kappa}); \quad k = 1, 2, \ldots, \tag{3}$$

where η is a concentration parameter and κ is a "stickyness" parameter (Fox et al., 2008) that encourages state persistence. The latent states for a particular sequence then evolve as z_t, evolve as $z_{t+1} \mid z_t, \{\pi_k\}_{k=1}^{\infty} \sim \pi_{z_t}$. Finally, observations at time step t, $y_t \in \mathbb{R}^D$ are drawn from a Normal Wishart distribution,

$$\begin{aligned} \mu_k \mid \mu_0, \lambda, \Lambda_k &\sim \mathcal{N}(\mu_0, (\lambda\Lambda_k)^{-1}) \\ \Lambda_k \mid S, n_0 &\sim \text{Wishart}(n_0, S) \\ y_t \mid z_t = k &\sim \mathcal{N}(y_t \mid \mu_k, \Lambda_k^{-1}) \end{aligned} \tag{4}$$

For all our experiments, we set κ to 10.0, $\gamma = 5.$ and $\eta = 0.5$. For the observation model, we set $n_0 = 1$ and S to an identity matrix I, encoding our belief that $\mathbb{E}[\Lambda_k^{-1}] = I$.

3.1. MoCAP data details

We consider the problem of discovering common structure in collections of related time series. Although such problems arise in a wide variety of domains, here we restrict our attention to data captured from motion capture sensors on joints of people performing exercise routines. We collected this data from the CMU MoCap database (`http://mocap.cs.cmu.edu`). Each motion capture sequence in this database consists of 64 measurements of human subjects performing various exercises. Following Fox et al. (2014), we select 12 measurements deemed most informative for capturing gross motor behaviors: 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 MoCAP sequence thus provides a 12-dimensional time series. We use a curated subset (Fox et al., 2014) of the data from two different subjects each providing three sequences. In addition to having several exercise types in common this subset comes with human annotated labels allowing for easy quantitative comparisons across different models.

4. Bayesian Neural Network Details

We use Bayesian neural networks with regularized horseshoe priors (Ghosh et al., 2019; 2018) as our local BNN models. In more detail, let a network with $L-1$ hidden layers be parameterized by a set of weight matrices $\mathcal{W} = \{W_l\}_1^L$, where each weight matrix W_l is of size $\mathbb{R}^{(K_{l-1}+1)\times K_l}$, and K_l is the number of units (excluding the bias) in layer l. Let the node weight vector $w_{kl} \in \mathbb{R}^{(K_{l-1}+1)}$ denote the set of weights incident to unit k of hidden layer l. Following Ghosh et al. (2019; 2018), we place regularized horseshoe priors over w_{kl}:

$$w_{kl} \mid \tau_{kl}, \upsilon_l, c \sim \mathcal{N}(0, (\tilde{\tau}_{kl}^2 \upsilon_l^2) I), \quad \tilde{\tau}_{kl}^2 = \frac{c^2 \tau_{kl}^2}{c^2 + \tau_{kl}^2 \upsilon_l^2}, \tag{5}$$

with $\tau_{kl} \sim C^+(0, b_0)$, $\upsilon_l \sim C^+(0, b_g)$, and $c \sim$ Inv-Gamma(c_a, c_b). Here, I is an identity matrix and $a \sim C^+(0, b)$ is the half-Cauchy distribution with density $p(a|b) = 2/\pi b(1 + (a^2/b^2))$ for $a > 0$; τ_{kl} is a unit specific scale parameter, while the scale parameter υ_l is shared across layer l. The regularized horseshoe distribution exhibits a spike at zero that provides strong shrinkage towards zero and encourages sparsity by turning off nodes in a layer that are not necessary for explaining the data. This allows the local BNNs to automatically select the appropriate number of nodes in each layer.

We resort to variational inference on an equivalent parameterization, $w_{kl} = \tilde{\tau}_{kl} \upsilon_l \beta_{kl}, \quad \beta_{kl} \sim \mathcal{N}(0, I)$. After learning the variational posteriors of this equivalent model, we arrive at $q(w_{kl})$ by first approximating the factorized variational approximations $q(c), q(\tau_{kl}), q(\upsilon_l)$ with their expected values $\mu_c, \mu_{\tau_{kl}}, \mu_{\upsilon_l}$, and defining

$$\mu_{\tilde{\tau}_{kl}}^2 = \frac{\mu_c^2 \mu_{\tau_{kl}}^2}{\mu_c^2 + \mu_{\tau_{kl}}^2 \mu_{\upsilon_l}^2}.$$

Given the expected values and the Gaussian variational distribution $q(\beta_{kl}) = \mathcal{N}(\mu_{\beta_{kl}}, \Psi_{\beta_{kl}})$, $w_{kl} = \tilde{\tau}_{kl} \upsilon_l \beta_{kl}$ follows a Gaussian distribution $\mathcal{N}(\mu_{w_{kl}}, \Sigma_{w_{kl}})$, with

$$\mu_{w_{kl}} = \mu_{\tilde{\tau}_{kl}} \mu_{\upsilon_l} \mu_{\beta_{kl}}; \quad \Sigma_{w_{kl}} = \mu_{\tilde{\tau}_{kl}}^2 \mu_{\upsilon_l}^2 \Psi_{\beta_{kl}}.$$

We thus recover the variational approximation on w_{kl}, $q(w_{kl}) = \mathcal{N}(\mu_{w_{kl}}, \Sigma_{w_{kl}})$. These variational distributions from different local BNNs are then used for fusion.

5. Initialization

To initialize KL-fusion algorithm we used a variation of k-means++ initialization as discussed in the conclusion of Arthur & Vassilvitskii (2006). This is a popular initialization scheme used in Scikit-learn (Pedregosa et al., 2011) for k-means clustering. In our KL-fusion initialization, we replaced squared Euclidean distance with KL divergence.

References

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

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

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.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., and Dubourg, V. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

Thibaux, R. and Jordan, M. I. Hierarchical Beta processes and the Indian buffet process. In *Artificial Intelligence and Statistics*, pp. 564–571, 2007.

Wang, C., Paisley, J., and Blei, D. Online variational inference for the hierarchical Dirichlet process. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 752–760, 2011.