Prediction-focused Mixture Models

Sanjana Narayanan [*1] **Abhishek Sharma** [*1] **Catherine Zeng** [1] **Finale Doshi-Velez** [1]

Abstract

In several applications, besides getting a generative model of the data, we also want the model to be useful for specific downstream tasks. Mixture models are useful for identifying discrete components in the data, but may not identify components useful for downstream tasks if misspecified; further, current inference techniques often fail to overcome misspecification even when a supervisory signal is provided. We introduce the prediction-focused mixture model, which selects and models input features relevant to predicting the targets. We demonstrate that our approach identifies relevant signal from inputs even when the model is highly misspecified.

1. Introduction

In many situations, it is valuable to have a generative model that can also be used to predict specific outcomes of interest. For example, we might be interested in grouping clinical phenotypes and biological markers that are correlated with a clinical outcome like mortality due to sepsis (Seymour et al., 2019). In this context, a generative model elegantly handles the missingness pervasive in medical datasets (Cios & Moore, 2002) and can be used to simulate data which can indicate to experts where the model might be going wrong. Despite these benefits, when it comes to predicting outcomes, discriminative models often outperform generative models (Ng & Jordan, 2002). In this work, we focus on learning a generative model while maintaining good predictive performance, even when the model has limited capacity, i.e. when the model does not have enough parameters to reproduce the data completely.

The main challenge of this task is to balance the trade-off between the generative modeling objective and the prediction objective. While maximizing likelihood of the observed data may use the limited model capacity to reproduce part of the data space well, the resulting model is not guaranteed to have good performance on the predictive task. Thus, a two-step procedure of learning a model using the observed data and then using the trained model on the prediction tasks will generally not perform well. It is necessary—but not sufficient—to jointly model the data and the prediction signal.

In this work, we introduce the prediction-focused mixture model (pfGMM): a model which provides the desired trade-off between generative and predictive performance and is tunable using a single interpretable hyperparameter. Our model selects features that are relevant to the downstream task, and explains these features well even in settings where it is misspecified. We further demonstrate that this behavior is a property of the model itself, and not due to any posterior approximation or a special objective. Finally, we validate the behavior of our model in a variety of simulated settings: our model selects useful features from the input and performs well on predicting targets, while also learning a high-density model of these features.

2. Related Work

Semi-supervised learning aims to augment generative modeling of inputs $\mathbf{x}$ with a supervisory signal y usually by training the joint objective $\log p(\mathbf{x}, y)$ (Nigam et al., 1998; Kingma et al., 2014; Ghahramani & Jordan, 1994). This approach fails to recognize the inherent asymmetry between the inputs and the targets, and therefore often ignores the target dimension as the model allocates its capacity to the more structured inputs. We address this issue by incorporating specific input-focused latent variables that treat the inputs and targets differently. Methods that incorporate an external signal to aid clustering of data exist under the paradigm of "supervised clustering" but they fail to provide a density of the observed data.

(Hughes et al., 2017; 2018) proposed **prediction-constrained learning** in the context of mixture and topic models, noting the asymmetry of the task in predicting labels from documents and not the other way round. The framework attempts to balance the maximum likelihood objective with predictive performance by introducing the latter as a constraint and then solving the resulting con-

[*]Equal contribution [1]SEAS, Harvard University, Cambridge, USA. Correspondence to: Abhishek Sharma <abhishek-sharma@g.harvard.edu>.

ICML 2021 Workshop on Information - Theoretic Methods for Rigorous, Responsible, and Reliable Machine Learning.

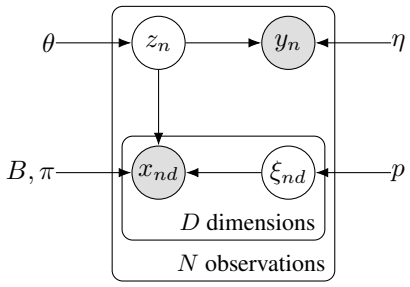


Figure 1. The pfGMM Graphical Model.

strained optimization problem. But the resulting constrained objective is often hard to optimize in practice, and does not correspond to maximum likelihood of a valid graphical model. In the context of topic models for text documents, Ren et al. (2020) find a case where optimization is easier by proposing a model which excludes or includes words depending on their relevance for prediction. But they rely on orthogonality of relevant and irrelevant topic vectors for the model to work—a condition which is specific to topic models and discrete distributions of the words. Our pfGMM does not rely on such conditions, and can work with both discrete and continuous observations.

Information bottleneck (IB) methods take a joint distribution $p(\mathbf{x}, \mathbf{y})$ and compress the inputs $\mathbf{x}$ into a latent code $\mathbf{z}$ that is most informative about the targets $\mathbf{y}$ (Tishby et al., 2000). Both the traditional IB and the deep variational IB formulations assume specific Markov structures for the generative model (Alemi et al., 2016; Wieczorek & Roth, 2020). In contrast, we achieve the trade-off between compressing inputs and predicting targets by proposing a specific probabilistic model which encodes our knowledge about the generative process. We show that this model is sufficient for getting the trade-off we seek, independent of the IB objective.

3. Prediction-focused Mixture Models

We now introduce prediction-focused mixture model, a mixture model that can select useful features which exhibit discrete latent structure (Fig. 1). Our first assumption is that the input $\mathbf{x}$ can be partitioned as being "relevant" and "irrelevant" dimensions, where "relevant" captures covariates that are predictive of the targets $\mathbf{y}$ while the "irrelevant" dimensions are not. Another assumption is about the nature of model misspecification - we assume that "irrelevant" part of the covariates may be misspecified. The generative process of the model is:

for each datum n **do**
  Draw latent state $z_n \sim$ **Cat**(θ)
  for each dimension d **do**
    Draw switch $\xi_{nd} \sim$ **Bern**(p)
    if $\xi_{nd} = 1$ **then**
      Draw observation $x_{nd} \sim F_X(x_{nd}|z_n; B_{z_n,d})$
    else if $\xi_{nd} = 0$ **then**
      Draw (irrelevant) observation $x_{nd} \sim F_X(x_{nd}; \pi_d)$
    end if
  end for
  Draw target $y_{nd} \sim F_Y(y_{nd}; \eta_{z_n})$
end for

where F_X, F_Y are usually chosen to be Gaussian or Categorical distributions but can also be other distributions depending on the data type. In our paper, we choose both of them to be Gaussian, and hence refer to the model as pfGMM. We show that pfGMM uses the switch variables ξ_d to filter out "irrelevant" dimensions even when it is (inevitably) misspecified with respect to the data generating process. This "prediction-focus" property is true even when the irrelevant dimensions are more structured than the relevant dimensions.

Hyperparameters The pfGMM requires us to tune only a single hyperparameter: the switch prior p. It encodes our belief about roughly how many dimensions should be relevant. From a model training point of view, it also trades off generative quality with predictive performance.

Robustness to misspecification Any real world analysis faces a situation where the proposed model's generative process doesn't match the true data generating process. We design the pfGMM to be robust to some of such misspecifications. While an arbitrarily high number of components can explain the data well, it is reasonable to expect only a few to be useful for the prediction task. We consider misspecification in the number of mixture components and ensure that the model prioritizes the use of components to learn those that are relevant. This property seeks out data dimensions that are relevant to the task at hand, instead of just picking out those that have more structure.

4. Inference

Since pfGMM is a valid graphical model, we can use the maximum likelihood objective to do learning and inference. The log likelihood of the pfGMM is:

$$\log p(\mathbf{x}, \mathbf{y}) = \log \sum_{\mathbf{z}, \boldsymbol{\xi}} p_\theta(\mathbf{z}) p_p(\boldsymbol{\xi}) p_{B,\pi}(\mathbf{x}|\mathbf{z}, \boldsymbol{\xi}) p_\eta(\mathbf{y}|\mathbf{z}) \quad (1)$$

which is intractable to compute. We empirically verify (for lower dimensional inputs) that maximizing the log likelihood gives us good predictive performance (as measured by target RMSE)—suggesting that the pfGMM model can balance generative and predictive performance.

We optimize the evidence lower bound (ELBO) as a

tractable lower bound to Eq. 1. We choose the following form for the approximate posterior:

$$q(\mathbf{z}, \boldsymbol{\xi}|\Theta, \varphi) = \prod_{n=1}^{N} q(z_n|\mathbf{x}_n, y_n, \varphi, \Theta) \prod_{d=1}^{D} q(\xi_{nd}|\varphi_d)$$

where d indexes the input dimension, n indexes the exchangeable data samples and $\Theta = \{\theta, B, \pi, \eta\}$ refers to the model parameters. The distribution $q(\xi_{nd}|\varphi_d)$ is parameterized as a Bernoulli distribution with a *tied* parameter φ_d for dimension d. The distribution $q(z_n|\mathbf{x}_n, y_n, \varphi, \Theta)$ is simply the posterior distribution conditioned on the variational parameters φ and can be computed as:

$$q(z = k|\mathbf{x}, y, \varphi) \propto p_\theta(z = k) \cdot p(\mathbf{x}|z = k, \varphi) \cdot p_\eta(y|z = k)$$

At prediction time, the distribution $q(z = k|\mathbf{x}, y, \varphi)$ is computed without the $p_\eta(y|z = k)$ term and the *feature selection parameter* φ helps select the right features in the term $p(\mathbf{x}|z = k, \varphi)$.

This results in the following ELBO objective (full derivation in Appendix A):

$$\begin{aligned}\text{ELBO}(q) = &\mathop{\mathbb{E}}_{q(\mathbf{z},\boldsymbol{\xi})} \left[\log p_\Theta(\mathbf{x}, \mathbf{y}|\mathbf{z}, \boldsymbol{\xi})\right] \\ &- \mathbf{KL}\left(q(\mathbf{z}, \boldsymbol{\xi}|\Theta, \varphi)||p_\Theta(\mathbf{z}, \boldsymbol{\xi})\right) \end{aligned} \quad (2)$$

which is tractable to compute because the posterior factorizes over $\boldsymbol{\xi}$ and $\mathbf{z}$.

Alternatively to the ELBO, (Ren et al., 2020; Hughes et al., 2018) proposed bounding the log likelihood to get an objective that emphasizes the connection between prediction-focused and prediction-constrained learning (full derivation in Appendix B):

$$\begin{aligned}\log p(\mathbf{x}, \mathbf{y}) \geq &\mathop{\mathbb{E}}_{p(\boldsymbol{\xi})} \left[\log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\xi})\right] + p \mathop{\mathbb{E}}_{p(\mathbf{z})} \left[\log p_B(\mathbf{x}|\mathbf{z})\right] \\ &+ (1 - p) \log p_\pi(\mathbf{x}) \end{aligned} \quad (3)$$

where the first term explicitly improves the prediction quality while the next two terms improve the density of the inputs. Nevertheless, this bound has not been experimentally verified to retain the desirable properties of the original MLE objective (Eq. 1). In our experiments, we show that this bound indeed balances the trade-off – achieving optimal predictions while achieving high likelihood for the observed dimensions. We refer to this bound as the 'alternate lower bound' (AltLB for short) from hereon.

While AltLB performs better than all other methods we looked at, it is intractable since the first term in Eq. 3 scales exponentially with the number of dimensions.

5. Experiments

In this section, we demonstrate that the pfGMM can select relevant signal from the input dimensions in both misspecified and fully specified data settings. We demonstrate that the model structure – involving the switch parameters and their prior – is able to recover useful dimensions and maintains the downstream performance even under severe misspecification.

5.1. Experimental Setup

Baselines We compare the generative and predictive performance of pfGMM trained by optimizing the ELBO against (1) an approach that first learns a generative model using only the inputs (corresponding to the model without y in Fig. 1) and then predicts the outputs ("2-step"), (2) a Gaussian mixture model without the switch parameters (GMM), (3) an approach that maximizes the log likelihood $\log p(\mathbf{x}, \mathbf{y})$ (pfGMM-LL), (4) an approach that maximizes the AltLB, which explicitly trades-off generative and predictive performance (pfGMM-AltLB), (5) and a "discriminative" approach that only maximizes the condition likelihood $\log p(\mathbf{y}|\mathbf{x})$ (pred-only).

Data We generate simulated data to compare the models on varying levels of misspecification we might encounter in real-world datasets. In our experiments, the model is misspecified whenever the noise dimensions are not distributed according to a fixed Gaussian. We generate the relevant input and target dimensions from the mixture distribution $.33\mathcal{N}(0, 1) + .66\mathcal{N}(6, 1)$. The irrelevant dimensions are generated from the distribution $.33\mathcal{N}(0, 1) + .66\mathcal{N}(\mu_{\textbf{gap}}, 1)$ where $\mu_{\textbf{gap}}$ is varied in $\{0, 6, 8\}$. This simulates a diverse set of datasets with varying signal-to-noise ratios and structures in irrelevant dimensions - some are easy to predict while others are much harder.

To evaluate prediction quality, we compare the RMSE for the continuous target dimension on heldout data. To further assess models on their generative and discriminative quality, we compute $\log p(\mathbf{x})$ and $\log p(\mathbf{y}|\mathbf{x})$ on heldout data (computed without any approximations to ensure consistent evaluation between the methods). Due to the cost of computing LL and AltLB with the number dimensions as $\mathcal{O}(D^3)$, we restrict our comparison to these objectives only when $D = 10$.

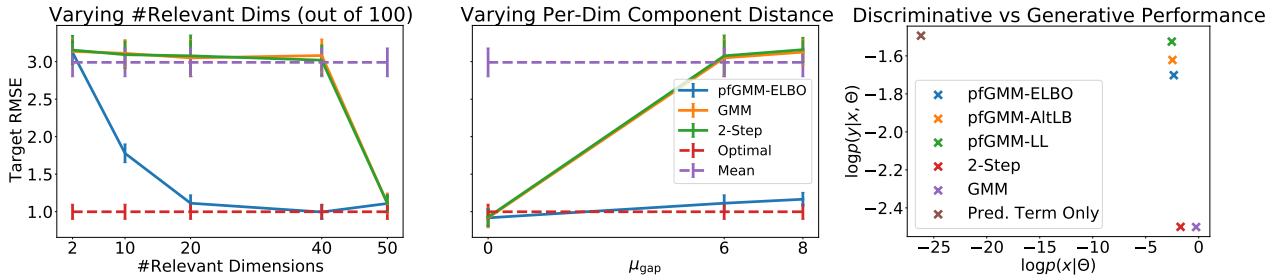
5.2. Results

LL, ELBO and AltLB objectives for pfGMM perform comparably at prediction We see from Table 1 that pfGMM trained using the ELBO objective matches the target RMSE of AltLB and LL objectives even when other methods find it hard to predict targets.

Also in Fig. 2b, we see that pfGMM achieves good discriminative and generative performance (as measured by corresponding likelihoods) when compared on the discriminative and generative objective-plane. This also provides evidence that the "prediction-focus" property is a virtue of

Target Mean and (SD) RMSE Across Different Levels of Misspecification										
Relevant Dims out of Total:	5 out of 10		2 out of 10		50 out of 100		10 out of 100		2 out of 100	
Component Mean Gap, μ_{gap} :	6	8	6	8	6	8	6	8	6	8
2-STEP	**0.9**(0.0)	**1.1**(0.1)	3.1(0.1)	3.0(0.1)	**1.1**(0.0)	3.1(0.1)	3.1(0.1)	3.1(0.1)	**3.1**(0.1)	3.0(0.1)
GMM	**0.9**(0.1)	3.1(0.1)	3.1(0.1)	3.0(0.1)	**1.1**(0.1)	3.1(0.1)	3.1(0.1)	3.2(0.1)	**3.2**(0.1)	3.0(0.1)
pfGMM-ALTLB	1.0(0.0)	**1.1**(0.1)	**1.2**(0.1)	1.2(0.0)	-	-	-	-	-	-
pfGMM-LL	**0.9**(0.1)	**1.1**(0.1)	**1.0**(0.1)	3.0(0.1)	-	-	-	-	-	-
pfGMM-ELBO	1.0(0.0)	**1.1**(0.0)	1.6(0.1)	1.5(0.1)	**1.1**(0.0)	**1.1**(0.1)	**1.8**(0.1)	**1.7**(0.1)	**3.1**(0.1)	**2.7**(0.1)
PRED-ONLY	1.0(0.0)	**1.1**(0.0)	**1.1**(0.1)	**1.0**(0.1)	-	-	-	-	-	-

Table 1. Mean and (SD) RMSE of the target across 10 bootstrap evaluations on the heldout test set. We **bold** every RMSE value within 1 SD of the smallest 2-SD interval. The first "level" varies the number of relevant dimensions and the second "level" varies the distance between components means of irrelevant dimensions in the data. Each level becomes harder to predict from left to right because the misspecification increases. pfGMMis more robust in the face of misspecification – degrading in performance only when the misspecification is severe (i.e. 2 dimensions out of 100 are relevant) – while continuing to do well when there is no misspecification.



(a) (b)

Figure 2. (a) Test RMSE as the model misspecification changes. *Left*: Varying the number of relevant dimensions in the input. All methods can do well when half the dimensions are relevant, but pfGMM can learn to select relevant dimensions even when they are greatly outnumbered by irrelevant dimensions. The irrelevant dimensions come from a GMM with the same marginal density as the relevant dimensions. *Right*: Varying the component distance, μ_{gap} while fixing relevant distribution and D_{rel}= 20. Increasing μ_{gap} increases the structure in irrelevant dimensions, and μ_{gap} = 0 corresponds to the case when there is no misspecification. pfGMM has good predictive performance even when each noise dimension has more structure than a signal dimension marginally. $D = 100$ for both the plots. (b) Each method's best model compared on the discriminative vs generative fitness landscape. The x-axis is the log likelihood of a test input per dimension, and the y-axis is the conditional log likelihood of the test set targets given the inputs. The input data has 10 dimensions, out of which only 2 are "relevant" .

the model and its LL objective, and not of the approximate posterior or of these bounds to the LL - a fact that previous work did not resolve. We further note that the two bounds merely retain this property.

We also present results for the case when irrelevant dimensions outnumber the relevant dimensions and are marginally more structured (μ_{gap} is higher) than relevant dimensions. This is a case which is hard for all models, and we also see a drop in performance of pfGMM. Improving this is an avenue for future work.

pfGMM outperforms 2-step training and GMM at prediction We see from Table 1 that pfGMM is at least as good as 2-step training in all datasets. When there is no misspecification in the data, or when the number of relevant dimensions is comparable to the number of irrelevant dimensions, the 2-step and GMM procedures have no problem in learning parameters that predict the targets well. But whenever the irrelevant dimensions exhibit more structure i.e. when irrelevant dimensions outnumber relevant dimensions or when irrelevant component means are better separated, 2-step approach fails (Fig. 2a). This is because the objective prefers to learn parameters which maximize likelihood and ignore the relevance to predicting the targets. In contrast, pfGMM has superior performance because it is able to trade off generative performance with predictive performance due to the combination of its hyperparameter p and the joint training of inputs and targets. Looking at the other baseline, pfGMM outperforms the jointly trained 'semi-supervised' GMM because it treats the targets and the input dimensions equally. pfGMM treats the targets and inputs differently, associating the switches only with the input dimensions.

pfGMM only selects features relevant for prediction The pfGMM marginal switch posterior $p(\xi_d|\mathbf{x}, \mathbf{y})$ provides an interpretable notion of feature relevance. pfGMM consistently gives higher density to features which are useful for predicting the targets. The 2-Step method is unable to recover the relevant features because the posterior $p(\xi_d|\mathbf{x})$ does not contain information about the targets. The GMM does not have switch variables which it can use to store relevance information about the inputs.

6. Discussion

In this paper we propose a prediction-focused mixture model that selects the 'right' features to cluster on even when it is faced with misspecification. We define 'right' as features that are relevant for the specific prediction task we're interested in. The pfGMM is correctly specified in that the LL objective performs the generative-predictive trade-off without appealing to the orthogonality of topics or to a specific form of the posterior as in (Ren et al., 2020) for good performance. We also show that model is able to achieve the trade-off without constraining the LL objective unlike in (Hughes et al., 2018). We also empirically validate the usefulness of the "alternate lower bound" but use ELBO as a tractable objective for learning prediction-focused models.

Acknowledgements AS acknowledges support from NSF project 2007076. FDV acknowledges support from NSF project 1750358.

References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Cios, K. J. and Moore, G. W. Uniqueness of medical data mining. *Artificial intelligence in medicine*, 26(1-2):1–24, 2002.

Ghahramani, Z. and Jordan, M. I. Supervised learning from incomplete data via an em approach. In *Advances in neural information processing systems*, pp. 120–127, 1994.

Hughes, M., Hope, G., Weiner, L., McCoy, T., Perlis, R., Sudderth, E., and Doshi-Velez, F. Semi-supervised prediction-constrained topic models. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1067–1076. PMLR, 09–11 Apr 2018.

Hughes, M. C., Weiner, L., Hope, G., McCoy Jr, T. H., Perlis, R. H., Sudderth, E. B., and Doshi-Velez, F. Prediction-constrained training for semi-supervised mixture and topic models. *arXiv preprint arXiv:1707.07341*, 2017.

Kingma, D. P., Rezende, D. J., Mohamed, S., and Welling, M. Semi-supervised learning with deep generative models. *arXiv preprint arXiv:1406.5298*, 2014.

Ng, A. Y. and Jordan, M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pp. 841–848, 2002.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. M. Learning to classify text from labeled and unlabeled documents. In Mostow, J. and Rich, C. (eds.), *AAAI/IAAI*, pp. 792–799. AAAI Press / The MIT Press, 1998. ISBN 0-262-51098-7.

Ren, J., Kunes, R., and Doshi-Velez, F. Prediction focused topic models via feature selection. In *International Conference on Artificial Intelligence and Statistics*, pp. 4420–4429. PMLR, 2020.

Seymour, C. W., Kennedy, J. N., Wang, S., Chang, C.-C. H., Elliott, C. F., Xu, Z., Berry, S., Clermont, G., Cooper, G., Gomez, H., and et al. Derivation, validation, and potential treatment implications of novel clinical phenotypes for sepsis. *JAMA*, 321(20):2003–2017, 2019. ISSN 0098-7484. doi: 10.1001/jama.2019.5791.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Wieczorek, A. and Roth, V. On the difference between the information bottleneck and the deep information bottleneck. *Entropy*, 22(2):131, Jan 2020. ISSN 1099-4300. doi: 10.3390/e22020131. URL http://dx.doi.org/10.3390/e22020131.

Prediction-focused Mixture Models: Supplemental Work

A. ELBO Derivation

$$\begin{aligned}
\log p(\mathbf{x}, y) &\geq \mathop{\mathbb{E}}_{z,\boldsymbol{\xi}}\left[\frac{\log p(z, \boldsymbol{\xi}, \mathbf{x}, y|\Theta)}{q(z, \boldsymbol{\xi}|\Theta, \varphi)}\right] \\
\textbf{ELBO}(\Theta, \varphi, p(z|\mathbf{x}, y, \varphi)) &= \mathop{\mathbb{E}}_{z,\boldsymbol{\xi}}\left[\log p(\mathbf{x}, y|z, \boldsymbol{\xi}, \Theta)\right] - \textbf{KL}\left(q(z, \boldsymbol{\xi}|\Theta, \varphi)||p(z, \boldsymbol{\xi}|\theta, p)\right) \\
&= \mathop{\mathbb{E}}_{z,\boldsymbol{\xi}}\left[\log p(\mathbf{x}|z, \boldsymbol{\xi}, B, \pi)\right] + \mathop{\mathbb{E}}_{q(z)}\left[\log p(y|z, \eta)\right] \\
&+ \mathop{\mathbb{E}}_{q(z)}\left[\log p(z|\theta)\right] - \mathop{\mathbb{E}}_{q(z)}\left[\log q(z|\mathbf{x}, y, \varphi, \Theta)\right] \\
&+ \mathop{\mathbb{E}}_{q(\boldsymbol{\xi})}\left[\log p(\boldsymbol{\xi}|p)\right] - \mathop{\mathbb{E}}_{q(\boldsymbol{\xi})}\left[\log q(\boldsymbol{\xi}|\varphi)\right]
\end{aligned}$$

Writing $q(z|\mathbf{x}, y, \Theta, \varphi) = q(z)$ for simplicity, the individual ELBO terms are:

$$\begin{aligned}
\mathop{\mathbb{E}}_{q}\left[\log p(\mathbf{x}|z, \boldsymbol{\xi}, B, \pi)\right] &= \mathop{\mathbb{E}}_{q(\boldsymbol{\xi}|\varphi)q(z|\mathbf{x},y,\Theta,\varphi)}\left[\sum_{d=1}^{D} \xi_d \log p_B(x_d|z, B) + (1 - \xi_d) \log p_\pi(x_d|\pi)\right] \\
&= \sum_{d=1}^{D} \varphi_d \mathop{\mathbb{E}}_{q(z)}\left[\log p_B(x_d|z, B)\right] + (1 - \varphi_d) \log p_\pi(x_d|\pi)
\end{aligned}$$

$$\mathop{\mathbb{E}}_{q}\left[\log p(y|z, \boldsymbol{\xi}, B, \pi)\right] = \mathop{\mathbb{E}}_{q(z|\mathbf{x},y,\Theta,\varphi)}\left[\log p(y|z, \eta^\mu, \eta^V)\right] = \mathop{\mathbb{E}}_{q(z)}\left[\log p(y|z, \eta^\mu, \eta^V)\right]$$

$$\mathop{\mathbb{E}}_{q}\left[\log p(z|\theta)\right] - \mathop{\mathbb{E}}_{q}\left[\log q(z|\mathbf{x}, y, \varphi, \Theta)\right] = \mathop{\mathbb{E}}_{q(z)}\left[\log \frac{p(z|\theta)}{q(z|\mathbf{x}, y)}\right]$$

$$\mathop{\mathbb{E}}_{q}\left[\log p(\boldsymbol{\xi}|p)\right] - \mathop{\mathbb{E}}_{q}\left[\log q(\boldsymbol{\xi}|\varphi)\right] = \mathop{\mathbb{E}}_{q(\boldsymbol{\xi}|\varphi)}\left[\sum_{d=1}^{D} \log \frac{p(\boldsymbol{\xi}_d|p)}{q(\boldsymbol{\xi}_d|\varphi_d)}\right] = \sum_{d=1}^{D} \varphi_d \log \frac{p}{\varphi_d} + (1 - \varphi_d) \log \frac{1 - p}{1 - \varphi_d}$$

We use either batch gradient descent or coordinate ascent to solve for parameters that maximize the ELBO.

B. AltLB Derivation

As in Ren et al., we find a lower bound for $p(y, \mathbf{x})$ (Ren et al., 2020). Applying Law of Total Probability, we can introduce $\boldsymbol{\xi}$:

$$\begin{aligned}
p(y, \mathbf{x}) &= \mathop{\mathbb{E}}_{p(\boldsymbol{\xi})}\left[p(y|\mathbf{x}, \boldsymbol{\xi}) p(\mathbf{x}|\boldsymbol{\xi})\right] \\
\log p(y, \mathbf{x}) &= \log \mathop{\mathbb{E}}_{p(\boldsymbol{\xi})}\left[p(y|\mathbf{x}, \boldsymbol{\xi}) p(\mathbf{x}|\boldsymbol{\xi})\right]
\end{aligned}$$

By Jensen's inequality:

$$\log p(y, \mathbf{x}) \geq \mathop{\mathbb{E}}_{p(\boldsymbol{\xi})} [\log p(y|\mathbf{x}, \boldsymbol{\xi})] + \mathop{\mathbb{E}}_{p(\boldsymbol{\xi})} [\log p(\mathbf{x}|\boldsymbol{\xi})]$$

Applying Law of Total Probability to introduce z:

$$\begin{aligned}\log p(\mathbf{x}|\boldsymbol{\xi}) &= \log \sum_z p(\mathbf{x}|z, \boldsymbol{\xi}) p(z) \\ &= \log \mathop{\mathbb{E}}_{p(z)} \left[\prod_d p_B(x_d|z)^{\xi_d} p_\pi(x_d)^{1-\xi_d} \right]\end{aligned}$$

Again, by Jensen's inequality:

$$\log p(\mathbf{x}|\boldsymbol{\xi}) \geq \mathop{\mathbb{E}}_{p(z)} \left[\sum_d \xi_d \log p_B(x_d|z) + (1-\xi_d) \log p_\pi(x_d) \right]$$

Substituting it into the original equation and using the fact that $\boldsymbol{\xi}$ is independent of z:

$$\begin{aligned}\log p(\mathbf{x}, y) &\geq \mathop{\mathbb{E}}_{p(\boldsymbol{\xi})} [\log p(y|\mathbf{x}, \boldsymbol{\xi})] + \mathop{\mathbb{E}}_{p(\boldsymbol{\xi})p(z)} \left[\sum_d \xi_d \log p_B(x_d|z) + (1-\xi_d) \log p_\pi(x_d) \right] \\ &= \mathop{\mathbb{E}}_{p(\boldsymbol{\xi})} [\log p(y|\mathbf{x}, \boldsymbol{\xi})] + \sum_d p \mathop{\mathbb{E}}_{p(z)} [\log p_B(x_d|z)] + (1-p) \log p_\pi(x_d)\end{aligned}$$

Since y is independent of x_d whenever $\xi_d = 0$ (those observations do not come from z), we can write the first term as:

$$\begin{aligned}\mathop{\mathbb{E}}_{p(\boldsymbol{\xi})} [\log p(y|\mathbf{x}, \boldsymbol{\xi})] &= \sum_{\xi_1} \dots \sum_{\xi_D} p(\xi_1, \dots \xi_D) \log p(y|\mathbf{x}, \boldsymbol{\xi}) \\ &= \sum_{\xi_1} \dots \sum_{\xi_D} \prod_d p(\xi_d) \log p(y|\{x_d : \xi_d = 1\}, \{x_d : \xi_d = 0\}, \boldsymbol{\xi})\end{aligned}$$

We consider d_1 as the set $\{d : \xi_d = 1\}$, x_{d_1} as the set $\{x_d : \xi_d = 1\}$, d_0 as the set $\{d : \xi_d = 0\}$, and x_{d_0} as the set $\{x_d : \xi_d = 0\}$.

$$\mathop{\mathbb{E}}_{p(\boldsymbol{\xi})} [\log p(y|\mathbf{x}, \boldsymbol{\xi})] = \sum_{\xi_1} \dots \sum_{\xi_D} \prod_{d_1} p \prod_{d_0} (1-p) \log p(y|x_{d_1}, \xi_{d_1} = 1)$$

We now compute the likelihood term in the last expression as:

$$\begin{aligned}\log p(y|x_{d_1}, \xi_{d_1} = 1) &= \log \left[\sum_z p(z|x_{d_1}, \xi_{d_1} = 1) p(y|z, x_{d_1}, \xi_{d_1} = 1) \right] \\ &= \log \left[\sum_z \frac{p(x_{d_1}|z, \xi_{d_1} = 1) p(z|\xi_{d_1} = 1)}{p(x_{d_1}|\xi_{d_1} = 1)} p(y|z) \right] \\ &= \log \sum_z p_B(x_{d_1}|z) p(z) p(y|z) - \log \sum_{z'} p_B(x_{d_1}|z') p(z')\end{aligned}$$

So our final derivation for the alternate lower bound is:

$$\log p(\mathbf{x}, y) \geq \mathop{\mathbb{E}}_{p(\boldsymbol{\xi})} [\log p(y|\mathbf{x}, \boldsymbol{\xi})] + \sum_d p \mathop{\mathbb{E}}_{p(z)} [\log p_B(x_d|z)] + (1-p) \log p_\pi(x_d) = \mathcal{L}_{alt}(p)$$

where

$$\mathop{\mathbb{E}}_{p(\boldsymbol{\xi})} [\log p(y|\mathbf{x}, \boldsymbol{\xi})] = \sum_{\xi_1} \dots \sum_{\xi_D} \prod_{d_1} p \prod_{d_0} (1-p) \Big(\log \sum_z p_B(x_{d_1}|z) p(z) p(y|z) - \log \sum_{z'} p_B(x_{d_1}|z') p(z') \Big)$$

Note that marginalizing over 2^D possible values for the switches $\boldsymbol{\xi}$ and over all K latent states is very computationally expensive.

C. Experimental Details

C.1. Data

We simulate data by varying several factors, aimed at replicating the real-world scenarios that one might use a mixture model in. We generate "relevant" and "irrelevant" dimensions separately and then concatenate them to form $\{\mathbf{x}_n, y_n\}_{n=1}^N$ pairs. For ease of exposition, assume that the first $D_{\textbf{rel}}$ dimensions are relevant. Relevant dimensions and targets are generated using the following generative process:

1. $z \sim \textbf{Cat}(z; [1/3, 2/3])$
2. $\mathbf{x}_d \sim (1-z)\mathcal{N}(0,1) + z\mathcal{N}(6,1)$ for all relevant dimensions $d \in \{1, \ldots, D_{\textbf{rel}}\}$
3. $y \sim (1-z)\mathcal{N}(0,1) + z\mathcal{N}(6,1)$

The irrelevant dimensions are generate using the following generative process:

1. $z' \sim \textbf{Cat}(z'; [1/3, 2/3])$
2. $\mathbf{x}_d \sim (1-z')\mathcal{N}(0,1) + z'\mathcal{N}(\mu_{\textbf{gap}}, 1)$ for all irrelevant dimensions $d \in \{D_{\textbf{rel}}+1, \ldots, D\}$

This gives us a few knobs to tweak:

1. D (number of input dimensions): we vary $D \in \{10, 100\}$. The $D = 10$ case is used as low-dimensional enough for all methods to be computable and hence comparable. The $D = 100$ case mimics a more real-world setting where only practical algorithms are compared.
2. $D_{\textbf{rel}}$(number of relevant dimensions): it is common for the input to only have a few relevant dimensions. Their number can vary a lot depending on the choice of target. Therefore, we vary $D_{\textbf{rel}}$in $\{$ 2, 5 $\}$ for $D = 10$ and in $\{$ 2, 10, 20, 40, 50 $\}$ for $D = 100$. We upper bound it by $D/2$ because we expect all methods to do well once the relevant signal is stronger than the irrelevant 'signal'.
3. $\mu_{\textbf{gap}}$ (distance between component means of irrelevant dimensions): the components again determine the level of misspecification for the model. $\mu_{\textbf{gap}} = 0$ corresponds to the case where there is no misspecification because the relevant dimensions are coming from a gaussian distribution. $\mu_{\textbf{gap}} = 6$ corresponds to the case where any irrelevant dimension is marginally distributed the same way as any relevant dimension. $\mu_{\textbf{gap}} = 8$ corresponds to the case where any irrelevant dimension marginally has more structure as compared to any single relevant dimension (and corresponds to highest misspecification).

C.2. Training Details

All methods are trained using the L-BFGS optimizer whenever possible and coordinate-ascent EM updates for the GMM implementation. All methods were trained several time starting at random initializations to avoid local minima, and the best performing models kept across initializations. For tuning hyperparameters (whenever relevant), model selection was performed on a separate validation set and the models were compared on heldout test sets.

D. Feature importance / Switch posteriors

We noted that the switch posteriors $\log p(\xi_d|\mathbf{x}, \mathbf{y})$ provide interpretable feature relevances. We show the inferred posterior probabilities for the ($D = 10$ and $\mu_{\textbf{gap}} = 6$) case in Fig. 3. For simplicity, the relevant dimensions are always plotted first. The left column shows the case when the misspecified dimensions would be preferred under the 2-Step or GMM approaches, while the right column has enough relevant dimensions that all models can prefer them. pfGMMvariants always give higher probability to relevant dimensions, except for the LLcase where one out of the two dimensions is considered relevant for first column and one irrelevant column is incorrectly considered relevant. We guess this happens because the hyperparameter grid for LL is much coarser than AltLBand ELBO. These posteriors end up being sufficient for getting good downstream predictions, which is the criteria for model selection.

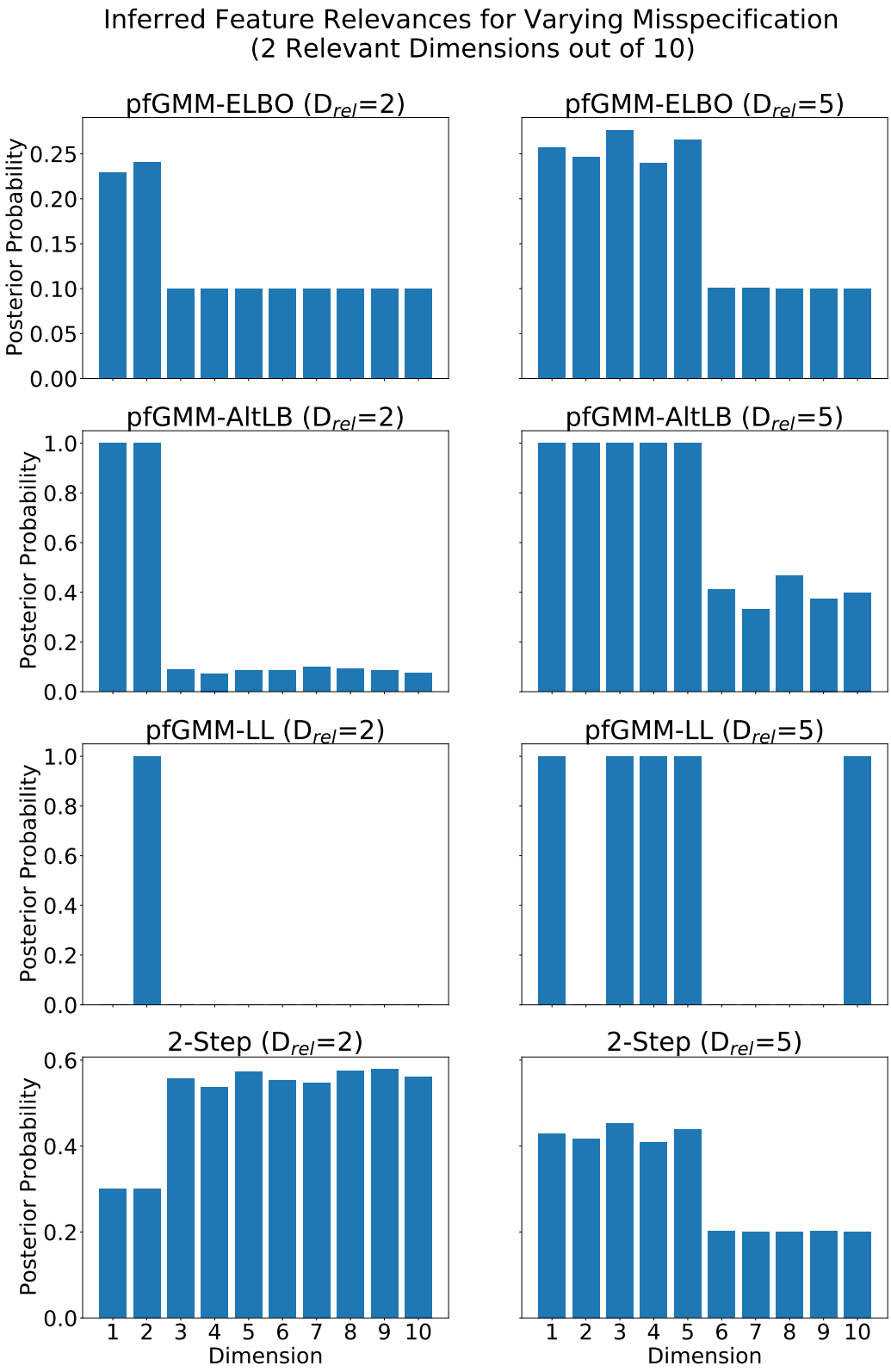


Figure 3. Inferred posterior when $D = 10$ and $\mu_{\text{gap}} = 6$. For simplicity, the relevant dimensions are always plotted first.