Variational Bayesian Unlearning through Shards

Yash Sinha 211195
Raj Agrawal 210809
Nikhil Mishra 210668
Himanshu Mittal 210441

Abstract

In our project we introduce a new method to speed up the process of removing sensitive data from machine learning models. We combine two techniques: Variational Bayesian Unlearning (VBU) and strategic data partitioning inspired by SISA training. VBU helps approximate the removal of data while keeping the model's performance intact. Strategic data partitioning minimizes the impact of individual data points, reducing the computational load. We test our approach on Bayesian models, showing its effectiveness in fast and safe data removal. This work contributes to better managing data privacy in machine learning.

1 Motivation for problem

Regulations like GDPR specify a "right to be forgotten" which requires machine learning model providers to enable mechanisms that allow for deletion of data and its contributions to the learning process. This can be achieved using Machine Unlearning models. Bayes' rule can be used to cast approximate unlearning of a small subset of the training data as an inference problem. However we face a challenge of not having access to the exact posterior of the model parameters. Existing unlearning methods, especially probabilistic ones, can be slow. We explore the method of dividing data into shards, and training multiple weak-learners to achieve faster unlearning, while maintaining accuracy.

2 Literature review

To address the problem of Machine unlearning we have gone through the following research works:

Variational Bayesian Unlearning frames unlearning as minimizing KL divergence between approximate posterior belief of model after unlearning the erased data, and the exact posterior obtained after retraining. The paper gives the following two propositions and shows that the minimization problem is equivalent to minimizing the Evidence Upper Bound (EUBO) as given below.

Proposition 1. $\mathrm{KL}[q_u(y|\mathcal{D}_r) \parallel p(y|\mathcal{D}_r)] \leq \mathrm{KL}[q_u(\theta|\mathcal{D}_r)]$

Proposition 2. EUBO, $U := \int q_u(\theta|D_r) \log p(\mathcal{D}_e|\theta)d\theta + \mathrm{KL}[q_u(\theta|\mathcal{D}_r) \parallel (\theta|\mathcal{D})]$

The EUBO can be seen as a trade-off between unlearning from D_e versus not forgetting the exact posterior belief $p(\theta|D)$. Often an approximate posterior belief $q(\theta|D)$, learnt using VI, has to be used instead of the exact posterior because of the latter's unavailability. But there can be inaccuracy in such an approximate posterior belief learnt using VI principles. $q(\theta|D)$ underestimates the variance of $p(\theta|D)$ as $q(\theta|D)$ can vary from $p(\theta|D)$ at places where $q(\theta|D)$ is very small. Also if stochastic optimization is used to learn the approximate posterior belief, samples with small approximate posterior belief may not have contributed to the maximisation. The paper introduces the following two tricks to address these inaccuracies. The first trick is a proposed adjusted likelihood approach to tackle this unlearning issue.

$$p_{\text{adj}}(\mathcal{D}_e|\theta;\lambda) = \begin{cases} p(\mathcal{D}_e|\theta) & \text{if } q(\theta|\mathcal{D}) > \lambda \max_{\theta'} q(\theta'|\mathcal{D}) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$\tilde{U}_{\text{adj}}(\lambda) = \int \tilde{q}_u(\theta|\mathcal{D}_r;\lambda) \log p_{\text{adj}}(\mathcal{D}_e|\theta;\lambda) d\theta + \text{KL}[\tilde{q}_u(\theta|\mathcal{D}_r;\lambda)||q(\theta|\mathcal{D})]$$

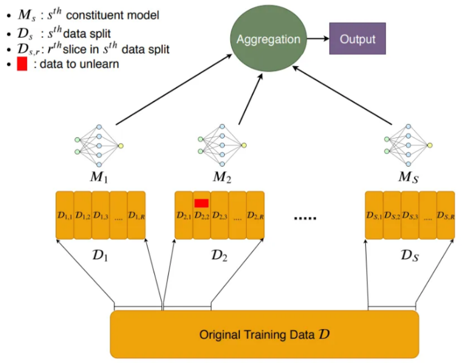
λ denotes the threshold for unlearning ($\lambda = 1$ means no unlearning and $\lambda = 0$ indicates full unlearning). However in practice when we visualize the effect on the unlearned EUBO distribution as λ tends to zero we see that $\tilde{q}_u(\theta|\mathcal{D}_r;\lambda)$ moves away from $q(\theta|\mathcal{D}_r)$

Reverse KL Method (rKL) The paper minimizes reverse KL divergence w.r.t. the approximate posterior belief recovered by directly unlearning from erased data $\mathcal{D}_e$.

$$\text{KL}[\tilde{p}(\theta|\mathcal{D}_r)||\tilde{q}_v(\theta|\mathcal{D}_r)] = C_0 - C_1 \mathbb{E}_{q(\theta|\mathcal{D})}[(\log \tilde{q}_v(\theta|\mathcal{D}_r))/p(\mathcal{D}_e|\theta)] \tag{2}$$

C_0 and C_1 are constants independent of $\tilde{q}_v(\theta|\mathcal{D}_r)$. In contrast to the optimized $\tilde{q}_u(\theta|\mathcal{D}_r;\lambda)$ from minimizing EUBO, the optimized $\tilde{q}_v(\theta|\mathcal{D}_r)$ from minimizing the reverse KL divergence overestimates variance of $\tilde{p}(\theta|\mathcal{D}_r)$.

Machine Unlearning introduces **SISA** (Sharded Isolated Sliced and Aggregated) training, where training data is divided into shards, which are further divided into slices. Models, called 'weak learners', are trained separately for each shard. All parameters of a shards are saved before a new slice is added to it. When a point needs to be unlearned, its corresponding shard, and slice is located. The slice obtained, and all slices in the found shard after the obtained slice, have to be retrained to account for the unlearning of the training point in the shard. The predictions of all weak-learners are aggregated to give final predictions.



M_s : s^{th} constituent model
$\mathcal{D}_s$: s^{th} data split
$\mathcal{D}_{s,r}$: r^{th} slice in s^{th} data split
🟥 : data to unlearn
Aggregation
Output
M_1
M_2
M_S
$\mathcal{D}_{1,1}$ $\mathcal{D}_{1,2}$ $\mathcal{D}_{1,3}$ .... $\mathcal{D}_{1,R}$
$\mathcal{D}_{2,1}$ $\mathcal{D}_{2,2}$ $\mathcal{D}_{2,3}$ .... $\mathcal{D}_{2,R}$
$\mathcal{D}_{S,1}$ $\mathcal{D}_{S,2}$ $\mathcal{D}_{S,3}$ .... $\mathcal{D}_{S,R}$
$\mathcal{D}_1$
$\mathcal{D}_2$
$\mathcal{D}_S$
Original Training Data $\mathcal{D}$

SISA training

Challenges and Pitfalls of Bayesian Unlearning evaluates the effectiveness of Laplace Bayesian Unlearning and Variational Bayesian Unlearning methods and discusses the drawbacks of both approaches.

For the Laplace approximation for the unlearning case, given an approximate posterior $q(\theta|\mathcal{D}_{\text{all}})$, and likelihood of deleted data, $p(\mathcal{D}_{\text{del}}|\theta)$, the ratio of these is defined as $\hat{q}(\theta)$, where

$$\log \hat{q}(\theta) = -\log p(\mathcal{D}_{\text{del}}|\theta) + \log q(\theta|\mathcal{D}_{\text{all}}) + C. \tag{3}$$

The first term $-\log p(\mathcal{D}_{\text{del}}|\theta)$, encourages "forgetting" on the deleted dataset, and the second term $\log q(\theta|\mathcal{D}_{\text{all}})$, retains closeness to previous posterior.

Let $\theta_{\text{L-BUN}} = \arg\max_\theta(\log \hat{q}_\theta)$.
If the initial approximate posterior distribution was a Gaussian, $q(\theta|\mathcal{D}_{\text{all}}) = \mathcal{N}(\theta|\mu_{\text{all}}, \Sigma_{\text{all}}),$ then $\hat{q}(\theta)$ can be approximated as a Gaussian with mean $\mu_{\text{L-BUN}} = \theta_{\text{L-BUN}}$ and precision $\Sigma^{-1}_{\text{L-BUN}} = \Sigma^{-1}_{\text{all}} + \nabla^2_\theta \log p(\mathcal{D}_{\text{del}}|\theta)|_{\theta_{\text{L-BUN}}}$.

Deep Ensembles from a Bayesian Perspective discusses deep ensemble methods to address the epistemic uncertainty in approximating Bayesian Models.

Approximate posterior predictive distribution when a collection of ensembles is used is given by

$$\pi(y|x, D) = \int \pi(\theta|D)\mathcal{N}(y; \eta_\theta(x), \sigma^2_\theta(x)\mathrm{I})d\theta \approx \frac{1}{L}\sum_{i=1}^{L} \mathcal{N}(y; \eta_{\hat{\theta}^{(l)}}(x), \sigma^2_{\hat{\theta}^{(l)}}(x)\mathrm{I}) \tag{4}$$

where a normal distributions are fitted around local MAP points obtained. Since would-be MAP estimates now have associated variances, this method provides much better uncertainty estimates.

Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks

Find a 'scrubbing' function $S(w)$, such that you can find another scrubbing function $S_0(w)$, independent of the data you want to remove, such that a readout function f can't distinguish between distributions of original scrubbed weights on complete data, and scrubbed weights on retained data. Modelling the information an attacker can access by the KL-divergence, we want that

$$\text{KL}(P(f(S(w))|\mathcal{D})\|P(f(S_0(w))|\mathcal{D}_r)) = 0 \tag{5}$$

To find a scrubbing function which forgets data while retaining model accuracy, the 'Forgetting Lagrangian' is defined as

$$\mathcal{L} = \mathbb{E}_{S(w)}[\mathcal{L}_{D_r}(w)] + \lambda\text{KL}(P(S(w)|D)\|P(S_0(w)|\mathcal{D}_r)) \tag{6}$$

where the first term seeks to minimize loss of model on retained data while the second term decreases obtainable information about the erased data.

The loss function is minimized by using the following hessian approximation:

$$\nabla^2\mathcal{L}_{\mathcal{D}}(w) \approx \mathbb{E}_{x\sim\mathcal{D}, y\sim p(y|x)}[\nabla_w \log p_w(y|x)\nabla_w \log p_w(y|x)^T] \tag{7}$$

3 Novelty of work compared to previous work

Existing unlearning methods, especially probabilistic ones, can be slow. We explore ensembles to achieve faster unlearning while maintaining accuracy.

We integrate the two approaches of Variational Bayesian Unlearning and SISA, aiming to improve VBU with the efficiency of SISA, while improving variance estimates of conventional SISA through VBU.

Dividing the model into shards is also expected to provide especially higher efficiency while sequentially unlearning data.

4 Experimental method

4.1 Description of data

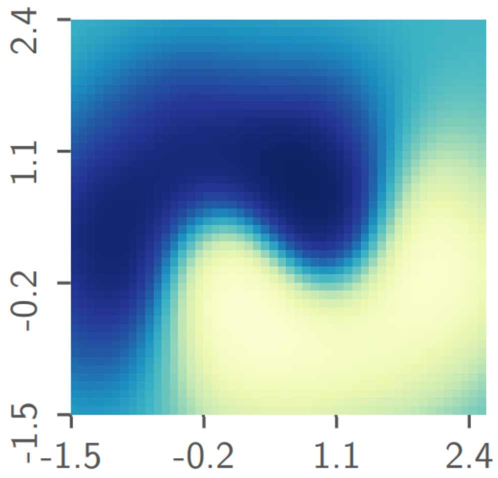
We employed a Moon classification dataset generated via the scikit-learn library. The dataset comprises of synthetic two-dimensional data points organized into moon-shaped clusters, with associated class labels. Configured with a total of 100 data points, the dataset was partitioned into retained and removed sets, with 80 data points retained, and 20 removed. Gaussian noise with a standard deviation of 0.2 was introduced during the data generation process.

In this dataset, The likelihood of input $x \in \mathbb{R}^2$ being classified into the 'blue' category (indicated by $y_x = 1$ and represented by blue dots in the figure shown below) is described as $1/(1 + \exp(f_x))$, where f_x is a latent function modeled using a sparse Gaussian Process. The GP is characterized by 20 inducing variables, and its approximate posterior beliefs are modeled as multivariate Gaussians with full covariance matrices prior of which can be defined by the widely-used squared exponential covariance function:

$$\sigma_f^2 k_{\mathbf{x}\mathbf{x}'} = \exp(-0.5||\mathbf{\Lambda}(x - \mathbf{x}')||^2), \quad \mathbf{\Lambda} = \text{diag}[\lambda_1, \lambda_2],$$
$$\lambda_1 = 1.56, \lambda_2 = 1.35, \sigma^2 = 4.74$$

The posterior belief of the latent function value $f_{\mathbf{x}}$ at a new input $\mathbf{x}$ is the Gaussian:

$$p(f_{\mathbf{x}}|f_{\mathcal{X}_u}) = \mathcal{N}(\mathbf{k}_{\mathbf{x}\mathcal{X}_u}\mathbf{K}^{-1}_{\mathcal{X}_u\mathcal{X}_u}\mathbf{f}_{\mathcal{X}_u}, \mathbf{k}_{\mathbf{x}\mathbf{x}} - \mathbf{k}_{\mathbf{x}\mathcal{X}_u}\mathbf{K}^{-1}_{\mathcal{X}_u\mathcal{X}_u}\mathbf{k}_{\mathcal{X}_u}\mathbf{x})$$



Visualization of synthetic moon dataset

4.2 Shards

We partition the dataset $\mathcal{D}_e$ into smaller, non-overlapping subsets known as shards, and subsequently apply unlearning techniques such as EUBO and rKL to each shard. This process yields posterior distributions for each unlearned shard. Through Distributed Learning, we combine these distributions and compute the KL divergence between this aggregated distribution and the distribution obtained by training the model from scratch with the remaining data. We analyze the outcomes of these techniques and explore the impact of varying the number of shards on predictions.

The following is the mathematical formulation of the approach we took for aggregating posteriors from different shards using a decentralized MCMC-like method.

Let S_i represent shard i, and let f_i represent the posterior distribution of parameters obtained from shard S_i. Each shard performs MCMC sampling independently, resulting in samples $\{ f_{i,j}\}_{j=1}^{N_i}$, where N_i is the number of samples obtained from shard S_i.

At each iteration t, shards exchange their samples with each other. Let $\mathcal{N}(i)$ denote the neighbors of shard i, and let $\{ f_{j,k}^{(t)}\}_{k=1}^{M_{i,j}^{(t)}}$ represent the samples received by shard i from its neighbor shard j at iteration t, where $M_{i,j}^{(t)}$ is the number of samples received.

The aggregation process involves updating the posterior distribution f_i of each shard based on the received samples. This can be done using a Bayesian update rule such as:

$$f_i^{(t+1)} \propto \prod_{j \in \mathcal{N}(i)} \left(\frac{\pi(f_i^{(t)})}{\pi(f_{j,k}^{(t)})} \right)^{w_{i,j}} \times \prod_{k=1}^{N_i} \pi(f_{i,k})$$

where: $\pi(f_i^{(t)})$ is the posterior distribution of shard i at iteration t.

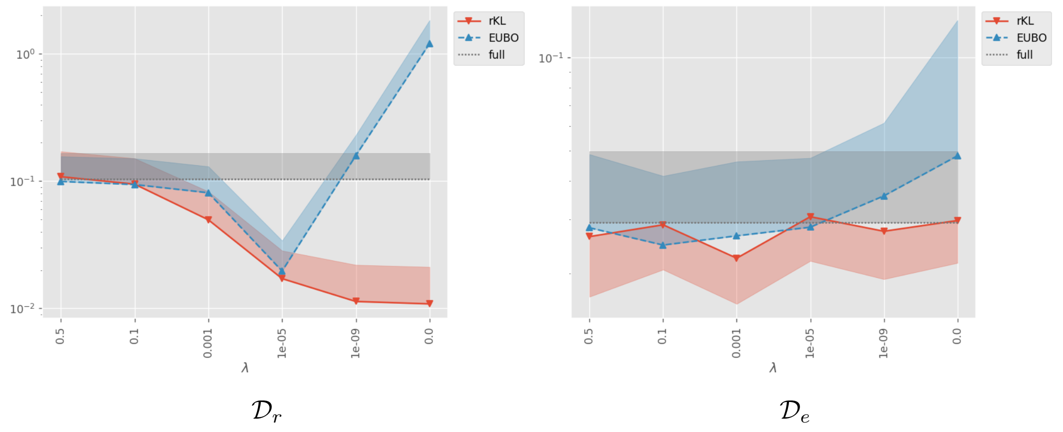
$\pi(\ f_{j,k}^{(t)})$ is the posterior distribution of shard j for the k-th received sample at iteration t.

$w_{i,j}$ is a weight representing the influence of shard j on shard i, which can be based on factors such as the similarity of data distributions or the reliability of shard j's samples. In our term we didn't really use it because the shards were chosen randomly and it was assumed that all the shards hold equal importance. This might be a possible reason in why our results don't have the accuracy which could have been possible if we had taken these weights into account.

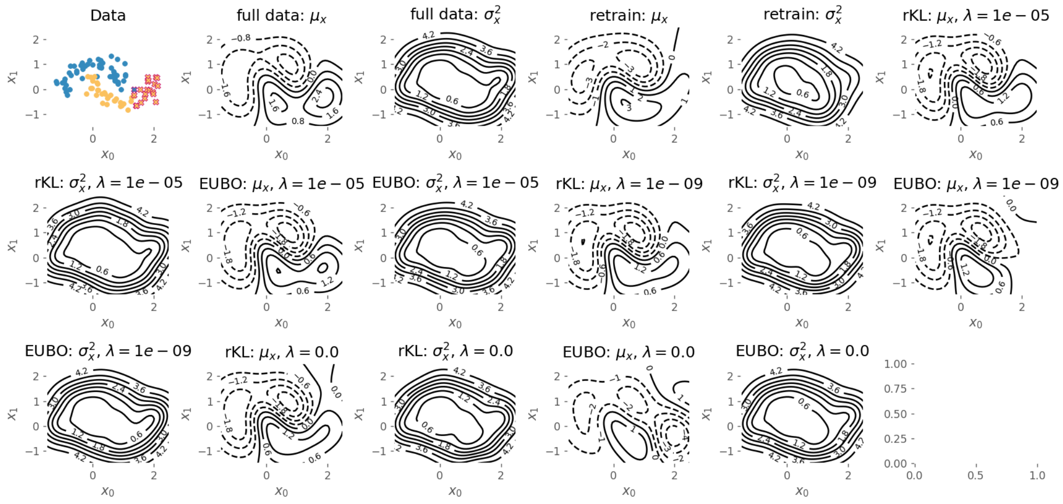
The product term $\prod_{k=1}^{N_i} \pi(\ f_{i,k})$ represents the contribution of shard i's own samples to the updated posterior.

The process iterates until convergence is reached, resulting in a consensus posterior distribution across all shards.

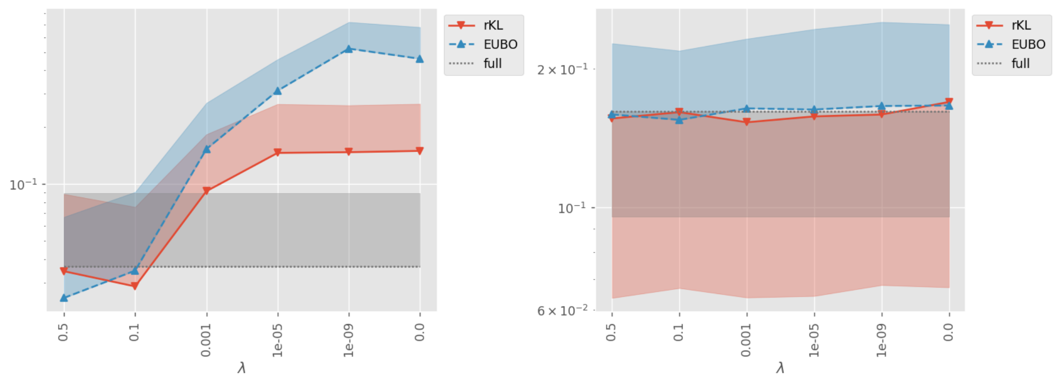
5 Results



rKL and EUBO variations when trained on 1 shard, i.e. same as the original model

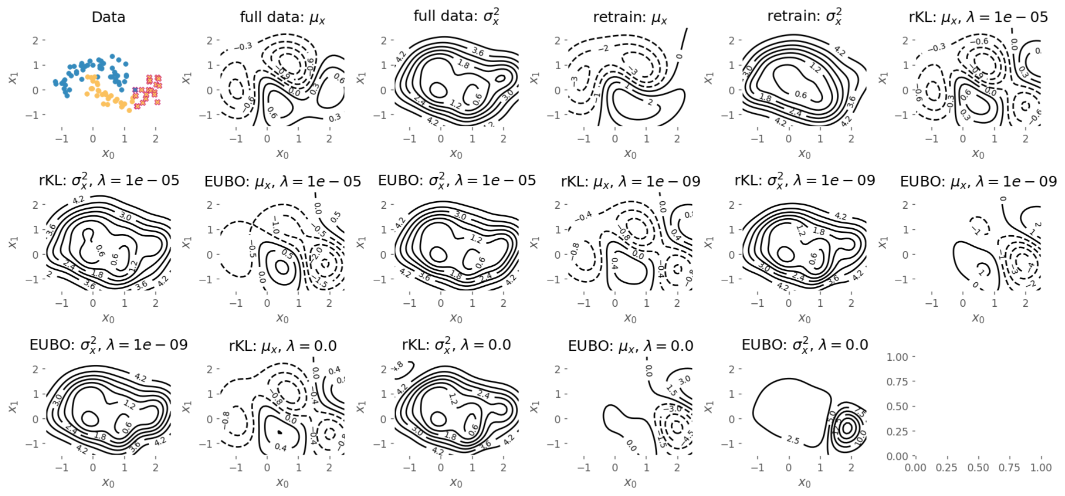


Approximate posterior beliefs, 1 shard

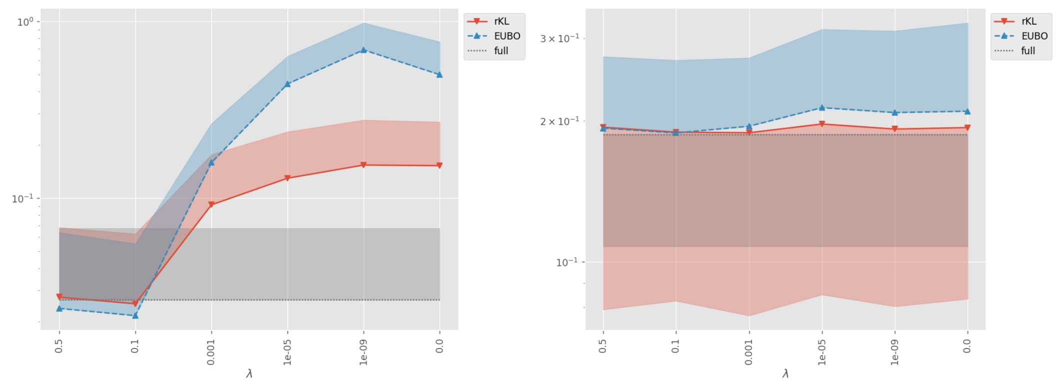


$\mathcal{D}_r$ $\mathcal{D}_e$

rKL and EUBO variations when trained on 3 shards

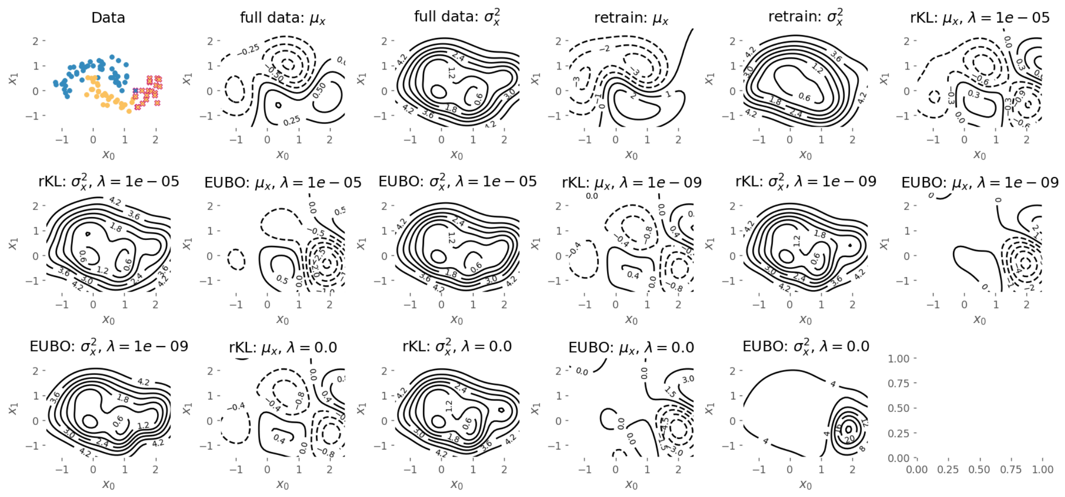


Approximate posterior beliefs, 3 shards



$\mathcal{D}_r$ $\mathcal{D}_e$

rKL and EUBO variations when trained on 5 shards



Approximate posterior beliefs, 5 shards

The graphs above depict the variation of KL divergence between the unlearned model and the retrained model, with varying levels of unlearning, λ.

Approximate posterior beliefs when using shards appear distorted and divided as expected, since sharding was done index-wise, with data points belonging to a shard lying close to each other.

When sharding the data, there is a noticeable increase in variance of rKL and EUBO on erased data $\mathcal{D}_e$, which increases with the number of shards. However, there is not much corresponding increase in the variance of these functions on retained data $\mathcal{D}_r$, which is desired, and indicates improvement over the original, un-sharded model.

Effectiveness of both the rKL and EUBO methods hinges on the λ parameter, which controls the degree of unlearning. For larger values of λ, the group of weak-learners approximate the required posterior quite well. However, the approximation becomes too crude for smaller values of λ, resulting in an increase in rKL and EUBO.

The saturation of rKL values on increasing λ implies the presence of a "base" divergence between the models, which is actually the divergence when no data is unlearned, and is due to the approximation in training multiple weak-learners.

The implementation can be found **here**.

6 Possible future work

Possible future work could include exploring dynamic shard allocation based on data importance or sensitivity. This could involve assigning stronger "forgetting" mechanisms to more sensitive data shards. Additionally, techniques for sequential unlearning could be developed, allowing the model to forget information iteratively as it encounters new data and evolves. Finally, investigating unlearning methods beyond VBU could be beneficial, potentially leading to better compatibility with a sharded architecture.

7 References

[1] Nguyen, Quoc Phong et al. “Variational Bayesian Unlearning.” ArXiv abs/2010.12883 (2020).

[2] Bourtoule, Lucas et al. “Machine Unlearning.” 2021 IEEE Symposium on Security and Privacy (SP) (2019): 141-159.

[3] Rawat, Ambrish et al. “Challenges and Pitfalls of Bayesian Unlearning.” ArXiv abs/2207.03227 (2022)

[4] Hoffmann, Lara and Clemens Elster. “Deep Ensembles from a Bayesian Perspective.” ArXiv abs/2105.13283 (2021).

[5] Golatkar, A., Achille, A., and Soatto, S. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9304–9312, 2020.