Machine Unlearning with Variational Inference

Ali Faraz (17807078) **Jayant Ranwka (170323)**

Shashank Kumar (170642) **Vishweshwar Tyagi (191173)**

{alifaraz, jayant, shankk, tyagi}@iitk.ac.in

Indian Institute of Technology Kanpur

1 Problem Overview

In this project we discuss the problem of *Machine Unlearning*, wherein the training dataset of our machine-learning model, denoted by $\mathcal{D}$, is partitioned into $\mathcal{D}_e$ and $\mathcal{D}_r$, i.e, $\mathcal{D} = \mathcal{D}_e \cup \mathcal{D}_r$ with $\mathcal{D}_e \cap \mathcal{D}_r = \phi$. Here, $\mathcal{D}_e$ denotes the *erased dataset*, which is to be forgotten by our model, and $\mathcal{D}_r$ denotes the remaining dataset. The task is then to have our model unlearn from $\mathcal{D}_e$ without having to retrain it on $\mathcal{D}_r$.

2 Motivation

Our motivation to study this challenging topic mainly comes from

- the fact that users are entitled to their right of privacy and may wish to exercise their *right to be forgotten*, wherein a user upon exiting an ML application, may wish that their user-data is not retained by the application.
- the persistent threat to data, wherein small chunks of large data may get corrupted and retraining on the remaining portion would prove to be too cost-inefficient.

In our study, we ensure that our model, upon unlearning, doesn't suffer from *catastrophic unlearning*, which takes place when the unlearned model performs significantly poorer than the one retrained on remaining dataset.

3 Literature Review

This problem was first addressed by [1] where they converted a learning algorithm to a statistical query learning form, which consists of a small number of summations. The learning algorithm only depends on these summations, which are the sum of some efficiently computable transformation of the training data samples. To unlearn, one just needs to subtract the transformations of that sample from all the summations. [3] proposes a Bayesian approach with loss function equal to the KL divergence between the approximate posterior by directly unlearning from the erased data and the exact posterior obtained by retraining with the remaining data.

3.1 Exact Bayesian Unlearning

We want the exact posterior $p(\boldsymbol{\theta} \mid \mathcal{D}_r)$. Using Bayes' theorem and assuming conditional independence between $\mathcal{D}_r$ and $\mathcal{D}_e$ given $\boldsymbol{\theta}$, we can show that

$$p(\boldsymbol{\theta} \mid \mathcal{D}_r) = \frac{p(\boldsymbol{\theta} \mid \mathcal{D})\, p(\mathcal{D}_e \mid \mathcal{D}_r)}{p(\mathcal{D}_e \mid \boldsymbol{\theta})} \propto \frac{p(\boldsymbol{\theta} \mid \mathcal{D})}{p(\mathcal{D}_e \mid \boldsymbol{\theta})}. \tag{1}$$

Thus, we are able to write the posterior given the remaining data up to a proportionality constant in terms of the complete data posterior and likelihood of the erased data. Equation 1 can be used only when the model parameters are discrete-valued or when we use a conjugate prior. Otherwise, we will have to resort to some approximation.

3.2 Evidence Upper Bound (EUBO)

[3] initially considers the loss function as the KL divergence between the approximate and the exact PPDs.

$$KL\left[q_u(y \mid \mathcal{D}_r) \,\|p(y \mid \mathcal{D}_r)\right]$$

Evaluating these PPDs in closed form will not always be possible. However, [3] shows that

$$KL\left[q_u(y \mid \mathcal{D}_r) \,\|p(y \mid \mathcal{D}_r)\right] \leq KL\left[q_u(\boldsymbol{\theta} \mid \mathcal{D}_r) \,\|p(\boldsymbol{\theta} \mid \mathcal{D}_r)\right]$$

motivating the use of $KL\left[q_u(\boldsymbol{\theta} \mid \mathcal{D}_r) \,\|p(\boldsymbol{\theta} \mid \mathcal{D}_r)\right]$ as the loss function.
Now, EUBO is defined as

$$\mathcal{U} \triangleq \int q_u(\boldsymbol{\theta} \mid \mathcal{D}_r) \log p(\mathcal{D}_e \mid \boldsymbol{\theta})\; d\boldsymbol{\theta} + KL\left[q_u(\boldsymbol{\theta} \mid \mathcal{D}_r) \,\|p(\boldsymbol{\theta} \mid \mathcal{D})\right].$$

The first term ensures that the likelihood of the erased data is low while the second term ensures that the new posterior is close to the original posterior to prevent *catastrophic unlearning*. Minimising $KL\left[q_u(\boldsymbol{\theta} \mid \mathcal{D}_r) \,\|p(\boldsymbol{\theta} \mid \mathcal{D}_r)\right]$ is equivalent to minimising EUBO since it can be simplified to $\mathcal{U} = \log p(\mathcal{D}_e \mid \mathcal{D}_r) + KL\left[q_u(\boldsymbol{\theta} \mid \mathcal{D}_r) \,\|p(\boldsymbol{\theta} \mid \mathcal{D}_r)\right]$.

3.2.1 Adjusted Likelihood

When the posterior over the complete data set is also approximate, we represent it using $q(\boldsymbol{\theta} \mid \mathcal{D})$ and put $\tilde{}$ over the other quantities.
When $q(\boldsymbol{\theta} \mid \mathcal{D})$ is learned using VI by minimising $KL\left[q(\boldsymbol{\theta} \mid \mathcal{D}) \,\|p(\boldsymbol{\theta} \mid \mathcal{D})\right]$, there can be two sources of inaccuracy [3].

1. Variance of $p(\boldsymbol{\theta} \mid \mathcal{D})$ is usually underestimated as when $q(\boldsymbol{\theta} \mid \mathcal{D})$ is close to 0, $p(\boldsymbol{\theta} \mid \mathcal{D})$ may not be close to 0
2. If stochastic optimisation is used for maximising the ELBO, it is unlikely that samples with small $q(\boldsymbol{\theta} \mid \mathcal{D})$ are used

To curb unlearning at values of $\boldsymbol{\theta}$ with small $q(\boldsymbol{\theta} \mid \mathcal{D})$ we use an adjusted likelihood of the erased data [3].

$$p_{adj}(\mathcal{D}_e \mid \boldsymbol{\theta}; \lambda) \triangleq \begin{cases} p(\mathcal{D}_e \mid \boldsymbol{\theta}) & \text{if } q(\boldsymbol{\theta} \mid \mathcal{D}) > \lambda \max_{\boldsymbol{\theta}'} q(\boldsymbol{\theta}' \mid \mathcal{D}), \\ 1 & \text{otherwise} \end{cases}$$

Here, $\lambda \in [0, 1]$. Using equation 1,

$$\tilde{p}_{adj}(\boldsymbol{\theta} \mid \mathcal{D}_r; \lambda) \propto \begin{cases} \frac{q(\boldsymbol{\theta}|\mathcal{D})}{p(\mathcal{D}_e|\boldsymbol{\theta})} & \text{if } q(\boldsymbol{\theta} \mid \mathcal{D}) > \lambda \max_{\boldsymbol{\theta}'} q(\boldsymbol{\theta}' \mid \mathcal{D}), \\ q(\boldsymbol{\theta} \mid \mathcal{D}) & \text{otherwise} \end{cases} \tag{2}$$

3.3 Effect of hyperparameter λ

We will now discuss the effect that λ has on unlearning process when using EUBO to minimize $KL\left[\tilde{q}_u\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right) \| \tilde{p}_{adj}\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)\right]$

- For $\lambda \simeq 0$, we have from equation (2)

$$\tilde{p}_{adj}\left(\boldsymbol{\theta} \mid \mathcal{D}_r; \lambda\right) \propto \frac{q(\boldsymbol{\theta} \mid \mathcal{D})}{p(\mathcal{D}_e \mid \boldsymbol{\theta})} \quad \forall \boldsymbol{\theta}$$

  and thus, $\tilde{p}\left(\boldsymbol{\theta} \mid \mathcal{D}_r; \lambda\right) = \tilde{p}\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)$, i.e, unlearning takes place without *adjusted* likelihood function and may result in *catastrophic unlearning*.

- As λ increases, the $\boldsymbol{\theta}-$space where unlearning takes place, i.e, $\{\boldsymbol{\theta} \mid q\left(\boldsymbol{\theta} \mid \mathcal{D}\right) > \lambda \max_{\boldsymbol{\theta}'} q\left(\boldsymbol{\theta}' \mid \mathcal{D}\right)\}$ reduces which ensures that unlearning happens only in a constrained space with sufficiently large $q(\boldsymbol{\theta} \mid \mathcal{D})$

- For $\lambda \simeq 1$, we have $\{\boldsymbol{\theta} \mid q\left(\boldsymbol{\theta} \mid \mathcal{D}\right) > \lambda \max_{\boldsymbol{\theta}'} q\left(\boldsymbol{\theta}' \mid \mathcal{D}\right)\} = \phi$ which results in

$$\tilde{p}_{adj}\left(\boldsymbol{\theta} \mid \mathcal{D}_r; \lambda\right) \propto q(\boldsymbol{\theta} \mid \mathcal{D}) \quad \forall \boldsymbol{\theta}$$

  and essentially no unlearning takes place.

Hence, choosing appropriate $\lambda \in (0, 1)$ becomes essential and in-order to do so we can either tune λ on a validation set or make use of *Reverse KL* method [3] which seems to offer robustness against the effect of λ.

3.4 Reverse KL

While minimizing $KL\left[\tilde{q}_u\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right) \| \tilde{p}\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)\right]$ with respect to $\tilde{q}_u\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)$, our recovered posterior upon unlearning used to underestimate the variance of $\tilde{p}\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)$

Now, in Reverse KL method, instead of minimizing $KL\left[\tilde{q}_u\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right) \| \tilde{p}\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)\right]$, we minimize $KL\left[\tilde{p}\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right) \| \tilde{q}_v\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)\right]$ with respect to $\tilde{q}_v\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)$ which results in recovered posterior upon unlearning $\tilde{q}_v\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)$ overestimating the variance, and is more desirable since source of inaccuracies lie in region with our recovered posterior upon unlearning close to 0.

Note that here both $\tilde{q}_u\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)$ and $\tilde{q}_v\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)$ are recovered posteriors upon unlearning and we use different subscripts only to differentiate between the methods used to obtain them.

This method can then be combined with the *adjusted* likelihood trick to minimize $KL\left[\tilde{p}_{adj}\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right) \| \tilde{q}_v\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)\right]$, or rather maximize $\mathbb{E}_{q(\boldsymbol{\theta}|\mathcal{D})}\left[\ln \tilde{q}_v\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right) / \tilde{p}_{adj}\left(\mathcal{D}_e \mid \boldsymbol{\theta}\right)\right]$ which follows from

$$KL\left[\tilde{p}_{adj}\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right) \| \tilde{q}_v\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)\right] = C_0 - C_1 \mathbb{E}_{q(\boldsymbol{\theta}|\mathcal{D})}\left[\frac{\ln \tilde{q}_v\left(\boldsymbol{\theta} \mid \mathcal{D}_r\right)}{\tilde{p}_{adj}\left(\mathcal{D}_e \mid \boldsymbol{\theta}\right)}\right]$$

where C_0 and C_1 are constants.

4 Experiment

In order to verify the robustness of Reverse KL method against the effect of varying λ, we implemented a *Sparse Gaussian Process classification* model on 2-D *Synthetic Moon* dataset with binary response which can be easily generated with sklearn.datasets.make_moons.

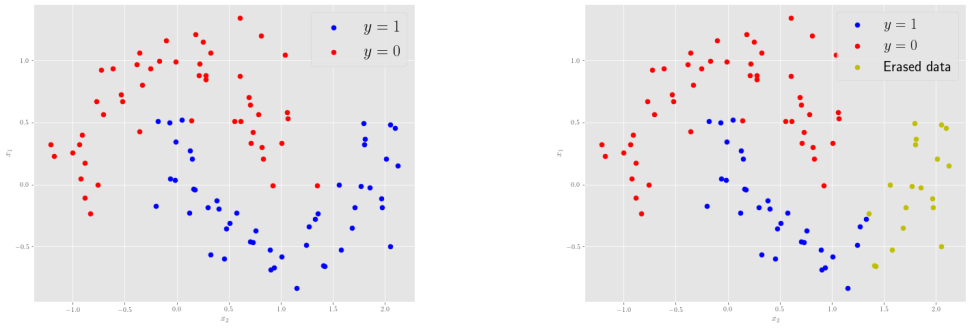
4.1 Dataset

Denote our dataset by $\mathcal{D}$ and size of dataset by N, we have $N = 100$ with

$\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$ where $y_n \in \{0, 1\}$ and $\boldsymbol{x}_n = (x_n^1, x_n^2)$ with $x_n^i \in \mathbb{R}\ \forall i \in \{0, 1\}$ and $n \in \{1, \ldots N\}$.

4.2 Training and Unlearning

We partitioned $\mathcal{D}$ into $\mathcal{D}_e$ and $\mathcal{D}_r$ with $\mid \mathcal{D}_e \mid = 20$ and $\mid \mathcal{D}_r \mid = 80$ and visualize the same below



(a) Synthetic Moon dataset

(b) Remaining and Erased data

Figure 1: Data Visualization

With a batch-size of 100 (Batch Stochastic-Gradient Descent) and number of iterations kept to $30,000$, we optimized our model parameters with tensorflow's [**?**] *Adam* optimizer and compared the effect of λ on *averaged* KL using EUBO and Reverse KL methods.

4.3 Results

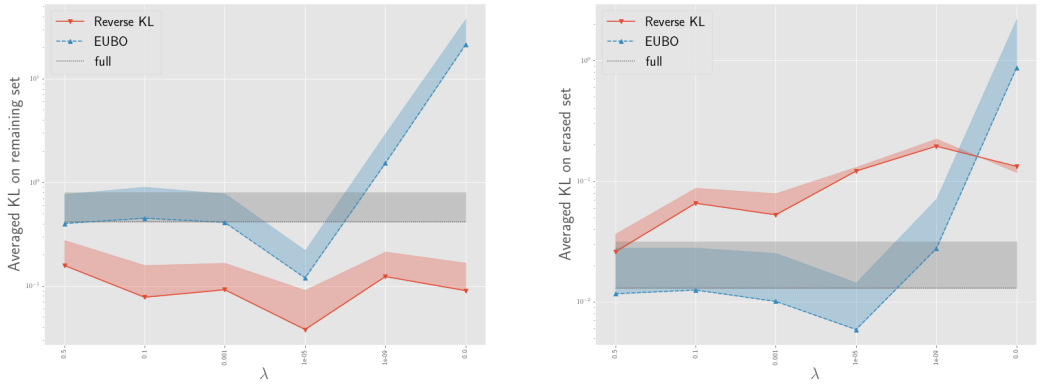


Figure 2: Effect of λ on EUBO and Reverse KL methods

From here, we can see that

- KL divergence minimized via EUBO overshoots as λ decreases.
- KL divergence minimized via Reverse KL stays relatively low even as λ decreases

Hence, Reverse KL seems to be more robust against the effect of varying λ and this experiment successfully demonstrates our theory.

5 BIF Framework

In this section, we look beyond the framework of variational inference which only applies to parametric models. We will look into the ideas the make up any Bayesian inference forgetting algorithm [2]. We first look into the definition of ϵ-certified knowledge removal[2] which quantifies the performance of knowledge removal in these algorithms. Next, we study energy functions and then we see how forgetting algorithms are designed based on these energy functions [2]. Lastly, we will see how a theoretical error bound is established on these algorithms [2].

5.1 ϵ-Certified Knowledge Removal

Let S be the original data and $S' \subset S$ be the removed data. Then, for $\epsilon > 0$, we say that algorithm $\mathcal{A}$ performs ϵ-certified knowledge removal if,

$$KL(\hat{p}_S^{-S'}, \hat{p}_{S-S'}) \leq \epsilon$$

Here, $\hat{p}_S^{-S'} = \mathcal{A}(\hat{p}_S, S')$ and $\mathcal{A}$ is a forgetting algorithm designed to process the distribution $\hat{p}_S$ in order to find an estimate for $\hat{p}_{S-S'}$. Here, $\hat{p}_S$ is the probabilistic model on the original data S and $\hat{p}_{S-S'}$ is that which would be obtained by re-training the model on the remaining data $S - S'$.

5.2 Energy Functions in Bayesian Inference

Analogous to the free-energy principle in physics, we can formulate Bayesian inference as minimizing an energy function. Let $F(\gamma, S)$ be the energy function of a probabilistic distribution $\chi(\gamma)$ parametrized by $\gamma \in \Gamma \subset \mathbb{R}^K$ over the model parameter space Θ. Typically, $F(\gamma, S)$ has the following form,

$$F(\gamma, S) = \sum_{i=1}^{n} h(\gamma, z_i) + f(\gamma)$$

Here, $h(\gamma, z)$ is a function defined on $\Gamma \text{x} \mathcal{Z}$ that characterizes the influence from individual datums, and $f(\gamma)$ (defined on Γ) characterizes the influence from the prior of γ.
Similarly, the energy function for $S - S'$ is $F(\gamma, S - S')$. Clearly,

$$F(\gamma, S - S') = F(\gamma, S) - \sum_{z \in S'} h(\gamma, z)$$

5.3 General Structure of a Forgetting Algorithm

Let us the start with the simplest case of removing the influence learned from a single datum $z_j \in S$. As we saw previously, the energy function for the posterior distribution $p(\theta|S - \{z_j\})$ is

$$F\left(\gamma, S - \{z_j\}\right) = \sum_{i=1}^{n} h\left(\gamma, z_i\right) + f\left(\gamma\right) - h\left(\gamma, z_j\right)$$

Since the probabilistic model $\hat{p}_S$ is parametrized by γ_S, γ_S is a local minimizer of the energy function $F(\gamma, S)$. For similar reasons, $\gamma_{S-\{z_j\}}$ is a local minimizer of the energy function $F(\gamma, S - \{z_j\})$. Then, the value of the gradients of the energy functions at these points of minima will be zero.

$$\nabla_\gamma F\left(\gamma_S, S\right) = \sum_{i=1}^{n} \nabla_\gamma h\left(\gamma_S, z_i\right) + \nabla_\gamma f\left(\gamma_S\right) = 0$$

$$\nabla_\gamma F\left(\gamma_{S-\{z_j\}}, S - \{z_j\}\right) = \nabla_\gamma F\left(\gamma_{S-\{z_j\}}, S\right) - \nabla_\gamma h\left(\gamma_S - \{z_j\}, z_j\right) = 0$$

We can clearly see that the above two equations are special cases of the following equation,

$$\nabla_\gamma F\left(\gamma, S\right) + \tau \nabla_\gamma h\left(\gamma, z_j\right) = 0$$

Here, $\tau \in [-1, 0]$, and for every value of τ, $\hat{\gamma}_S^{-z_j}(\tau)$ is a solution of the above equation (an implicit mapping is induced on $\hat{\gamma}_S^{-z_j}$). Therefore, $\hat{\gamma}_S^{-z_j}(\tau)$ is a critical point of the function

$$F_{-z_j,\tau}(\gamma, S) = F(\gamma, S) + \tau h(\gamma, z_j).$$

$\gamma_{S-\{z_j\}}$ can therefore be approximated using first-order approximation in the following manner,

$$\gamma_{S-\{z_j\}} = \hat{\gamma}_S^{-z_j}(-1) \approx \hat{\gamma}_S^{-z_j}(0) - \frac{\partial \hat{\gamma}_S^{-z_j}(0)}{\partial \tau} = \gamma_S - \frac{\partial \hat{\gamma}_S^{-z_j}(0)}{\partial \tau}$$

Therefore, the forgetting algorithm will process the request of removing the datum z_j by replacing γ_S with $\gamma_S^{-z_j}$ where,

$$\gamma_S^{-z_j} := \gamma_S - \frac{\partial \hat{\gamma}_S^{-z_j}(0)}{\partial \tau}.$$

5.4 Theoretical Guarantee

In the BIF paper [2], the authors have proved the following three things under a few mild assumptions.

1. The mapping $\hat{\gamma}_S^{-z_j}$ uniquely exists.
2. $\gamma_{S-z_j} = \hat{\gamma}_S(-1)$ is the global minimizer of $F(\gamma, S - \{z_j\})$
3. The approximation error between $\gamma_S^{-z_j}$ and $\gamma_{S-\{z_j\}}$ is not larger than order $O(1/n^2)$, where n is the size of the training set.

Thus we see that, provably certified forgetting algorithms for Bayesian inference methods can be developed under the BIF framework using the above ideas. In the BIF paper, the authors have clearly derived the forgetting algorithms for Variational Inference and MCMC methods, but we do not include them in our report for brevity.

6 Experiment

In order to evaluate the methods given in the BIF paper, we conducted an experiment where we applied SGHMC (an MCMC method) to GMM on synthetic data.

6.1 Dataset

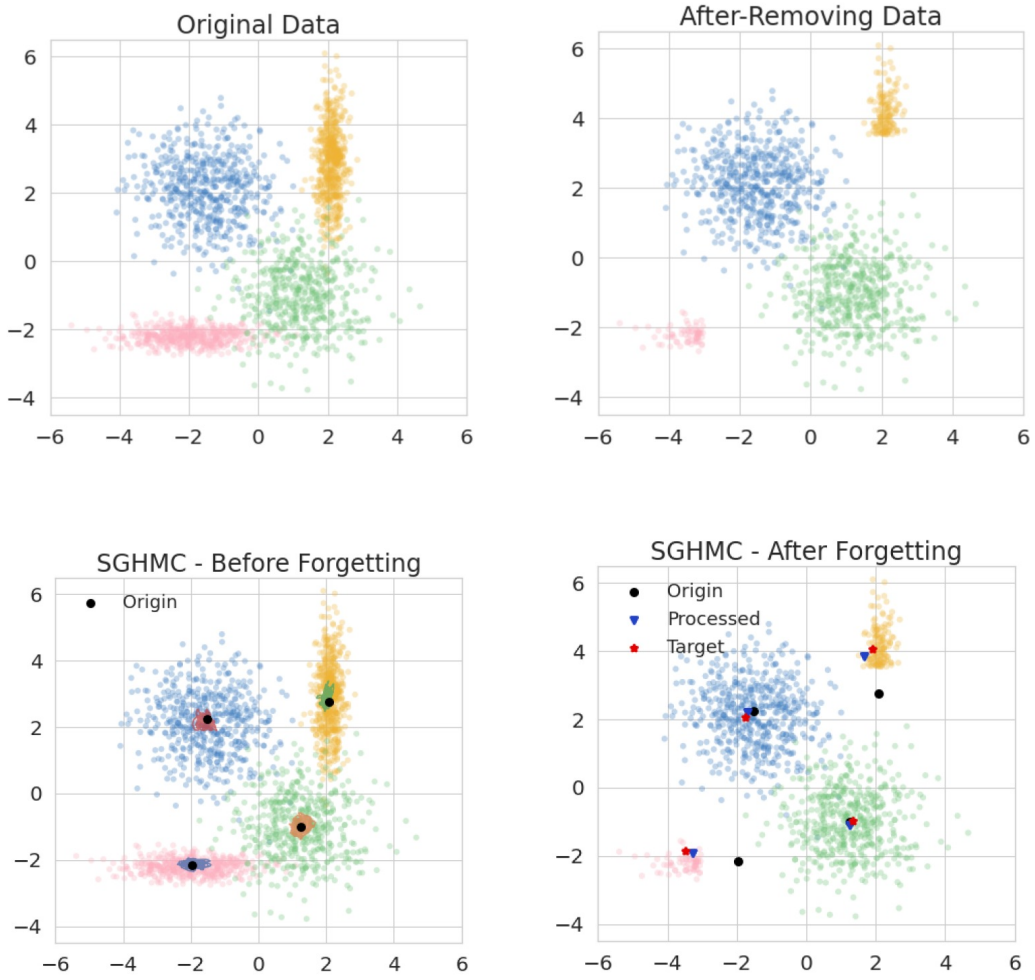
A dataset of size 2,000 was generated for evaluating the algorithm. Every datum is two-dimensional and is possibly from K classes and K was taken to be 4 in the experiment. We've shown the visualization of the raw data where the different colors represent different classes.

6.2 Training phase and Forgetting phase

The training was done for 2,000 iterations and the batch size was set as 64. The learning rate in SGHMC was set as $2t^{-0.15}/n$ and the initial α factor was set as 0.4. In the forgetting phase, 400 points in total were removed from the pink and the yellow classes. A batch of 4 datums was removed each time, and the expectations in the SGHMC influence function was calculated using Monte Carlo method.

6.3 Results

We now show the visualization of the original data and that of the data when 400 points were removed. We also show the clustering results on SGHMC before forgetting and after forgetting.



We can clearly see in the figure that the learnt model is close to the target model. This demonstrates the power of BIF framework for SGHMC. In the BIF paper, the authors have applied the methods for SVI, SGLD as well as Bayesian Neural Networks.

7 What we learnt and Future Work

In this project, we learned about the problem of *Machine Unlearning* and how we can tackle it using the framework of *Variatioanl Inference*. We learnt of two novel techniques, *Adjusted* likelihood and *Reverse KL*, which help in preventing *catastrophic unlearning*. While carrying out our experiment, we got the opportunity to further acquaint ourselves with various python libraries such as sklearn and tensorflow, which will surely help us in our future endeavours.

While the major motivation for studying *Machine Unlearning* stems from the fact that re-training on large remaining datasets can be too cost-inefficient, due to insufficient computational power at hand, we were unable to tread in this direction. Hence, in future, we aim to carry out machine-unlearning using variational inference framework on large datasets and more complex models such as Bayesian Neural Nets.

References

[1] Y. Cao and J. Yang. Towards making systems forget with machine unlearning. *Proc. IEEE S&P*, pages 463–480, 2015.

[2] Shaopeng Fu, Fengxiang He, Yue Xu, and Dacheng Tao. Bayesian inference forgetting, 2021.

[3] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Variational bayesian unlearning, 2020.