SSSE: Efficiently Erasing Samples from Trained Machine Learning Models

Alexandra Peste
IST Austria

Dan Alistarh
IST Austria

Christoph H. Lampert
IST Austria

{firstname.lastname}@ist.ac.at

Abstract

The availability of large amounts of user-provided data has been key to the success of machine learning for many real-world tasks. Recently, an increasing awareness has emerged that users should be given more control about how their data is used. In particular, users should have the right to prohibit the use of their data for training machine learning systems, and to have it *erased* from already trained systems. While several *sample erasure* methods have been proposed, all of them have drawbacks which have prevented them from gaining widespread adoption. In this paper, we propose an efficient and effective algorithm, SSSE, for samples erasure that is applicable to a wide class of machine learning models. From a second-order analysis of the model's loss landscape we derive a closed-form update step of the model parameters that only requires access to the data to be erased, not to the original training set. Experiments on CelebFaces attributes (CelebA) and CIFAR10, show that in certain cases SSSE can erase samples almost as well as the optimal, yet impractical, gold standard of training a new model from scratch with only the permitted data.

1 Introduction

One of the main reasons for the recent success of deep learning for many computer vision tasks is the availability of large user-provided datasets. For example, the popular ImageNet dataset consists of over 14 million images that were publicly accessible on the Internet [DDS+09]. More recently, Facebook disclosed the existence of an in-house dataset which consists of 3.5 billion Instagram images [MGR+18]. For a long time, when users uploaded their data to online services they silently agreed on transferring a broad range of usage rights to the service. However, several legal initiatives, such as the European Union's General Data Protection Regulation (GDPR) [Man13], have been proposed in the recent years, to give users more control over their data, for example the right to withdraw it from the online services at any time. While the legal consequences of such a requirement are so far unclear when it comes to machine learning models, it is a realistic possibility that it would imply that the withdrawn data also has to be erased from already-trained models.

Despite the fundamental nature of the problem of erasing certain training examples from an already trained machine learning model, no satisfactory general-purpose solution exists so far. The gold standard of simply training a new model on all data except the withdrawn part implies storing and reprocessing all samples whenever a single example should be erased, which is not practical for most real-world problems. Although several alternative approaches have been proposed, none of them has found widespread adoption, either in the research community or in commercial applications. A possible explanation is that the proposed methods are either not efficient enough to be practical, or not powerful enough to provide satisfactory results. For example, *machine unlearning* methods designed to accelerate the (re)training of models from subsets of the training data [BCCC+21, CY15, DCL+19, GGVZ19, IASCZ21, LMLM20, WDD20] are typically limited to specific model classes, or they invoke substantial changes to the original training step. Methods using *differential privacy* [DR14] can provide strong guarantees, but typically come with the draw-

Preprint.

backs of more difficult interpretability and reduced prediction accuracy. A method closer to ours consists of the use of *influence functions* [KL17], which quantify the importance of each training sample to the overall model. Despite being a generic and deterministic method, a major challenge is computational tractability, since determining the influence of any sample requires computing and inverting the Hessian matrix of the model's loss, which can be extremely costly for high-dimensional models. Consequently, efficient influence-based data removal has so far only been demonstrated for low-dimensional models [GGHvdM20].

In this work, we aim at closing this gap, by introducing a new method which we call *Single-Step Sample Erasure (SSSE)*. We study the scenario of updating a trained model, to reflect the removal of a subset of training samples. To address the intractability of computing and inverting the Hessian matrix, we approximate it with the empirical Fisher Information Matrix (FIM), which allows easy computation, and fast matrix inversion using rank-one updates. Most related to our work is [GAS20], where the authors also discuss a second-order update step for sample removal, which uses the FIM in place of the Hessian. However, there the FIM is approximated by its diagonal, and an additional noise term is used to ensure the updated model has a similar statistical behaviour to that of one trained from scratch without the removed samples.

SSSE can be used for both convex and non-convex models, as well as for removing a single sample or entire subsets from the training set. Moreover, SSSE is broadly applicable to a wide variety of existing machine learning models, as it puts only minor restriction on *how* the original model was trained. In addition to being analytically justified by means of a Taylor expansion of the model's loss landscape, SSSE is also deterministic. Consequently, it is easy to apply and understand, including for practitioners, allowing them to, for example, check how far their model of choice fulfills SSSE's underlying assumptions. Moreover, SSSE has an important practical advantage, as it only requires access to the data to be deleted, not the original training set, and is also efficient, as samples are erased by simple closed-form updates of the model parameters.

2 Method

Background Consider the setting of supervised learning using a dataset $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\} \subseteq \mathcal{X} \times \mathcal{Y}$ and a twice differentiable loss function $\ell : \mathbb{R}^d \times \mathcal{Y} \to [0, \infty)$. For every $i \in \{1, \dots n\}$, let $\ell_i(\theta)$ be the loss of sample (x_i, y_i). Let $\theta^\star = \operatorname{argmin}_\theta L(\theta)$, where $L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\theta)$. We assume that L is a strictly convex function of θ, with an unique global minimum $\theta^\star$. Given $\mathcal{S} \subset \mathcal{D}$, with $|\mathcal{S}| = k$, we want to update $\theta^\star$ in a single step, such that the new model behaves as if it had been trained from scratch on $\mathcal{D} \setminus \mathcal{S}$. Let $\theta^\star_{-\mathcal{S}} = \operatorname{argmin}_\theta L_{-\mathcal{S}}(\theta)$, where $L_{-\mathcal{S}}(\theta) = \frac{1}{n-k}\sum_{(x_i,y_i)\notin\mathcal{S}} \ell_i(\theta)$. From a first-order Taylor approximation of $\nabla L_{-\mathcal{S}}(\theta^\star_{-\mathcal{S}}) = 0$ around $\theta^\star$, with $H_{-\mathcal{S}}(\theta^\star) = \nabla^2 L_{-\mathcal{S}}(\theta^\star)$, we obtain the following approximation for $\theta^\star_{-\mathcal{S}}$:

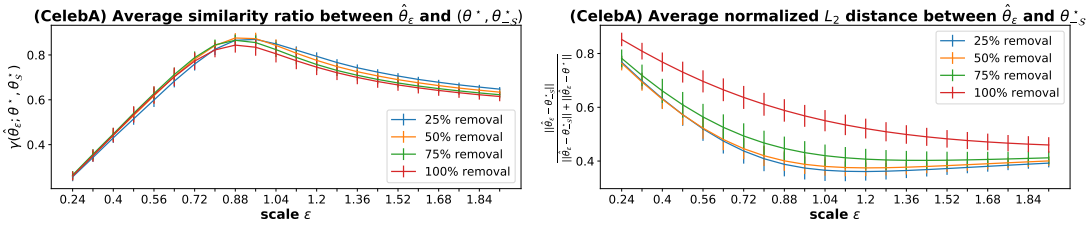
$$\theta^\star_{-\mathcal{S}} \approx \theta^\star + \frac{1}{n-k} H^{-1}_{-\mathcal{S}}(\theta^\star) \sum_{(x_i,y_i)\in\mathcal{S}} \nabla \ell_i(\theta^\star) \tag{1}$$

This update step has been previously used for convex models in [GGHvdM20], where in addition a change to the loss function through random perturbations is proposed. Also, Equation (1) has been used for influence functions [CW80, KL17], except using the Hessian over the full training set $H_{\mathcal{D}}(\theta^\star)$. For defining SSSE we will assume $H_{-\mathcal{S}}(\theta^\star) \approx H_{\mathcal{D}}(\theta^\star)$ and use the latter instead.

Single-Step Sample Erasure For most practical applications, computing and inverting the Hessian is prohibitively expensive. However, when the loss function is the negative log likelihood (i.e. $\ell_i(\theta) = -\log p(y_i|x_i;\theta)$), it is well-known that the expected Hessian over $x \sim p(x)$ and $y \sim p(y|x;\theta)$ is equal to the Fisher Information Matrix (FIM): $F(\theta) = \mathbb{E}_{(x,y)}[\nabla \log p(y|x;\theta) \cdot \nabla^T \log p(y|x;\theta)]$. When the discriminative model $p(y|x;\theta)$ is a good approximation for the true conditional distribution $p(y|x)$, the Hessian matrix can be estimated using the FIM. Since FIM requires multiple gradient computations for each data sample, a good compromise is the use of the *empirical FIM*, which needs only a single gradient computation per sample, by using the true label of each data point. The empirical FIM is defined as:

$$\hat{F}_{\mathcal{D}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \nabla \log p(y_i|x_i;\theta) \cdot \nabla^T \log p(y_i|x_i;\theta) \tag{2}$$

Besides tractability, a major advantage of using the empirical FIM is that it allows efficient inverse computation, without having to perform an explicit matrix inversion. This has been explored in [SA20] for pruning neural networks, and we follow here a similar approach. We construct $\hat{F}^{-1}_{\mathcal{D}}$ incrementally from a sequence of rank-1 updates, by employing the Sherman-Morrison



(a) AUC similarity ratio

(b) Normalized parameters distance

Figure 1: **(CelebA)** The similarity ratio $\gamma_{\mathcal{S}}(\hat{\theta}; \theta^\star, \theta^\star_{-\mathcal{S}})$ and the normalized parameters distance for $\hat{\theta}_\epsilon$ on the removed samples $\mathcal{S}$, as a function of the scaling factor ϵ. The results are averaged across multiple attributes, removed at different rates.

lemma [SM50]. Initially, we choose $\lambda > 0$ and define $\hat{F}_0(\theta) = \lambda I_d$. Afterwards, we incrementally add the contribution of all $(x_i, y_i) \in \mathcal{D}$, for $i = 1, \ldots, n$, by computing at each step:

$$\hat{F}_i^{-1} = \hat{F}_{i-1}^{-1} - \frac{\hat{F}_{i-1}^{-1} \nabla \ell_i \cdot \nabla^T \ell_i \hat{F}_{i-1}^{-1}}{n + \nabla^T \ell_i \hat{F}_{i-1}^{-1} \nabla \ell_i,} \tag{3}$$

where all gradients and estimates are taken at the fixed θ. The final $\hat{F}_n^{-1}$ is identical to the desired $\hat{F}_{\mathcal{D}}^{-1}$. The value of the dampening factor $\lambda > 0$ used for initializing the recurrence for the empirical FIM ensures that the matrix is invertible. Furthermore, we noticed that the elements of the empirical FIM may have a different scale than those of the Hessian, and we therefore add an additional scale hyper-parameter ϵ. Based on these approximations, for $\epsilon > 0$, we define a *Single-Step Sample Erasure (SSSE)* update for a removed subset $\mathcal{S}$ as:

$$\hat{\theta}_\epsilon = \theta^\star + \frac{\epsilon}{n-k} \hat{F}_{\mathcal{D}}^{-1}(\theta^\star) \cdot \sum_{(x_i, y_i) \in \mathcal{S}} \nabla \ell_i(\theta^\star). \tag{4}$$

3 Experiments – Convex Multi-Attribute Classification

We examine the task of erasing samples from the large-scale CelebFaces Attributes Dataset (CelebA) [LLWT15], where the samples are human faces, each with 40 binary attribute annotations. We use features obtained from a VGG-16 [SZ15] neural network pre-trained on the larger VGGFace dataset [PVZ15], and randomly subsample 10% of the train set, on which we fine-tune an ℓ_2-regularized linear multi-attribute classifier, with no bias, to predict each of the 40 binary attributes. We remove different percentages of attributes with at most 20% frequency. Although the weights for each attribute are independent, removing a group of samples with the same attribute has a non-trivial effect on the model, since the data available for the remaining attributes also changes.

Evaluation Method We introduce a method for evaluating how well SSSE removes samples, based on the intuition that the SSSE update $\hat{\theta}_\epsilon$ and the retrained model $\theta^\star_{-\mathcal{S}}$ are similar if their performance is close on all data splits. The most informative data subset for establishing the relative distance from $\hat{\theta}_\epsilon$ to either $\theta^\star$ or $\theta^\star_{-\mathcal{S}}$ is $\mathcal{S}$, since it is part of the train set for $\theta^\star$ and of the test set for $\theta^\star_{-\mathcal{S}}$. For an attribute a, we define $\alpha_i^{\mathcal{S}}(\theta)$ as the area-under-the-curve (AUC) score corresponding to the receiver operating characteristic (ROC) curve, computed on $\mathcal{S}$. If an attribute is absent, its corresponding AUC score will be set, by convention, to 0. AUC scores are preferred for imbalanced data, and they are also robust against the class threshold. We define the *similarity ratio* between $\hat{\theta}_\epsilon$ and both $\theta^\star$ and $\theta^\star_{-\mathcal{S}}$ on the removed subset $\mathcal{S}$, through the following quantity:

$$\gamma_{\mathcal{S}}(\hat{\theta}_\epsilon; \theta^\star, \theta^\star_{-\mathcal{S}}) = \frac{\mathbb{D}_{\mathcal{S}}(\hat{\theta}_\epsilon, \theta^\star)}{\mathbb{D}_{\mathcal{S}}(\hat{\theta}_\epsilon, \theta^\star) + \mathbb{D}_{\mathcal{S}}(\hat{\theta}_\epsilon, \theta^\star_{-\mathcal{S}})} \tag{5}$$

where by notation $\mathbb{D}_{\mathcal{S}}(\theta_1, \theta_2) = \sum_{i=1}^{|A|} |\alpha_i^{\mathcal{S}}(\theta_1) - \alpha_i^{\mathcal{S}}(\theta_2)|$ for any two models θ_1 and θ_2. Clearly, $\gamma_{\mathcal{S}}(\hat{\theta}_\epsilon; \theta^\star, \theta^\star_{-\mathcal{S}}) > 0.5$ implies $\hat{\theta}_\epsilon$ is closer to $\theta^\star_{-\mathcal{S}}$ than to $\theta^\star$, in terms or AUC scores. Therefore, we choose $\epsilon^\star$ as the value achieving $\max_\epsilon \gamma_{\mathcal{S}}(\hat{\theta}_\epsilon; \theta^\star, \theta^\star_{-\mathcal{S}})$. Next, we will see this method of selecting ϵ gives SSSE models that are close, in terms of distance in parameter space, to retraining from scratch.

Results We compare SSSE against retraining from scratch without $\mathcal{S}$. Given $\theta^\star$, we first compute and store $\hat{F}_{\mathcal{D}}^{-1}(\theta^\star)$, and use it later for each sample removal update. We fix the dampening $\lambda = 10^{-4}$,

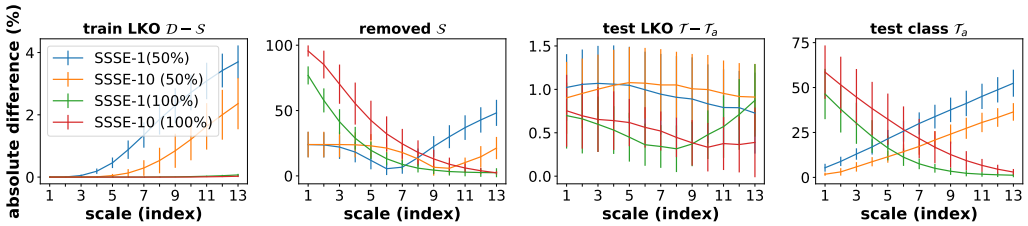


Figure 2: **(CIFAR10)** Absolute difference in accuracy between $\hat{\theta}_\epsilon$ and $\theta^\star_{-\mathcal{S}}$, with different batch sizes for $\hat{F}^{-1}_{\mathcal{D}}(\theta^\star)$ in SSSE. Average and standard deviations over the first 5 classes are reported.

which is equal to the ℓ_2 regularization parameter. In this case, $\hat{F}_{\mathcal{D}}$ has a block-diagonal structure, with 40 equal blocks, corresponding to the individual weights of each attribute. We then simulate the removal of different ratios of a single attribute by computing the SSSE update, while also retraining to obtain each $\theta^\star_{-\mathcal{S}}$. This procedure is repeated for all attributes that appear in at most 20% of the train samples, which consist of almost half of all available attributes. We search the best value of the scale ϵ for SSSE using $\gamma_{\mathcal{S}}(\hat{\theta}_\epsilon; \theta^\star, \theta^\star_{-\mathcal{S}})$ as our performance measure.

The results in Figure 1a show the similarity ratio $\gamma_{\mathcal{S}}(\hat{\theta}_\epsilon; \theta^\star, \theta^\star_{-\mathcal{S}})$ averaged over all chosen attributes, for each scale ϵ, and for different removal percentages. The maximum average similar similarity ratio, across the chosen values of ϵ, is higher than 0.8, which corresponds to SSSE being more than 4 times closer, in terms of AUC scores, to $\theta^\star_{-\mathcal{S}}$ than to $\theta^\star$. Furthermore, the region for ϵ where SSSE is closest to $\theta^\star_{-\mathcal{S}}$ is consistent across different attributes and removal percentages, which suggests that in this case ϵ is indeed a property of $F_{\mathcal{D}}$. As the problem is convex, we can consider the normalized Euclidean distance in parameter space $\frac{\|\hat{\theta}_\epsilon - \theta^\star_{-\mathcal{S}}\|}{\|\hat{\theta}_\epsilon - \theta^\star\| + \|\hat{\theta}_\epsilon - \theta^\star_{-\mathcal{S}}\|}$. Figure 1b shows that, in general, the region with the highest similarity ratio is close to the one where the minimum of the normalized distance is attained. Although we cannot conclude that $\hat{\theta}_\epsilon$ converges exactly to $\theta^\star_{-\mathcal{S}}$, nonetheless SSSE yields a model that is similar in behavior to retraining from scratch.

4 Experiments – Non-Convex Models

SSSE is equally applicable in the non-convex setting. To see this, we train a ResNet20 [HZRS16] architecture, on the CIFAR10 [KH09] dataset, optimized without additional data augmentation, using standard SGD with momentum, for the task of removing either all samples belonging to a single class, or a chosen percentage. We assume a block diagonal structure for $\hat{F}_{\mathcal{D}}$, and use blocks of size 70000. For a class a, we consider $\mathcal{T}_a$ to be all test samples that belong to that class.

Erasing samples from trained deep learning models is difficult, as they can effectively memorize the training set [ZBH+16]. However, SSSE still results in similar performance to a model trained from scratch on $\mathcal{D} \setminus \mathcal{S}$, from the same random seed as $\theta^\star$. In Figure 2 we show the absolute differences between the accuracy of SSSE and of $\theta^\star_{-\mathcal{S}}$, computed on the train and test splits of the available datasets. To improve the efficiency of SSSE, we also approximate FIM using mini-batches of 10 gradients. Such approximations have been also recently used for neural network compression [SA20]. With the appropriate scale ϵ, both methods can achieve similar performance.

We note that the task of erasing all samples from a given class is in fact easier than removing only a subset; for partial removal, even in the "optimal" region SSSE tends to mis-classify the removed class samples more aggressively than $\theta^\star_{-\mathcal{S}}$. A similar effect was also noticed in the experiments with convex models. We believe that improvements could be made to SSSE, to induce more aggressive perturbations in the higher layers, by, for example, computing FIM only on the leave-k-out samples.

5 Discussion

We proposed SSSE, a method for erasing samples from a trained model. It is inspired by influence functions and made efficient through the use of FIM in combination with efficient low-rank matrix updates instead of an intractable Hessian. Our results show that influence-based updates are not just theoretically a good idea for samples erasure, but that, with the right numerical tools, they can actually be made practical. We hope that this insight will inspire other researchers to build on our work and practitioners to add influence-based sample erasure to their toolboxes.

References

[BCCC+21] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy*, 2021. 1

[CW80] R Dennis Cook and Sanford Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4):495–508, 1980. 2

[CY15] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy*, 2015. 1

[DCL+19] Min Du, Zhi Chen, Chang Liu, Rajvardhan Oak, and Dawn Song. Lifelong anomaly detection through unlearning. In *ACM Conference on Computer and Communications Security (CCS)*, 2019. 1

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. In *Foundations and Trends in Theoretical Computer Science*, 2014. 1

[GAS20] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[GGHvdM20] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning (ICML)*, 2020. 2

[GGVZ19] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making AI forget you: Data deletion in machine learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 1

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[IASCZ21] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021. 1

[KH09] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009. 4

[KL17] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, 2017. 2

[LLWT15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)*, December 2015. 3

[LMLM20] Yang Liu, Zhuo Ma, Ximeng Liu, and Jianfeng Ma. Learn to forget: User-level memorization elimination in federated learning. *arXiv preprint arXiv:2003.10933*, 2020. 1

[Man13] Alessandro Mantelero. The EU proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013. 1

[MGR+18] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision (ECCV)*, 2018. 1

[PVZ15] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference (BMVC)*, 2015. 3

[SA20] Sidak Pal Singh and Dan Alistarh. Woodfisher: Efficient second-order approximations for model compression. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 2, 4

[SM50] Jack Sherman and Winifred J Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950. 3

[SZ15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015. 3

[WDD20] Yinjun Wu, Edgar Dobriban, and Susan B. Davidson. Deltagrad: Rapid retraining of machine learning models. In *International Conference on Machine Learning (ICML)*, 2020. 1

[ZBH+16] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2016. 4