Article

Exact Unlearning with Convex and Non-Convex Functions

Cassandra Lindstrom *

Posted Date: 26 September 2024

doi: 10.20944/preprints202409.2061.v1

Keywords: exact unlearning; machine unlearning; convex function; non-convex function

 doi:10.20944/preprints202409.2061.v

Article

Exact Unlearning with Convex and Non-Convex Functions

Cassandra Lindstrom

Affiliation 1; cli1194@bloomberg.net

Abstract: Machine unlearning, the process of selectively forgetting or removing the influence of specific data points from a machine learning model, is increasingly important for privacy and compliance with regulations like the GDPR. This paper explores the concept of exact unlearning, focusing on its implementation in models trained using convex and non-convex functions. Convex functions, due to their well-behaved optimization landscapes, lend themselves to efficient unlearning through methods such as inverse optimization, duality-based approaches, and incremental learning. In contrast, non-convex functions, common in deep learning models, present more complex challenges due to their multiple local minima and high-dimensional parameter spaces. Techniques like checkpoint-based retraining, gradient inversion, and meta-learning are discussed as viable, though computationally expensive, methods for non-convex exact unlearning. The paper also highlights real-world applications in fields such as finance and healthcare, where exact unlearning can enhance privacy and security without compromising model performance. Finally, it outlines key challenges and future research directions, particularly the need for more efficient unlearning algorithms in non-convex settings and the development of secure, adversarial-resistant methods for sensitive data removal.

Keywords: exact unlearning; machine unlearning; convex function; non-convex function

Introduction

The advent of machine unlearning has sparked significant interest in creating methods that effectively erase or forget specific data from trained models. As machine learning applications scale up, so do the concerns about data privacy, security, and regulatory compliance. Exact unlearning aims to fully remove the influence of specific data points without retraining models from scratch. While unlearning is generally difficult, it becomes more complex when dealing with different types of optimization functions, especially in convex and non-convex settings. This paper aims to explore the intricate dynamics of exact unlearning within these frameworks, analyze their theoretical foundations, and present practical implementations across various domains. We are going to take a look at exact unlearning using both convex and non-convex functions.

Literature Review

Research in data privacy, especially with the advent of the General Data Protection Regulation (GDPR), has driven the need for efficient unlearning mechanisms. This need is exacerbated in graph data where the erasure of one node or edge could impact the global structure. Related work in data deletion, differential privacy, and adversarial robustness provides a foundation, but applying these techniques to graph data remains a challenge.

Exact unlearning approach are the ones that forget certain points in a relatively determined way. This requires retraining sub-models or performing complex mathematical computations, leading to inefficiencies and limited applicability in complex models and large datasets according to (Cao & Yang, Towards making systems forget with machine unlearning, 2015) and (Li, et al., 2024).

The conventional exact unlearning uses convex function. (Cao, et al., 2018) use SVM and Bayesian-based classifiers for its exact unlearning approach. (Schelter, 2019) relies on logistic regression for its exact unlearning. (Jose & Simeone, 2021) leverages on Bayesian model while (Kashef, 2021) uses SVMs.

 doi:10.20944/preprints202409.2061.v

At the same time, there are complex models with non-convex functions. (Ullah, Mai, Rao, Rossi, & Arora, 2021) presents models with non-convex functions. (Brophy & Lowd, 2021) applied non-convex model to random forests. (Schelter, 2019) applies non-convex model to randomized trees. (Bourtoule, et al., 2021) and (Yan, et al., 2022) contribute to this topic by applying it to DNN (Deep Neural Network). (Chen, et al., 2022) and (Wang, Huai, & Wang, 2023) have the application to GNN (Graph Neural Network). Emerging work in graph learning, including Graph Neural Networks (GNNs), has only recently begun addressing privacy-preserving and unlearning mechanisms. (Li, et al., 2024) and (Shaik, et al., 2023) have provided a summary of the most relevant research on federated unlearning. (Wang, et al., 2024) has proved that GNN is very successful in representing complex relationships in machine learning. When GNN framework is combined with treasury (Li, Wang, & Chen, Incorporating economic indicators and market sentiment effect into US Treasury bond yield prediction with machine learning, 2024) and crypto trading (Li, Wang, & Chen, A Contrastive Deep Learning Approach to Cryptocurrency Portfolio with US Treasuries, 2024),it becomes very powerful in machine unlearning.

Convex Functions and Exact Unlearning

Convex functions possess specific characteristics that make exact unlearning more feasible. A function is convex if its epigraph, the set of points lying on or above its graph, forms a convex set. Convexity ensures that any local minimum is also a global minimum, simplifying optimization and retraining tasks. In exact unlearning, the objective is to remove the effect of a certain data point without requiring the entire model to be retrained from scratch. For convex models, such as linear regression or support vector machines (SVMs), removing data points from the training set can be achieved through analytical adjustments to the model's parameters. The convexity ensures that the retraining process follows predictable patterns, often allowing for incremental learning or unlearning methods to converge rapidly to the desired state.

One key advantage of convex functions is the availability of gradient-based optimization methods that converge quickly and efficiently. Techniques such as stochastic gradient descent (SGD) allow updates to model parameters based on small batches of data, making the removal of specific data points computationally easier in exact unlearning. For instance, in linear models, recalculating the weight vector after removing a data point can be done by solving a modified set of equations derived from the original optimization problem. This procedure avoids the need for full retraining while ensuring that the model's performance is unaffected by the removed data.

Non-Convex Functions and Exact Unlearning

In contrast to convex functions, non-convex functions introduce substantial challenges to the exact unlearning process. A non-convex function may have multiple local minima, making it difficult to guarantee that removing a particular data point will lead to predictable changes in model behavior. Non-convex optimization problems are common in neural networks and deep learning models, where the loss function often has numerous minima due to the complex nature of the function landscape.

Exact unlearning in non-convex settings requires more sophisticated methods to ensure that the model behaves as though the removed data point was never present. The primary challenge lies in ensuring that the optimization process does not get trapped in local minima after data removal. Various techniques, such as second-order optimization methods, trust region methods, or meta-learning approaches, can help navigate the complex landscape of non-convex functions. However, these methods are computationally expensive and may not always converge to a globally optimal solution.

Moreover, neural networks and other non-convex models often have a high-dimensional parameter space, which exacerbates the difficulty of exact unlearning. The removal of even a single data point can affect thousands of parameters, requiring careful recalibration of the entire model. Exact unlearning algorithms in non-convex settings typically involve fine-tuning the model post-unlearning, where gradient-based methods are employed to correct any shifts in model behavior.

 doi:10.20944/preprints202409.2061.v

However, this often leads to approximation rather than exact solutions, making it difficult to fully eliminate the influence of removed data points.

Methods for Exact Unlearning in Convex and Non-Convex Settings

Exact unlearning involves making trained machine learning models "forget" specific data points without needing to retrain them from scratch. While the challenge of unlearning varies depending on the underlying optimization function, the methods differ significantly between convex and non-convex settings due to their respective mathematical properties. Below is a detailed exploration of methods used for exact unlearning in convex and non-convex models.

1. Exact Unlearning in Convex Settings

Convex optimization problems, where the loss function is convex, exhibit properties that make exact unlearning more tractable. Since convex functions have a unique global minimum, the optimization process is simpler and the effects of removing individual data points can be more easily reversed. Some of the primary methods include:

a. Inverse Optimization

Inverse optimization techniques attempt to reverse the optimization process to “undo” the effects of a particular data point that was part of the training set. Given the convex nature of the optimization problem, the relationship between the model parameters and the loss function is well-defined and predictable.

In practice, inverse optimization works by solving an optimization problem that identifies the new model parameters after the removal of the selected data point. For convex models, this can often be done efficiently, especially when the training data points are independent or have separable effects on the model parameters. This method is applicable to models such as linear regression, logistic regression, and support vector machines (SVMs), where the solution space is linear or near linear.

For example, in linear regression, the weight vector w is computed based on the sum of contributions from individual data points. If a specific data point is removed, the new weight vector can be recalculated by solving the linear system without that point’s contribution. This is possible because the linearity of the problem allows for an analytical solution.

b. Duality-Based Methods

Convex optimization problems often have a corresponding dual problem, where the constraints of the original (primal) problem are transformed into a different but related form. The dual problem provides insights into how the solution to the primal problem will change as data points are added or removed. By analyzing the dual problem, it is possible to derive closed-form updates for model parameters after data deletion.

Duality-based methods exploit this relationship by solving the dual optimization problem to directly update the model parameters. Since convex functions guarantee that the solution space remains stable, the dual formulation provides an efficient way to calculate new model parameters without the need for full retraining. These methods are particularly effective in models where the relationship between the data points and the model parameters is governed by constraints, such as SVMs.

c. Incremental and Decremental Learning

Incremental learning refers to updating the model as new data is added, whereas decremental learning refers to adjusting the model as data points are removed. In convex models, these updates can be performed efficiently without retraining the entire model. When a data point is removed, decremental learning algorithms adjust the model parameters in a way that the resulting model is as if the data point had never been included in the training set.

For example, in linear models, removing a data point involves recomputing the inverse of the covariance matrix (used in ordinary least squares) and adjusting the weight vector. These updates can be computed in closed form, making exact unlearning highly efficient for convex models.

2. Exact Unlearning in Non-Convex Settings

In non-convex settings, where the loss function has multiple local minima, the task of exact unlearning becomes much more challenging. Non-convex functions often have complex, high-

 doi:10.20944/preprints202409.2061.v

dimensional landscapes, making it difficult to guarantee that removing a particular data point will result in a predictable change in the model's parameters. However, several methods have been proposed to address these challenges.

a. Checkpoint-Based Retraining

Checkpoint-based retraining involves periodically saving model states during training. When a data point needs to be unlearned, the model can be "rolled back" to a checkpoint that was saved before the data point was included in the training set. From that checkpoint, the model is retrained, excluding the data point to be removed.

While this method is not strictly "exact," as the model is retrained from a saved intermediate state, it provides an approximation to exact unlearning without requiring the model to be retrained from scratch. In practice, the performance of this method depends on the frequency of checkpointing. More frequent checkpoints result in more accurate unlearning but increase storage and computational overhead.

b. Gradient Inversion and Gradient Subtraction

Non-convex models, such as deep neural networks, rely heavily on gradient-based optimization techniques like stochastic gradient descent (SGD) for training. One approach to exact unlearning in these models is to reverse the effect of the gradient updates associated with the data point to be unlearned. This can be done by subtracting the gradients computed for the data point from the model parameters.

Formally, during training, the model's parameters θ are updated as:

$$\theta_{t+1} = \theta_t - \eta \nabla L(xi, \theta t)$$

where η is the learning rate, $L(xi, \theta t)$ is the loss function for data point x_i, and ∇ is the gradient. To unlearn the contribution of x_i, the model parameters are updated as $\theta' = \theta - (-\eta \nabla L(xi, \theta))$. By effectively "inverting" the gradient, this method attempts to cancel out the contribution of the data point. However, due to the non-convex nature of the optimization problem, the effect of this gradient subtraction may not completely restore the model to a state that would have existed if the data had never been used. This method works best when the contribution of the removed data point to the loss function is small.

c. Data Perturbation Techniques

In non-convex models, exact unlearning can sometimes be approximated by perturbing the data and adjusting the model parameters accordingly. Data perturbation techniques involve modifying the input data to ensure that the influence of the removed data point is neutralized.

For example, one approach is to introduce noise or modify the remaining data points in the training set to "compensate" for the removal of a particular point. By carefully adjusting the training data, it is possible to minimize the difference between the model trained with and without the removed data point. However, this technique is generally more useful for approximate unlearning rather than exact unlearning, as it introduces additional complexities into the training process.

d. Meta-Learning Approaches

Meta-learning, or "learning to learn," is an emerging field in machine learning where models are trained to rapidly adapt to new data. In the context of exact unlearning, meta-learning can be used to train models that can efficiently forget specific data points when required. The key idea is to train the model not only on the task at hand but also on the ability to remove or unlearn data points without significantly altering the overall model.

One meta-learning approach is to design a model that includes unlearning as part of its training objective. During training, the model is exposed to data points that are later removed, and it learns to adapt to the absence of these data points. This creates a model that is inherently more robust to the removal of specific data, making exact unlearning more feasible in non-convex settings.

3. Hybrid Methods for Complex Models

In practice, many machine learning models involve both convex and non-convex components. For example, deep learning models often include convex layers (such as linear or convolutional layers) alongside non-convex activation functions. In such cases, hybrid unlearning methods can be

 doi:10.20944/preprints202409.2061.v

employed, where convex components are handled using inverse optimization or duality methods, while non-convex components rely on gradient-based techniques or meta-learning.

Applications of Exact Unlearning

The theoretical foundations of exact unlearning have important implications for real-world applications. In fields such as finance, healthcare, and privacy-preserving machine learning, ensuring that sensitive data can be removed from trained models is essential for compliance and security. Exact unlearning can be particularly useful in dynamic environments where models need to adapt quickly to changing data distributions without retraining from scratch.

In finance, for example, machine learning models are increasingly used for fraud detection, risk assessment, and portfolio optimization. As financial data is often sensitive and subject to regulations, exact unlearning can be used to remove specific transactions or client data from these models without compromising their overall performance. Convex optimization models like linear regression are commonly used in financial risk assessments, making exact unlearning feasible using methods like inverse optimization.

In healthcare, machine learning models trained on patient data must often comply with strict privacy regulations such as the Health Insurance Portability and Accountability Act (HIPAA). Exact unlearning ensures that sensitive patient information can be removed from predictive models without retraining, allowing healthcare providers to offer personalized services while maintaining compliance with privacy laws.

Conclusion

Exact unlearning is a vital area of research that addresses the growing need to erase sensitive data from machine learning models without requiring full retraining. The distinction between convex and non-convex functions plays a critical role in determining the feasibility and efficiency of exact unlearning. Convex models, with their predictable optimization properties, offer more straightforward solutions, while non-convex models pose substantial challenges due to the complexity of their loss landscapes. The development of effective unlearning methods in both settings has wide-ranging implications for privacy, security, and compliance in machine learning systems, particularly in fields such as finance and healthcare. Future research must focus on overcoming the computational limitations of exact unlearning in non-convex models and developing secure, adversary-resistant algorithms to ensure the broad applicability of these techniques.

References

1. Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., . . . Papernot, N. (2021). Machine unlearning. *2021 IEEE Symposium on Security and Privacy (SP)*, (pp. 141–159).
2. Brophy, J., & Lowd, D. (2021). Machine unlearning for random forests. *International Conference on Machine Learning*, (pp. 1092–1104).
3. Cao, Y., & Yang, J. (2015). Towards making systems forget with machine unlearning. *2015 IEEE symposium on security and privacy*, (pp. 463–480).
4. Cao, Y., Yu, A. F., Aday, A., Stahl, E., Merwine, J., & Yang, J. (2018). Efficient repair of polluted machine learning systems via causal unlearning. *Proceedings of the 2018 on Asia conference on computer and communications security*, (pp. 735–747).
5. Chen, M., Zhang, Z., Wang, T., Backes, M., Humbert, M., & Zhang, Y. (2022). Graph unlearning. *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*, (pp. 499–513).

6. Jose, S. T., & Simeone, O. (2021). A unified PAC-Bayesian framework for machine unlearning via information risk minimization. *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, (pp. 1–6).
7. Kashef, R. (2021). A boosted SVM classifier trained by incremental learning and decremental unlearning approach. *Expert Systems with Applications, 167*, 114154.
8. Li, N., Zhou, C., Gao, Y., Chen, H., Fu, A., Zhang, Z., & Shui, Y. (2024). Machine Unlearning: Taxonomy, Metrics, Applications, Challenges, and Prospects. *arXiv preprint arXiv:2403.08254*.
9. Li, Z., Wang, B., & Chen, Y. (2024). A Contrastive Deep Learning Approach to Cryptocurrency Portfolio with US Treasuries. *Journal of Computer Technology and Applied Mathematics, 1*, 1-10. doi:10.5281/zenodo.13357988
10. Li, Z., Wang, B., & Chen, Y. (2024). Incorporating economic indicators and market sentiment effect into US Treasury bond yield prediction with machine learning. *Journal of Infrastructure, Policy and Development, 8*, 7671. doi:10.24294/jipd.v8i9.7671
11. Schelter, S. (2019). amnesia–towards machine learning models that can forget user data very fast. *1st International Workshop on Applied AI for Database Systems and Applications (AIDB19).*
12. Shaik, T., Tao, X., Xie, H., Li, L., Zhu, X., & Li, Q. (2023). Exploring the landscape of machine unlearning: A comprehensive survey and taxonomy. *arXiv preprint arXiv:2305.06360*.
13. Ullah, E., Mai, T., Rao, A., Rossi, R. A., & Arora, R. (2021). Machine unlearning via algorithmic stability. *Conference on Learning Theory*, (pp. 4126–4142).
14. Wang, C.-L., Huai, M., & Wang, D. (2023). Inductive graph unlearning. *32nd USENIX Security Symposium (USENIX Security 23)*, (pp. 3205–3222).
15. Wang, Z., Zhu, Y., Li, Z., Wang, Z., Qin, H., & Liu, X. (2024). Graph neural network recommendation system for football formation. *Applied Science and Biotechnology Journal for Advanced Research, 3*, 33–39. doi: 10.5281/zenodo.12198843
16. Yan, H., Li, X., Guo, Z., Li, H., Li, F., & Lin, X. (2022). ARCANE: An Efficient Architecture for Exact Machine Unlearning. *IJCAI*, *6*, p. 19.