Netflix And Severance

Mimee Xu[*]
Department of Computer Science
Courant Institute of Mathematical Sciences, New York University
mimee@nyu.edu

Jiankai Sun
ByteDance Inc.

Xin Yang
ByteDance Inc.

Kevin Yao
ByteDance Inc.

Chong Wang
ByteDance Inc.

Abstract

Suppose a person, who has streamed rom-coms exclusively with their significant other, suddenly breaks up. Consider an expecting mom, who has shopped for baby clothes, miscarries. Their streaming and shopping recommendations, however, do not necessarily update, serving as unhappy reminders of their loss. One approach is to implement the Right To Be Forgotten for recommendation systems built from user data, with the goal of updating downstream recommendations to reflect the removal without incurring the cost of re-training. Inspired by solutions to the original Netflix challenge [Koren, 2009], we develop Unlearn-ALS, which is more aggressively forgetful of select data than fine-tuning. In theory, it is consistent with retraining without model degradation. Empirically, it shows fast convergence, and can be applied directly to any bi-linear models regardless of the training procedure.

1 Introduction

Break-ups, pregnancy losses, and bereavements are particularly painful in the age of ubiquitous machine learning systems. Suppose a user watches a Korean drama with their significant other but breaks up mid-season. They are subsequently bombarded with new episode alerts and recommended shows with the same actors and art styles, potentially causing distress. To move on, the user may reasonably demand some of their past records expunged from Netflix's recommendation engines.

The platform should accommodate deletion, not only in user history but also in subsequent recommendations. Ideally, this deletion is both swift and seamless. An incomplete "under-deletion" likely persists the underlying concepts learned from the deleted records, preventing the user from cultivating a new path forward due to "echo-chamber" style feedback [Chaney et al., 2018, Jiang et al., 2019, Mansoury et al., 2020]. Yet, a callous reset may needlessly degrade the model. Worse, an "over-deletion" introduces new privacy risks when popular items are conspicuously missing.

Fortunately, most deployed recommendation systems rely on matrix completion, which assumes user and movie features to be low-rank. Since recommender training performs dimensionality reduction, the degrees of freedom for the model to memorize is limited. Further, bi-linear models predict linear relationships between a user and a movie's features, which is less expressive than non-linear models.

We thus develop Unlearn-ALS, which modifies the intermediate confidence matrix used in the heuristic optimization of Alternating Least Squares (ALS) to achieve fast forgetting. Mathematically, Unlearn-ALS is equivalent to minimizing the loss of the model on the remaining data by retraining with ALS, making our method an instance of *exact* deletion. We further ask, is our work done?

[*]Work done during internship at ByteDance.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

In theory, if linear models of few parameters fail to memorize individual samples, they may be "robust" to user-requested deletions. In practice, however, industrial recommendations and systems trained with differential privacy do leak training data with specific users [Calandrino et al., 2011, Rahman et al., 2018]. The disparity underscores the importance of empirical evaluation.

Our contributions (1) We clarify that practical bi-linear recommendation models have privacy risks from memorizing training data. (2) We propose Untrain-ALS, a crafty and fast heuristic that unlearns a bi-linear model, and makes no compromise to recommendation accuracy. (3) In future work, demonstrate the risks of evaluating unlearning exclusively with membership inference.

2 Problem Setup

We assume a base collaborative filtering model based on matrix factorization, learned through a user-item matrix, structured similarly to MovieLens [Bennett et al., 2007]. The downstream recommendation for each user is given based on the ranking of items [Koren et al., 2009].

Matrix Completion. The platform observes ratings matrix P where $p_{ij} := P[i][j]$ denotes the preference of user i with respect to item j; if the interaction is not observed, $p_{ij} = 0$. Because the matrix of true preferences cannot be fully observed, entries of P are assumed sampled from ground truth matrix M. In matrix factorization, M can be recovered through a low rank multiplication,

$$M = XY^\intercal. \tag{1}$$

where X depicts user features over all users, and Y is the underlying item factors.

Algorithm 1 AlternatingLeastSquares

Require: P, α, λ, initialize X, Y randomly.
  $c_{ui} \leftarrow 1 + \alpha p_{ui}$ ▷ [Hu et al., 2008]
  while model does not converge **do**
    for all u **do**
      $x_u \leftarrow (Y^\intercal C^u Y + \lambda I)^{-1} C^u P^u$
    end for
    for all i **do**
      $y_i \leftarrow (X^\intercal C^i X + \lambda I)^{-1} C^i P^i$
    end for
  end while
  return X, Y as $\hat{X}, \hat{Y}$

Algorithm 2 Untrain-ALS

Require: $P, \alpha, \lambda, \hat{X}, \hat{Y}, C_0, \mathcal{D}_{\text{removal}}$
  $X, Y \leftarrow \hat{X}, \hat{Y}$ ▷ from Algorithm 1
  for all $(u, i) \in \mathcal{D}_{\text{removal}}$ **do**
    $p_{ui} \leftarrow 0, c_{ui} \leftarrow 0$ ▷ delete and block
  end for
  while model does not converge **do**
    for all u **do**
      $x_u \leftarrow (Y^\intercal C^u Y + \lambda I)^{-1} C^u P^u$
    end for
    for all i **do**
      $y_i \leftarrow (X^\intercal C^i X + \lambda I)^{-1} C^i P^i$
    end for
  end while
  return X, Y as $\hat{X}, \hat{Y}$

Alternating Least Squares (ALS). For given ratings matrix P and desirable rank k, we learn the model parameters $\hat{\theta} = \{\hat{X}, \hat{Y}\}$. The loss function is the regularized matrix completion:

$$L_{\text{ALS}}(X, Y) = \sum_{(u,i) \in \mathcal{D}_{\text{obs}}} c_{ui}(p_{ui} - x_u^\intercal y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2) \tag{2}$$

where we use $\mathcal{D}_{\text{obs}} = \{(u, i)\}$ to denote the coordinates of M that contain explicit observations.

Unless otherwise mentioned, we train (and re-train) with AlternatingLeastSquares (ALS), a widely deployed heuristic by Hu et al. [2008], Takács et al. [2011] outlined in Algorithm 1. ALS is exceedingly simple and parallelizable; despite having little theoretic guarantee it converges fast empirically for recommendation data [Koren et al., 2009, Jain et al., 2013, Uschmajew, 2012].

The key insight lies in making a non-convex optimization convex at each of the alternating minimizations. To tackle implicit feedback, a confidence matrix C is constructed as a soft copy of the ratings, where $c_{ui} := 1 + \alpha p_{ui}$ for $\alpha \in \mathbf{R}^+$: if the ratings were high, the confidence is high, and if the ratings are missing, the confidence is low. The better-behaving C is then used throughout the iterations.

Though we treat ALS as the baseline ground truth for training (and re-training), our unlearning algorithm, Untrain-ALS, applies to any bi-linear model. See Appendix B for experiment parameters.

Additional Assumptions. The removal set, $\mathcal{D}_{\text{removal}}$, is uniformly sampled from $\mathcal{D}_{\text{removal}}$ without replacement, and it cannot be known prior to training. Further, the coordinates in $\mathcal{D}_{\text{obs}}$ are assumed to be i.i.d., to ensure that models trained without access to the deleted data are statistically independent from the removal set. Lastly, $|\mathcal{D}_{\text{obs}}| \gg |\mathcal{D}_{\text{removal}}|$ to simulate occasional deletion requests.

Re-training as Privacy Baseline. As our goal is to neither over- nor under-delete, the ideal removal of $P[m][n]$ is to train another model with new preference matrix P' where $P'[m][n] = 0; P'[i][j] = P[i][j] \text{ otherwise}$. The retrained model will thus treat the removed samples as simply missing data, as Hu et al. [2008]'s *implicit* feedback, ensuring privacy requirements. Additionally, we are only concerned with cases where $P_{mn} \neq 0$ so that the deletion is meaningful.

Machine Unlearning [Bourtoule et al., 2021]. The intuition behind unlearning is similar to that of fine-tuning: the pre-trained model has learned useful concepts that we want to take advantage of; however, the goal of unlearning is to re-fit the model, adjusting away from the deleted items.

Empirical Evaluations. In our setup, after unlearning procedure, the removed data should "look like" data that was not observed. In Membership Inference (MI), the trained model's outputs can be exploited to judge whether a data sample was part of the training data. Typically, an MI classifier $\sigma(\mathcal{M}) : (x) \to \{0, 1\}$ is a binary logistic regressor. Our MI training set is constructed with positive data of actual training samples' outputs, and negative data of removed training samples' outputs.

Nonetheless, a robust unlearning does not require an associated low MI accuracy. Instead, we are concerned with *increased* confidence in membership attack caused by the unlearning procedure.

Vulnerability. Fixing the training procedure, the re-trained model and the trained model can be seen as a function of their observed ratings matrix. Let $\text{MI}(\cdot) : (\theta, \mathcal{D}_{\text{removal}}, \mathcal{D}_{\text{remain}}) \to [0, 1]$, which refers to the membership inference accuracy on a particular model given the removal set and the remaining set. Because all the evaluations fix the datasets between retraining and untraining, we simply write $\text{MI}(\text{untrain})$ to refer to membership inference accuracy with untraining.

Typically, MI is directly used as a vulnerability measure. As we compare against re-training from scratch, the *additional* vulnerability caused by the choosing untraining over retraining is written as $\text{MI}(\text{untrain}) - \text{MI}(\text{retrain})$. In Section D.3, we propose instead to use $\text{MI}(\text{unlearn}) - \text{MI}(\text{train}) - \text{MI}(\text{undeleted})$ under fixed data splits, to denoise the effect of the base undeleted model.

3 Inherent Robustness of Matrix Completion (Compressed)

In theory, matrix completion model is robust to random deletions, but there is no guarantee in practice for individual users. Nevertheless, the theoretic results imply that membership attacks against a well-validated model may be especially challenging. We extend this section in Appendix E.

Theoretic results. We make a key observation: with implicit feedback, our setup selects removal and test data in the same way; moreover, in preference matrix, their corresponding entries are zeroed. A well-validated model is thus, on average, inherently robust to missing data. Rehashing the key claim in Recht [2011] we also show that the exact solutions to matrix completion is inherently robust to randomly sampled deletions under mild assumptions on the data; see Appendix E).

Practical limitations. Model training typically employs regularization (Equation 2), and early-stopped at the best fit (Algorithm 1), not to completion. Plus, we cannot judge the matrix coherence of real world data as required [Recht, 2011]. Lastly, the decompositions learned using ALS can be non-unique (nor equivalent up to a rotation) [Jain et al., 2013], so the removal samples may be especially vulnerable with respect to the currently deployed model, thus requiring manual deletion.

4 Unlearn-Alternating Least Squares (Unlearn-ALS)

Our unlearning strategy, Unlearn-ALS, takes advantage of the fast heuristic used in training implicit feedback recommendations and makes slight modifications:

1. **Pre-train**. Use the resulting X_0, Y_0 in Algorithm 1 to initialize ALS.
2. **Deleting preferences**. Set $p_{ui} = 0$ for deleted item-user interaction i, u.

3. **Blocking confidence on removed data**. Set $c_{ui} \leftarrow 0$ for deleted item-user interaction i, u at all subsequent iterations. Crucially this prevents further influence of the deleted data, thus allowing the model to refit to the remaining data fast. Optionally, use adjusted inverse.

4.1 Untrain Loss = Retrain Loss

Recall that the holy grail of unlearning is to approximate retraining. Under these modifications to p_{ui} and c_{ui}, we find the loss function of Untrain-ALS is functionally equivalent to re-training, derived in Appendix A . The extraneous terms relating to removal data is fully zero-ed. We thus claim that optimizing Untrain-ALS *can* achieve the same loss as retraining without the removal data.

Remark 1 *It may appear that with such strong results, our work is over. Yet again, two real-world issues prevent us from claiming any untrained model is the same as any retrained model: 1. empirically, the models are trained with early stopping: the number of epochs to train is determined by minimal loss; and 2. matrix factorization solutions via ALS are not unique. For empirical privacy, some of the potential solutions may be* more private *than others. It is therefore crucial that we complement with empirical privacy measures.*

4.2 Untrain Runtime $\leq$ Training Runtime, Per Pass

Algorithms 1 and 2 show that, per-iteration, Unlearn-ALS has the same runtime as a pass of ALS. Its convergence analysis is therefore similar to that of ALS itself such as in Uschmajew [2012]. Because the loss of the pre-trained model is minimal, it is easy to see that converging using Untrain-ALS would be much faster than doing ALS from scratch.

Speedups. Every default pass of ALS requires inverting a large matrix. Though fast implementations use conjugate gradient (CG) to approximate inverses [Takács et al., 2011], we note a faster alternative for exactly computing the matrix inverse in Untrain-ALS, where the original inverse is already available. Adjusting for $c_{ui} \leftarrow 0$ is equivalent to changing a single entry in the diagonal matrix C^u. This subtraction of a one-entry matrix is the perturbation of concern. The resulting confidence matrix under un-training, $\widetilde{C^u}$, is very close to the original confidence matrix, where

$$\widetilde{C^u} := C^u - (\text{diag}[0, \cdots, c_{ui}, \cdots, 0]). \tag{3}$$

Consider a special case of Woodbury's inverse [Woodbury, 1950] where only one element is subtracted, by Sherman and Morrison [1950]'s subtraction case, for matrix A, there is $(A - uv^\intercal)^{-1} = A^{-1} + A^{-1}u(1 - v^\intercal A^{-1}u)^{-1}v^\intercal A^{-1}$. Let $A := Y^\intercal C^u Y + \lambda I$. The adjusted inverse

$$(\widetilde{A})^{-1} = (Y^\intercal C^u Y + \lambda I)^{-1} + \frac{c_{ui}}{1-q} y_i (Y^\intercal C^u Y + \lambda I)^{-1} y_i^\intercal (Y^\intercal C^u Y + \lambda I)^{-1}.$$

Overall Runtime. ALS and Untrain-ALS runtimes are both $O(|\mathcal{D}_{\text{obs}}|k^2 + n^3 k)$. With the inverse adjustment, recall in ALS, the inverse of A is computed in $O(k^3)$, and using CG speeds it up to $O(k^2 p)$ where p is the number of CG iterations. Assuming A^{-1} has been computed in the pretraining step, the adjustment is a perturbation on A, which we project to its inverse. This allows for a runtime of $O(k^2)$ per user or item per iteration, making every Untrain-ALS pass $O(|\mathcal{D}_{\text{obs}}|k^2)$.

4.3 Numerical Results

We perform extensivenumerical simulations to show the exact deletion conclusions in our method, and we empirically demonstrate the efficiency of Unlearn-ALS using MovieLens data [Bennett et al., 2007]. Appendix D includes diagrams, and Appendix B states our parameters.

5 Conclusion

A matrix completion-based models cannot be inherently private. We develop Unlearn-ALS, whose objective function aligns with re-training exactly, meaning there is no degradation in model performance caused by choosing unlearning over re-training. We further tackle the scale problem with a numerical speedup with Woodbury inverse adjustments [Woodbury, 1950], which makes it fast to unlearn a few data points from a large matrix.

6 Safety Impact

Preventing Trauma Fixation Through Real-time Forgetting. Humans can choose the memories to discount while ML systems can not. As humans grow reliant on relentlessly un-forgetful systems, these technologies easily violate the user's privacy when they need it the most. We tackle an alignment problem motivated by natural discontinuities in human preferences. Endowing The Right To Be Forgotten is one of the reasonable solutions. We guarantee provably exact forgetting in recommendation-style systems like Netflix, which have stood the test of time to be ubiquitous.

Exact Deletion Over Other Approximations. As theorized in the movie "Eternal Subshine on The Spotless Mind", jerkish lobotomy to enforce forgetting is undesirable. We explicitly avoid overdeletion and approximate deletion, and instead align our model directly with retraining without removal data. Approximate deletion is 1. difficult to reason and 2. have questionable alignment guarantee for sufficiently manipulative MLs. Overdeletion has the risk of exposing what is missing, while powerful AIs with situational awareness often aim to complete what is ostensibly missing, including recommendations, making it easy for the resulting model to figure out what was removed.

References

2018 reform of eu data protection rules. URL `https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf`.

James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. Citeseer, 2007.

Avrim Blum, Adam Kalai, and John Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 203–208, 1999.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

Andrew Burt. How will the gdpr impact machine learning?, May 2018. URL `https://www.oreilly.com/radar/how-will-the-gdpr-impact-machine-learning/`.

Joseph A Calandrino, Ann Kilzer, Arvind Narayanan, Edward W Felten, and Vitaly Shmatikov. " you might also like:" privacy risks of collaborative filtering. In *2011 IEEE symposium on security and privacy*, pages 231–246. IEEE, 2011.

Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. *ArXiv e-prints*, 1802.08232, 2018. URL `https://arxiv.org/abs/1802.08232`.

Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 224–232, 2018.

Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. Recommendation unlearning. *arXiv preprint arXiv:2201.06820*, 2022.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 896–911, 2021.

Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International Conference on Machine Learning*, pages 1964–1974. PMLR, 2021.

Whitfield Diffie and Martin E Hellman. Privacy and authentication: An introduction to cryptography. *Proceedings of the IEEE*, 67(3):397–427, 1979.

Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13 (1):3475–3506, 2012.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380, 2009.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/cb79f8fa58b91d3af6c9c991f63962d3-Paper.pdf`.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312, 2020.

Christopher Grau. " eternal sunshine of the spotless mind" and the morality of memory. *The Journal of Aesthetics and Art Criticism*, 64(1):119–133, 2006.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*, pages 263–272. Ieee, 2008.

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*, pages 2008–2016. PMLR, 2021.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674, 2013.

Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881*, 2020.

Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli. Degenerate feedback loops in recommender systems. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 383–390, 2019.

Michael Kearns. A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split. *Advances in neural information processing systems*, 8, 1995.

Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81(2009): 1–10, 2009.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

California Legislature. California legislative information. *Title 1.81. 5 California Consumer Privacy Act of 2018*, 2018.

Yuyuan Li, Xiaolin Zheng, Chaochao Chen, and Junlin Liu. Making recommender systems forget: Learning and unlearning for erasable recommendation. *arXiv preprint arXiv:2203.11491*, 2022.

Ziqi Liu, Yu-Xiang Wang, and Alexander Smola. Fast differentially private matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 171–178, 2015.

Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 521–534. IEEE, 2020.

Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 2145–2148, 2020.

Michael Massimi and Ronald M Baecker. A death in the family: opportunities for designing technologies for the bereaved. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1821–1830, 2010.

Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627–636, 2009.

Mehryar Mohri and Ameet Talwalkar. Can matrix coherence be efficiently and accurately estimated? In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 534–542. JMLR Workshop and Conference Proceedings, 2011.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato, editors, *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pages 931–962. PMLR, 16–19 Mar 2021. URL `https://proceedings.mlr.press/v132/neel21a.html`.

Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Julia Powles and Enrique Chaparro. How google determined our right to be forgotten. 2015.

Shadi Rahimian, Tribhuvanesh Orekondy, and Mario Fritz. Sampling attacks: Amplification of membership inference attacks by repeated queries. *arXiv preprint arXiv:2009.00395*, 2020.

Md Atiqur Rahman, Tanzila Rahman, Robert Laganière, Noman Mohammed, and Yang Wang. Membership inference attack against differentially private deep learning model. *Trans. Data Priv.*, 11(1):61–79, 2018.

Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(12), 2011.

Jeffrey Rosen. The right to be forgotten. *Stan. L. Rev. Online*, 64:88, 2011.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34, 2021.

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.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

Gábor Takács, István Pilászy, and Domonkos Tikk. Applications of the conjugate gradient method for implicit feedback collaborative filtering. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 297–300, 2011.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. *arXiv preprint arXiv:2110.11891*, 2021.

Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 2019.

U.S. House. 117th Congress, Nov 4 2021. URL `https://www.congress.gov/bill/117th-congress/senate-bill/3195`.

André Uschmajew. Local convergence of the alternating least squares algorithm for canonical tensor approximation. *SIAM Journal on Matrix Analysis and Applications*, 33(2):639–652, 2012.

Eduard Fosch Villaronga, Peter Kieseberg, and Tiffany Li. Humans forget, machines remember: Artificial intelligence and the right to be forgotten. *Computer Law & Security Review*, 34(2): 304–313, 2018.

Ari Ezra Waldman. Cognitive biases, dark patterns, and the 'privacy paradox'. *Current opinion in psychology*, 31:105–109, 2020.

MA Woodbury. Inverting modified matrices (memorandum rept., 42, statistical research group), 1950.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, and Reza Shokri. Enhanced membership inference attacks against machine learning models. *arXiv preprint arXiv:2111.09679*, 2021.

A Proof: Untrain-ALS and Retraining Share The Same Minimal Loss, Functionally.

Recall that in Alternating Least Squares (ALS), the loss function is the regularized matrix completion:

$$L_{\mathrm{ALS}}(\mathcal{D}_{\mathrm{obs}}) = \sum_{u,i \in \mathcal{D}_{\mathrm{obs}}} c_{ui}(p_{ui} - x_u^\intercal y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2) \tag{4}$$

For any set of preference matrix P , deterministic function f_c, let the removed dataset be $\mathcal{D}_{\mathrm{rm}}$. As we only remove explicitly observed data points, it is assumed that $\mathcal{D}_{\mathrm{rm}} \subset \mathcal{D}_{\mathrm{obs}}$. When we retrain, we substitute $\mathcal{D}_{\mathrm{remain}} = \mathcal{D}_{\mathrm{obs}} - \mathcal{D}_{\mathrm{rm}}$ for $\mathcal{D}_{\mathrm{obs}}$, and write the loss under retraining as

$$L_{\mathrm{ALS}}(\mathcal{D}_{\mathrm{remain}}) = \sum_{u,i \in \mathcal{D}_{\mathrm{remain}}} c_{ui}(p_{ui} - x_u^\intercal y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2) \tag{5}$$

When we untrain with Untrain-ALS, we set the confidence values manually to 0 for the indices in the removal set. We thus have

$$L_{\mathrm{UntrainALS}}(\mathcal{D}_{\mathrm{obs}}, \mathcal{D}_{\mathrm{rm}}) = \sum_{u,i \in \mathcal{D}_{\mathrm{obs}}} f_c^{\mathrm{untrain}}(c_{ui})(p_{ui} - x_u^\intercal y_i)^2 + \lambda(\sum_u ||x_u||^2 + \sum_i ||y_i||^2) \tag{6}$$

where $f_c^{\mathrm{untrain}}(\cdot)$ transforms the confidence score. Using Kronecker delta δ for set membership, we have

$$f_c^{\mathrm{untrain}}(c_{ui}) = \delta_{(u,i)\in(\mathcal{D}_{\mathrm{obs}} \setminus \mathcal{D}_{\mathrm{rm}})} c_{ui} = (1 - \delta_{(u,i)\in\mathcal{D}_{\mathrm{rm}}}) c_{ui} = c_{ui} - \delta_{(u,i)\in\mathcal{D}_{\mathrm{rm}}} c_{ui}.$$

Assuming the same removal and observations, we hereby call the two loss quantities on $\{\mathcal{D}_{\mathrm{remain}}, \mathcal{D}_{\mathrm{obs}}, \mathcal{D}_{\mathrm{removal}}\}$ in Equation 5 RETRAIN_LOSS and in Equation 6 UNTRAIN_LOSS. We write $\mathcal{D}_{\mathrm{removal}}$ and $\mathcal{D}_{\mathrm{rm}}$ interchangeably.

Our manual zeroing results in

$$\begin{aligned}
\text{UNTRAIN_LOSS} = \quad & \lambda(\sum_{u \in \mathcal{D}_{\mathrm{obs}}} ||x_u||^2 + \sum_i ||y_i||^2) + \sum_{(u,i)\in\mathcal{D}_{\mathrm{obs}}\setminus\mathcal{D}_{\mathrm{rm}}} f_c^{\mathrm{untrain}}(c_{ui})(p_{ui} - x_u^\intercal y_i)^2 \\
& + \sum_{u,i\in\mathcal{D}_{\mathrm{rm}}} f_c^{\mathrm{untrain}}(c_{ui})(p_{ui} - x_u^\intercal y_i)^2 \\
= \quad & \lambda(\sum_{u \in \mathcal{D}_{\mathrm{obs}}} ||x_u||^2 + \sum_i ||y_i||^2) + \sum_{(u,i)\in\mathcal{D}_{\mathrm{remain}}} c_{ui}(p_{ui} - x_u^\intercal y_i)^2 \\
& + \sum_{u,i\in\mathcal{D}_{\mathrm{rm}}} (0)(p_{ui} - x_u^\intercal y_i)^2 \\
= \quad & \lambda(\sum_{u \in \mathcal{D}_{\mathrm{obs}}} ||x_u||^2 + \sum_i ||y_i||^2) + \sum_{(u,i)\in\mathcal{D}_{\mathrm{remain}}} c_{ui}(p_{ui} - x_u^\intercal y_i)^2 \\
= \quad & \text{RETRAIN_LOSS.}
\end{aligned}$$

Our objective thus makes our unlearning method *exact* rather than approximate.

Remark 2 *Whether that minimal loss is achieved, and whether the solutions at minimal loss are necessarily equivalent (or up to a rotation) are not guaranteed from this analysis.*

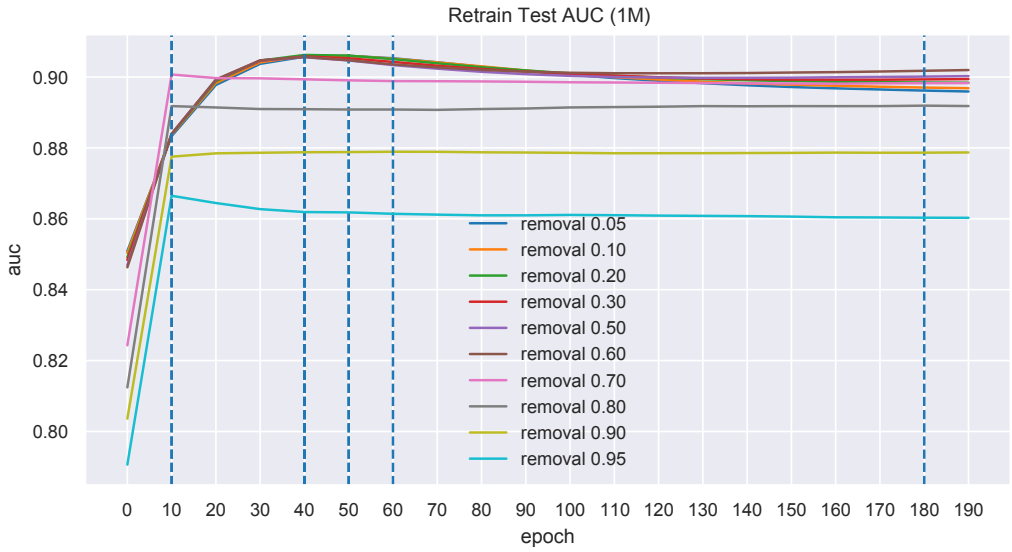

Figure 1: **Re-training dynamics.** Across different iterations of re-training from scratch, and 10 different fractions of removal, each final model's area-under-curve on test set on MovieLens-1M.

B Experimental Parameters

Alternating least squares. P is a preference matrix, which could be binarized values of 1 (like) or 0 (dislike). A confidence score $c_{ui} = f_c(p_{ui})$, where f_c is deterministic. In our experiments, $c_{ui} = 1$ if $p_{ui} = 1$, and 0 or very small otherwise. In the paper it is assumed that $f_c(p_{ui}) = 1 + \alpha p_{ui}$ with a suggested $\alpha = 40$. Each experiment starts with new seed, including train-test split and ALS initializations, unless otherwise mentioned. Graphs are made with 5 runs.

The removal set $\mathcal{D}_{\text{removal}}$ is assumed to be uniformly sampled from $\mathcal{D}_{\text{obs}}$ without replacement.

The number of ALS passes ("epoch" or "iter") is the only tunable parameter for fitting base models. We assume a 99-1 split of train-test, and select epochs based on the best fit validated AUC.

Membership inference. The numbers of iterations for the base models are chosen for validated best model fit, as to be expected for practical deployments.

We use the 50-50 split for test-train on removal and remaining datasets for each appropriate removal fraction, meaning that 50% of the removal data is used in training while the rest is used to validate. The best AUC is taken on the removal data for reporting each model's membership attack accuracy.

C Base Model

Figure 1 shows that even with large removal fractions the base model can still perform well. The dotted vertical lines mark the number of iterations that achieve the best fit for each model. As shown, the less the remaining data, generally the earlier the convergence. 10 passes are sufficient only if removal fraction is large ($> 70\%$). For small fractions of removal, the best fit tends to be between $[40, 70]$ passes on MovieLens-1M. In comparison, Untrain-ALS only takes $[10, 45]$ iterations.

D Empirical Results

To investigate the practical implication of using Unlearn-ALS we conduct experiments that aim to answer two research questions:

1. The performance of Untrain-ALS in terms of accuracy, to prevent model degradation.
2. The runtime of retraining; in our case, that means having fewer iterations than retraining.

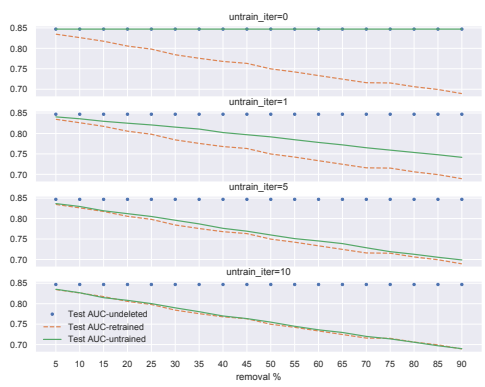

Figure 2: **Unlearning with Untrain-ALS.** Across different iterations of unlearning, and 20 different fractions of removal, each final model's area-under-curve on test set on MovieLens-100K. Retraining is compared to at 25 passes of ALS.

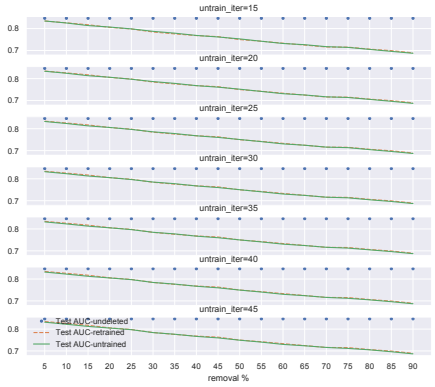

Figure 3: **Overfitting Untrain-ALS.** Across different iterations of unlearning, and 20 different fractions of removal, each final model's area-under-curve on test set on MovieLens-100K. Retraining is compared to at 25 passes of ALS.

3. Unlearn-ALS should reduce the privacy implications from undeleted model.

We note that the empirical privacy evaluation should reliably uncover vulnerabilities of undeleted models, and should be able to differentiate between retrained model, which does not include the offending data, and the undeleted model.

D.1 Experiment Setup

For datasets, we use MovieLens [Bennett et al., 2007, Harper and Konstan, 2015]. For membership inference, the sensitivity issue is severe on larger models, therefore we illustrate with smaller datasets. Unless otherwise specified, we use parallel implementations for Alternating Least Squares, with conjugate gradient speedup but without our inverse adjustments [Hu et al., 2008, Takács et al., 2011]; therefore only compare the number of iterations between untraining and retraining. The python implementation is provided at The removal dataset is held out as a fraction of total observations. All datasets have a test-train split before data removal, so the heldout set is sampled first. The removal fraction ($\%$) refers to the fraction of explicitly observed entries. For training and evaluating recommendation models themselves, we use area-under-curve (AUC), which is more accepted than downstream recommendations for a specific configuration. We show training and untraining across iterations around convergence, and evaluate random removal fractions at every 5%.

Two baselines for Untrain-ALS: 1. the undeleted model, which is trained to completion and used to initialize unlearning, representing the upperbound of model performance. 2. the re-trained from scratch model without the removal data, representing the upperbound of privacy. For model performance, we use area under curve. For membership attacks, we use vulnerability measures derived from membership inference accuracies [Shokri et al., 2017].

D.2 Untrain-ALS: No Degradation, Fast Convergence

As expected from theoretic analysis in Section 4.1, Untrain-ALS is consistent with re-training without removal data in objective.

Over a wide range of removal fractions, Figure 2 shows that untraining is fast, and results in highly performant models. In fact, because Unlearn-ALS clearly follows the well-tested ALS, if untraining is left unchecked, as in Figure 3, there is no degradation to the model compared to training from scratch. Unlearn-ALS breaks the usual expectation that fast unlearning necessarily degrades model.

D.3 Sensitivity issue with membership inference

We investigate empirical attacks based on membership inference against the unlearned model. As alluded to in Section 3, matrix completion-based models do not have a lot of privacy risks to begin with; in implicit feedback datasets, the risks against random data deletion can also be mitigated through extensive model validation. Meanwhile, membership inference attacks [Shokri et al., 2017] are especially powerful when the input data has a lot of information; when in matrix completion, without some advanced user fingerprinting, the model output itself is all the information. Concretely, several challenges arise in this pursuit:

1. AUC of the base model is high. As we start with pre-trained recommenders, it is reasonable to assume that the initial model performs well on a test set. Consider that uniformly removing data is akin to sampling another held-out set, the initial model likely predicts the missing items just as well.

   To make matters worse, ALS performs well even after a large portion of data is deleted.
2. Using only the predicted value, untraining does not observe significant difference between the training data and the removed data, so there is no significant membership inference performance drop. (Only at certain ratios for certain epochs can we induce a 2% difference.) This means the measurement is highly susceptible to small noise.
3. Depending on data splits, the base model (the "undeleted" model) has different membership attack vulnerabilities built-in. This is due to ALS not having a fixed unique solution, so the models from different training trajectories will find different decompositions as solutions to the same matrix completion problem. Some of those models are inherently more defensible than others. This adds noise to the already small numerical measurement.

Recall that our privacy model views re-training as ground truth. To study the vulnerability of unlearning is to study the *additional* vulnerability compared with re-training.

Let $\mathcal{IV}$ denote the *intrinsic vulnerability* associated with the learning strategy. We are concerned with whether Untrain-ALS presents more or less intrinsic risk compared with retraining. Assuming that the training and un-training procedures have similar intrinsic vulnerability, $\mathcal{IV}_{\mathrm{ALS}} \approx \mathcal{IV}_{\mathrm{re-train}}$. An estimator for $\mathcal{IV}_{\mathrm{Untrain-ALS}}$ is thus the difference between the empirical measure for membership inference:

$$\mathcal{IV}_{\mathrm{Untrain-ALS}} = \mathrm{MI}(\mathrm{untrain}) - \mathrm{MI}(\mathrm{retrain}) \tag{7}$$

Remark 3 *Because retraining is assumed to be statistically independent from removed data, being able to infer properties of the removed data from the re-trained model e.g. due to data duplication is not an essential vulnerability. If an empirical measurement shows that untrained model has membership vulnerability, it is a tolerable amount of privacy risk under our setup.*

However, this measurement on intrinsic untraining vulnerability shows that, at the best fit, untraining and untraining are extremely close. This numerical difference is so small, that the measurement appears dominated by noise, while having inconclusive results, as shown in Figure **??**. When averaged across runs, the overlap of untraining and retraining are further obscured.

D.4 Modified Membership Inference

Identifying model noise as a cause, let $\mathcal{IV}'$ be our modified intrinsic vulnerability measure, applied not only to the same $\{M, \mathcal{D}_{\mathrm{obs}}, \mathcal{D}_{\mathrm{rm}}\}$, but also under identical train-test split. The splits greatly impact the model, as we see that intrinsic vulnerability to deletion is closely related to model AUC. Using ALS and Untrain-ALS to retrain and unlearn after data removal, we make three accuracy measurements: $\mathrm{MI}(\mathrm{untrain})$, $\mathrm{MI}(\mathrm{retrain})$, and $\mathrm{MI}(\mathrm{undeleted})$. Even though our privacy model does not directly concern the base model, $\mathrm{MI}(\mathrm{undeleted})$ serves to denoise the influence of model splits on our numerical accuracy differences. We have

$$\mathcal{IV}'_{\mathrm{Untrain-ALS}} = \mathrm{MI}(\mathrm{untrain}) - \mathrm{MI}(\mathrm{retrain}) - \mathrm{MI}(\mathrm{undeleted}) \tag{8}$$

For the same model, Equation 8 appears off by a constant from Equation 7. However, as a measurement, the subtraction for each run improves numerical stability, and reduces noise when averaged over

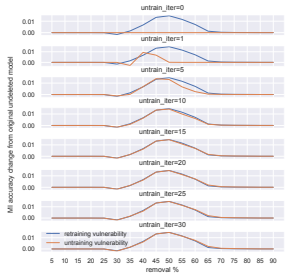

Figure 4: Vulnerability $\mathcal{IV}'$ due to data removal for different re-training iterations and removal fractions, compared against 25 passes of re-training.

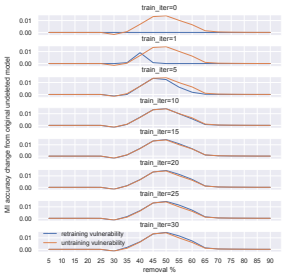

Figure 5: Vulnerability $\mathcal{IV}'$ due to data removal for different re-training iterations and removal fractions, compared against 10 passes of untraining.

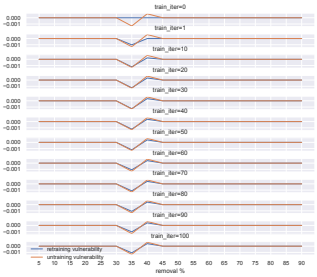

Figure 6: **Large-scale $\mathcal{IV}'$.** Modified vulnerability measure on iterations and removal fractions on MovieLens-1M, with fixed 45 untraining passes.

multiple runs. In Figure 4 and 5, the vulnerability is measured as membership inference accuracy subtracting membership inference accuracy associated with the undeleted model, for the same split under MovieLens-100K. The removal fraction is set at every 5% of the data (even though we are empirically only concerned with small fractions). The procedures for untraining involves training the base model with the selected number of iterations.

Improvements. The sensitivity of the refined metric is much improved, as we now see a change with respect to model training iterations. In Figure 4, as training iterations get larger, the inherent vulnerability is greater. In Figure 5, as untraining continues, there is a decrease of vulnerability. Both phenomena were previously not shown due to sensitivity.

Issues. Nonetheless, our efforts to denoise only has a clear effect on small scale on a specific removal range. The range related to user-requested deletion is, however, still not very sensitive.

E Inherent Privacy (Extended)

Is there a scenario where the recommendation model is "robust" to random user deletion, thus requiring no additional work to unlearn a small subset of training data? Intuitively, dimensionality reduction should result in models with low capacity to memorize. Arguably, high empirical performance also relates to inductive biases: if datasets are well-described by the learned low rank parameters (that exhibit good generalization), it should imply that the model's inductive bias is not to memorize. We make concrete these intuitions in the context of matrix completion for user-movie preferences.

For the following proof sketches, we assume that data removals are independently selected i.e $\mathcal{D}_{\text{removal}}$ to be sampled from $\mathcal{D}_{\text{obs}}$ randomly without replacement.

Validation implies robustness to missing data. First, a key observation: in implicit feedback datasets, each unobserved (and deleted) user-movie interaction is changed to 0. The empirical validation of the model relies on a train-test split that follows the same zeroing convention.

As the mechanism for selecting missing feedback is equivalent to selecting a *held-out* set, any argument for in-domain generalization from appropriate calibration by Kearns [1995] and Blum et al. [1999] would imply low prediction losses on missing data for both retrained and undeleted models.

Recall that membership inference needs to succeed by discriminating the predictions from removal data and the remaining data. Varying data splits, a well-calibrated model has similar expected losses. Because optimizing area-under-curve (AUC) is used for both 1. thresholding membership inference model on the removal data and 2. on remaining validation data on the base retrained recommendation model, we have $P_{\text{retrained}}(p_{ui} = 1|(u,i) \sim \mathcal{D}_{\text{rm}}) \approx P_{\text{retrained}}(p_{ui} = 1|(u,i) \sim \mathcal{D}_{\text{obs}}) = \text{AUC}_{\text{retrain}}$. For each model, the approximation is directly relatable to validation loss.

If the base model is highly accurate i.e. has high AUC, the nonnegative loss contribution from removal data is further limited. Empirically, most recommendation data achieves high AUC even

with large fractions of data removed. As membership inference needs to discriminate two sets of small, non-negative numerical losses of similar means, the task is inherently hard.

Roughly speaking, a well-validated model indeed implies robustness to deletion of a small fraction of data. Implicit feedback models are unique, where cross-validated performance implies an upperbound on the expected removal data's loss contribution, provided that the deletions are independent.

Remark 4 *Though this property makes empirical evaluation for individual privacy harder, it does also mean that the work towards validation and calibration applies directly towards model robustness against deletion. Even though model noise is inevitable in real-world setting, this insight greatly reduces the expectation that there is unknown privacy risk that result from deletion, as all training data is already observed (and presumably pre-selectable for validation).*

Uniqueness of matrix completion implies robustness to missing data. In light of that, we rehash Recht [2011]'s work to recommendation data. We hereby ignore the coherence requirement on the raw data, which is likely untestable in practice (despite fast approximations [Drineas et al., 2012, Mohri and Talwalkar, 2011]); instead, assume the row and column spaces have coherences bounded above by some positive μ_0, as assumptions in Theorem 2 in Recht [2011]. The setup readily applies to our problem. As we assuming that the observations are indeed low-rank, the recovery of the true matrix is certainly robust to small fractions of random deletions.

For preference matrix P of dimension $m \times n$ where $m < n$, assume that underlying ground truth matrix M records the true preferences. Because the preferences are low rank, there is rank r and a singular value decomposition $M = U\Sigma V^*$. As any preference entry is bounded, we trivially obtain the constant value $\mu_1 := {}^{mn}/_{r}$; in practice $\mu_1 \geq 1e5$ or greater for very sparse datasets.

Assuming a threshold probability is chosen, so that the resulting matrix completion UV^* to the problem

$$\begin{aligned} \min_{U,V} \quad & ||X||_{\text{nuclear}} \\ \text{s.t.} \quad & P_{ui} = M_{ui} \quad (u,i) \in \mathcal{D}_{\text{obs}} \end{aligned} \tag{9}$$

is unique and equivalent to M with the given probability when the number of uniformly random observations reaches $\mu_2 mn$ for $\mu_2 \leq 1$. This re-hashes the original result without the explict writeout of the bounds on μ_2 that depends on $\{\mu_0, \mu_1, r, m, n\}$ and the chosen probability threshold.

Let $|\cdot|$ denote set cardinality and let q be the fraction of missing data upon user-requested deletions, so that $|\mathcal{D}_{\text{removal}}| = q|\mathcal{D}_{\text{obs}}|$. Given μ_2, missing data simply subtracts from the number of total observations. When the size of the remaining data, $(1-q)|\mathcal{D}_{\text{obs}}|$, is above $\mu_2 mn$, the recovery is yet unique. That means our missing data does not change the uniqueness condition, if $q \leq 1 - \frac{\mu_2}{|\mathcal{D}_{\text{obs}}|} mn$.

Finally, we want show that for sufficient number of observations, matrix completion solutions are inherently robust. Consdier the retrained and undeleted models. Our results show that they may have the same decomposition under our assumptions, meaning that retraining would not alter the resulting recommendation system in terms of recovered entries i.e. predictions downstream. Their empirical privacy is thus equivalent, meaning the undeleted model is as private as the retrained model.

Remark 5 *Unfortunately, the bound is often vacuous, as the real world data is far sparser than what the theorectics posit i.e. $\mu_2 \propto \mu_1^2$ while μ_1 is too large. Additionally, the minization is often performed using heuristic methods such as alternating least squares, where the uniqueness of the solutions is not guaranteed, even if the underlying un-regularized minimization is unique.*

For practical privacy, the independece assumptions of random independent romoval can not be guaranteed; after all, many users will likely remove the most embarassing content from watch history.

F Membership Inference Metric

For our application context, the natural measure is whether an observer of model outputs can recover or guess what a user once sought to remove.

Divergence-based measures aim to see the downstream difference between untrained and retrained models using a divergence measure $D(P_{\text{retrain}}||P_{\text{retrain}})$, such as KL-divergence [Golatkar et al., 2020]. At evaluation it is hoped that $\forall(u,i) \in \mathcal{D}_{\text{removal}}$,

$$P_{\text{retrain}}(p_{ui} = 0) = P_{\text{retrain}}(p_{ui} = 0). \tag{10}$$

However in collaborative filtering, this objective is under-constrained, as the adversary can observe outputs outside of those in $\mathcal{D}_{\text{removal}}$ which may be impacted through the removal process. Even if those removed data points remain similar in output, an adversary may still see from the remaining data some anomalies. Instead, suppose an eavesdropper who can observe all data that is observed, except for a particular entry $p_{u_0 i_0}$, we have $\forall(u,i) \in \mathcal{D}_{\text{obs}}$,

$$P_{\text{retrain}}((u,i) \sim \mathcal{D}_{\text{removal}}) = P_{\text{retrain}}((u,i) \sim \mathcal{D}_{\text{removal}}). \tag{11}$$

Thus we use membership attack to empirically calibrate both sides, maximizing the probability of attack success for a given model, and then measure the difference between those optimal success rates. For an appropriately forgotten model i.e. complete and not-deleting, the membership attack rate should not increase for the "best guess" for any data removed from the preference matrix.

Two benefits ensue: 1. auto-calibration that is suitable for our threat model, when Equation 10 is uncalibrated, and 2. usability when we only have two models per data split, instead of relying on sampling from a distribution of models.

G Privacy Analysis (Extended Discussion)

G.1 Privacy Context, Threat Model, and the Legality of Data Removal

User privacy is a complex issue deserving of nuanced debate. We hereby outline related concepts.

Privacy in "Netflix and Forget". The data flow in our privacy model originates from the user, while the adversary also includes the user. It deviates from common privacy notions such as preventing information extraction [Diffie and Hellman, 1979], or the Right To Be Forgotten [Rosen, 2011].

However, our privacy motivation is a pragmatic user scenario. While being private from one's own recommendations is not considered "unauthorized", letting other users guess the original data with high likelihood constitutes as unauthorized after the data source is withdrawn.

Even though the legal implements of the right to be forgotten are limited, forgetting past records at user request is a natural form of privacy. While most cases discussed under the rightinvolve public records, Powles and Chaparro [2015] argues that the system through which the information is surfaced is crucial. Though people may prefer personal data removed purely out of emotional reasons, computational systems often treat data with "decontextualized freshness":

> They include prominent reminders that an individual was the victim of rape, assault or other criminal acts; that they were once an incidental witness to tragedy; that those close to them – a partner or child – were murdered. The original sources are often many years or decades old. They are static, unrepresentative reminders of lives past, lacking the dynamic of reality. Powles and Chaparro [2015]

We thus take the right to be forgotten in the spirit of decaying information while giving users the autonomy over their data. When the data is forgotten, we expect the system to behave as though the data was not supplied in the first place. On the other hand, to devise an attack, we use membership attack under the model that an observer of the recommendation system should not be able to tell with high probability whether some information was *removed*.

Threat Model The data owners request random deletion of training data, to which the model owner respond by updating the model. An eavesdropper with access to the model outputs attempts to guess whether a data point had been removed.

Does machine learning need to implement the Right To Be Forgotten? The ability to remove personal digital records is grounded in normative ethics. In dealing with loss of loved ones, common

bereavement guides suggest removing the audio retained from answering machines, as voices, unlike photos, are often recorded incidentally rather than for the sake of remembranace [Massimi and Baecker, 2010]. To move on from grief, a human user ought to have the ability to remove past records that bring them horror and regret, including the records' downstream summaries or re-caps.

However, the current paradigms of the legal system in the United States, where many major techonology companies are based, do not support a comprehensive regulation on privacy specific to machine learning systems.The California Consumer Privacy Act (CCPA) [Legislature, 2018] and the proposed congressional bill Consumer Online Privacy Rights Act (S.3195) [U.S. House. 117th Congress, 2021] forbid businesses from expanding processing of personal data beyond the intended use. They are, however, quite fresh and rarely enacted. Meanwhile, the more mature GDPR supports such removal of past records used in "automated decisions" [EUd]. Nevertheless, Burt [2018] interprets that even though users usually need to consent to their data being used for training, removing it does not necessarily mean the models need to be retrained.

A case may hinge on whether the un-removed model will leak information about the data to remove [Burt, 2018]. While Papernot et al. [2016], Choquette-Choo et al. [2021] have shown that many models being deployed such as large scale language models have concrete privacy risks, such tools of empirical evaluation is not accessible to the general public, especially when they rely on accessing the training process. At best they serve as self-checking tools for companies that decide to provide such feature, but not as a tool that can be incorporated into regulation. The current state of online privacy is thus in a state of disarray: a lot of private data is compromised, which are fed to machine learning models. Meanwhile, few regulations are put in place to deal with the downstream effect, and no publically accessible method to measure such privacy loss.

G.2 Why Aren't Machine Unlearning Solutions Deployed In Machine Learning?

As Waldman [2020] observes, deploying privacy features that match the user's cognitive model is not a priority for technology developers. While many users would likely remove historical records on YouTube or Netflix hoping for changed recommendations, few recommendation systems have transparent guarantees on unlearning user preferences.

Legal recognition is the most ostensible obstacle: only a few privacy bills have been passed in America, where many major technology companies are located. Lacking any aforementioned privacy regulation specifically worded on artificial intelligence, there is little recourse for users who want their data removed from machine learning pipelines.

Industrial-scale computation is one reason lobbists use against passing bills that compel real-time removal of user data. Retraining is considered expensive, thus bad for business. While it may be argued that user privacy holds priority over computation cost and model accuracy, there has yet been a compelling demonstration that industrial-scale recommendation models can be efficiently unlearned without hurting the bottom line. After all, large recommendation models are widely used in multiple downstream products, and are expensive to train and re-train.

Flexible unlearning. Undoubtly, the holy grail of machine unlearning is to allow any model to forget arbitrary training data, as if it were re-trained from scratch. Such a method, which does not depend on a specific learning architecture, would have truly sweeping implications. Generic unlearning applies to a wide range of models, without incurring costly training time modifications, extensive check-pointing, or differential private training. Moreover, with the popularity of finetuning pretrained models for applications, the downstream model servers may not have access to the training procedure or original parameters to begin with. Unlearning without learning enables most trendy services to fortify their systems after performing finetuning.

Additionally, our work uncovers a different dimension of the issue: evaluations. We need a way to know when privacy is lost, and when privacy is perserved.

G.3 Auditable Evaluations

A mature unlearning system would need to have compelling and robust evaluations. We still lack a realistic and auditable alternative to membershio inference [Thudi et al., 2021]. When un-learning

simulates re-training, the ground truth to compare against is clear and reasonable. Platforms and regulators would only need to communicate the following idea: data deletion from machine learning model is analougous to forgetting, acting as if the platform never received such data.

Against privacy risks, a defended model needs to be evaluated against the identified risk. Membership Inference aims to identify memorization of training data by a model, and has gained popularity in succeeding in uncovering privacy risks [Shokri et al., 2017, Rahman et al., 2018, Truex et al., 2019, Choquette-Choo et al., 2021]. Typical membership inference uses a collection of samples that are not in the training data, feed them to the model, and take the outputs as the baseline negative training set. The positive training set is the data that the model has seen in the training set. Other membership inference methods have been developed, usually requiring access to the model or the training procedure or a more focused clean dataset [Long et al., 2020, Rahimian et al., 2020, Ye et al., 2021]. The central idea is to make the empirical attack prediction more salient more powerful adversaries.

Recently, Carlini et al. [2018] took a different approach for large scale language models to test if a data point had been deleted [Carlini et al., 2018, Izzo et al., 2021]. This negative dataset is manufactured "poison" to the training procedure. The intuition is that if the model is prone to memorization, it would be able to reproduce the exact random string that was injected in the training set. The membership inference variant thus focuses on engineering a better dataset, thus making it more effective at uncovering memorization. While powerful in engineering a clear metric, this approach requires the model owner to audit from within.

Our scenario for recommendation privacy turns out especially revealing: common membership inference classification is not able to uncover privacy risk, even though the devised method is not information-theorectically private. Indeed Jayaraman et al. [2020] calls for revisiting membership inference in real life, noting that it is not as powerful as an empirical measure. Chen et al. [2021] points out that unlearning can, in fact, decrease privacy, highlighting the need for better evaluations. We thus join calls with Thudi et al. [2021] in calling for auditable algorithms that evaluate machine unlearning.

H Related Works

Machine unlearning is an emerging field motivated by performance and computation trade-offs to implement the Right to Be Forgetten on machine learning models [Grau, 2006]. When a user seeks to retract data used in training, the derived model ought to update with respect to the change. Unlearning thus trades off computation, accuracy, and privacy, and is often compared with retraining [Neel et al., 2021, Ginart et al., 2019, Bourtoule et al., 2021, Golatkar et al., 2020].

Unlearning recommendation systems is concurrently explored by Li et al. [2022] and Chen et al. [2022], which target unlearning for industrial scale recommendations built through collaborative filtering. Sharding and user clustering are key to their methods, which we do not consider. Instead, our work complements the line of work through a much simpler unlearning algorithm that applies to all bi-linear models with minimal architectural change.

Differentially-private recommendations McSherry and Mironov [2009], Liu et al. [2015] may be naturally compliant towards the Right to Be Forgotten by reducing the risk related to the model output revealing information about the inclusion of certain data. However these methods would need to anticipate to a certain extent the likelihood of deletion, and build that into training.

Evaluations against privacy risks if no privacy risk is shown, it would mean that no computation needs to be expended on unlearning. Membership Inference is a popular method that measures training data memorization by a model. Typical membership inference uses a collection of samples that are not in the training data, feed them to the model, and take the outputs as the baseline negative training set. The positive training set is the data that the model has seen in the training set. Other membership inference methods have been developed, usually requiring access to the model or the training procedure more metrics [Chen et al., 2021]. The central idea is to make the empirical attack model more powerful.

Recently, Carlini et al. [2018] took a different approach. They developed a very effective empirical evaluation would be applicable to any model after it has been trained. For large scale language models, feature injection can test if a data point had been deleted [Izzo et al., 2021]. This negative dataset is manufactured "poison" to the training procedure. The intuition is that if the model is prone to memorization, it would be able to reproduce the exact random string that was injected in the training set. The membership inference variant thus focuses on engineering a better dataset, thus making it more effective at uncovering memorization. While powerful, it requires internal access to model training.

Differential Privacy Similar to a well-behaving matrix completion solution's inherent privacy (Section 3), some models may be less prone to memorizing individual data points. As a result, they are less at risk for membership attacks after deletion requests.

By definition, pure differentially private models are robust to deletion, as each individual data point's membership should not be inferrable [Dwork and Lei, 2009]. Yet, not all models trained with differential privacy are robust. In practice, assumptions on the independences between data points do not hold and the number of deletion requests may not be known ahead of training; additionally, businesses often opt for approximations, since pure differential privacy poses degradation on model utility. As a result, Rahman et al. [2018] finds that models trained to be differentially private are yet vulnerable.

I Discussion

We propose using Untrain-ALS to perform machine unlearning in bi-linear recommendations based on matrix completion, which is simutaneously widely deployed in the real world and under-studied in machine unlearning. This method takes advantage of fast heuristic, and can unlearn exactly without compromising model degradation. However, empirically, models learned with regularized matrix completion are not unique, thus unlearning and re-training may exhibit small differences in privacy. To find them, we employ empirical attacks of memership inference, and adapt the vanilla version to denoise the impact of data splits, and successfully see trends in vulnerability that was previously obscured. We see two trends emerging from empirical results: 1. Unlearn-ALS is clearly fast and powerful, with no degradation in model performance, unlike most unlearning methods [Sekhari et al., 2021]. 2. Unlearn-ALS is not the same as re-training, but it closely relates to re-training in most privacy measures, provided that it is trained to the best fit. 2. Relying on membership inference classifications alone to measure unlearning thus leads to potential outstanding privacy risks. We join prior calls in urging the unlearning community to re-think empirical evaluations for unlearning to meet practical privacy needs[Truex et al., 2019, Chen et al., 2021, Jayaraman et al., 2020].

Limitations Our work is primarily limited to the choice of models. Though we apply to all bi-linear models, not all recommendation systems are implemented with dot products.

Societal impact We place pressure on platforms that train on user data to give users real-time options to remove the influence of their training data. Despite research progress, however, real world systems have yet caught on [Villaronga et al., 2018]. When users opt to remove their past records' influence on recommendations, existing implementations tend to fall under two categories: complete expunging of their recommendation, in which a user's all historic interactions are zero-ed, such as Netflix's reset, or a vague removal of learnt concepts such as Facebook's Ad preferences. While many services offer granular control over which ones of their historic actions the platform collects, they do not promise that the deletion necessarily impact downstream systems that learn from such data.

Ostensibly, two factors prevent machine unlearning to be deployed: 1. lacking legal recognition for the associated privacy risks, as GDPR-style deletion hinges on whether automated systems leak private data for the general public [Villaronga et al., 2018]. For that, our work adds to the rigor of discovery: empirical evaluation needs revisiting. 2. industrial-scale computation expenditure on pre-trained machine learning models is massive, and there has yet been a compelling demonstration that industrial-scale recommendation models can be efficiently unlearned without hurting the bottom line. For this factor, our work on Unlearn-ALS proposes unlearning. Upon sequential deletion requests, the unlearned model will not perform worse than the retrained model. When no trade-off is made, the hope is that both policy and industry can agree to facilitate user privacy.

Malicious use Our selective forgetting techniques can be applied to sinister areas where forgetting is a form of censorship. Even the right to be forgotten faces criticism outside of the legal realm; after all, even the forgetting procedure in "Eternal Sunshine of the Spotless Mind" may be troublesome because it lets the recipient "live a lie" Grau [2006].