Differentially Private Recommender Systems

David Madras

University of Toronto

April 4, 2017

Introduction

- Today I'll be discussing "Differentially Private Recommender Systems", by Frank McSherry and Ilya Mironov in 2009 [1]
- Modern recommendation systems aggregate many user preferences
- This allows for better recommendations
- Can compromise privacy
- Improved privacy can lead to "a virtuous cycle"
- Better privacy $\rightarrow$more user data $\rightarrow$better privacy $\rightarrow$...

Introduction

- Example: Netflix movie recommendation system
- Has database of ratings (1 - 5 stars) of many movies by many users
- Will recommend movies based on past ratings by you and similar users
- Information can be used to link profiles
- Attackers can make inferences about others by injecting own input

Figure 1: Netflix

Contribution of this paper

- Develops "realistic" DP recommender system
- Integrate DP into the calculations, rather than presenting private data
- Proves privacy guarantees
- Tests algorithm performance on Netflix Prize dataset

Related Work

- Survey of DP-analogues of various machine learning algorithms [2]
- Demonstrations of privacy attacks on Netflix (or similar) data
  - Can identify rows based on few data points [3]
  - Can make valid inferences about user history by observing recommendations (Amazon data) [4]
- Data anonymization techniques [5, 6]
  - These tend to destroy performance of recommender algorithms
- Cryptographic solutions [7, 8]
  - Focus on removing central trusted party with complete access

High-level Recommendation Algorithm Framework

- Given: users, items, ratings on a subset of (user, item) pairs
- Want to predict held-out values at (user, item) locations
  1. Global Effects: Centre ratings by subtracting per-user/per-movie averages
     - ★ Augment with artificial ratings at global average to stabilize averages with small support
  2. Find covariance matrix C
  3. Apply geometric recommendation algorithm to C
     - ★ Roughly, we can compute many learning algorithms using the covariance matrix e.g. factor analysis, clustering, etc.
     - ★ If covariance matrix is DP, the whole algorithm will be DP

A DP Recommendation Algorithm - Notation

- Let r_u be user u's ratings vector, and r_{ui} be user u's rating on item i
- Let e_u, e_{ui} be the binary vectors and elements denoting presence of ratings
- Let $c_u = \|e_u\|_1$ be the number of ratings by user u
- $X = x + \mathit{Noise}$ means we're adding some type of DP noise - either Laplacian or Gaussian depending on what guarantee we want to satisfy

A DP Recommendation Algorithm - Item Effects

- First calculate global average G privately

$$G = \frac{GSum}{GCount} = \frac{\sum_{u,i} r_{ui} + Noise}{\sum_{u,i} e_{ui} + Noise} \tag{1}$$

- Then calculate per-item averages $MAvg_i$ privately, stabilizing with β_m fictitious ratings of G for each item

$$MAvg_i = \frac{MSum_i + \beta_m G}{MCount_i + \beta_m} \tag{2}$$

where $MSum_i = \sum_u r_{ui} + Noise$, $MCount_i = \sum_u e_{ui} + Noise$

- These averages are DP and can be published - we can incorporate them into further computation with no additional privacy cost

A DP Recommendation Algorithm - User Effects

- We can subtract these per-item averages, and then centre ratings by user as well
- The per-user average (not DP) $\bar{r}_u$ is calculated as

$$\bar{r}_u = \frac{\sum_i (r_{ui} - MAvg_i) + \beta_p G}{c_u + \beta_p} \tag{3}$$

- Calculate centred $\hat{r}_{ui} = r_{ui} - \bar{r}_u$
- Clamp these to a sensible interval $[-B, B]$ to lower sensitivity of measurements

Effect of a Single Rating Change

- What is the maximum effect of a single rating change on centred and clamped ratings $\hat{r}$?
- Let r^a, r^b be two sets of ratings with a single new rating at r^b_{ui}
- Then the only difference in $\hat{r}^a$ and $\hat{r}^b$ is in $\hat{r}_u$
- For any j where r^a, r^b have common ratings:

$$|\hat{r}^b_{uj} - \hat{r}^a_{uj}| \leq |\bar{r}^b_u - \bar{r}^a_u| = \frac{|r^b_{ui} - \bar{r}^a_u|}{c^b_u + \beta_p} \leq \frac{\alpha}{c^b_u + \beta_p} \tag{4}$$

where α is the maximum possible difference between ratings (for Netflix, $\alpha = 5 - 1 = 4$)

Effect of a Single Rating Change

- $|\hat{r}^b_{uj} - \hat{r}^a_{uj}| \leq \frac{\alpha}{c^b_u + \beta_p}$ is a bound on the difference in a single clamped, centred rating
- Using that $|\hat{r}^b_{ui}| \leq B$, we can bound the difference between the clamped, centred databases as well (they only differ on one row)

$$\begin{aligned} \|\hat{r}^b - \hat{r}^a\|_1 &\leq c^a_u \times \frac{\alpha}{c^b_u + \beta_p} + B < \alpha + B \\ \|\hat{r}^b - \hat{r}^a\|_2^2 &\leq c^a_u \times \frac{\alpha^2}{(c^b_u + \beta_p)^2} + B^2 < \frac{\alpha^2}{4\beta_p^2} + B^2 \end{aligned} \tag{5}$$

- Since $c^a_u + 1 = c^b_u$, we can bound the first squared term from above with $\frac{\alpha^2}{4\beta_p} + B$ by taking derivative w.r.t. c^a_u and maximizing
- As β increases, these differences become arbitrarily close to B, B^2

Calculating the Covariance Matrix - User Weights

- For a single change in rating (in row u), the difference in covariance matrices is bounded by (maybe times a constant)

$$\|Cov^a - Cov^b\| \leq \|r_u^a\| + \|r_u^b\| \tag{6}$$

- For users with many ratings, this can be very high
- We introduce weights $w_u = \frac{1}{\|e_u\|}$ for each user, to normalize the contributions of each user
- These weights will be used to calculate the covariance matrix

Calculating the Covariance Matrix

- We want to find good low dimensional subspaces of the data - three similar approaches:
  1. Apply SVD to the data matrix
  2. Apply SVD to the items x items covariance matrix
  3. Apply SVD to the user x user Gram matrix
- Adding noise for privacy makes some of these approaches inconvenient
  1. Data matrix: error scales with # users
  2. Item cov. matrix: error scales with # items
  3. User Gram matrix: error scales with # users, # items, max covariance between two users
- For most applications, item covariance matrix is best
- To calculate the covariance matrix C of movies in a DP way

$$C = \sum_u w_u \hat{r}_u \hat{r}_u^T + Noise \tag{7}$$

Calculating the Covariance Matrix

- We want to show that given a change in a single rating, this covariance matrix will not change too much
- Again, we'll take r^a, r^b be two sets of ratings with a single new rating at r^b_{ui}
- How big can $\|C^a - C^b\|$ be?
- First, note that since the ratings r only differ on one row, $\|C^a - C^b\| = \|w^a_u \hat{r}^a_u \hat{r}^{aT}_u - w^b_u \hat{r}^b_u \hat{r}^{bT}_u\| = \|w^a_u \hat{r}^a_u (\hat{r}^a_u - \hat{r}^b_u)^T\| + \|w^b_u (\hat{r}^{aT}_u - \hat{r}^{bT}_u) \hat{r}^{bT}_u\| + \|(w^a_u - w^b_u) \hat{r}^a_u \hat{r}^{bT}_u\|$
- Since $\|e^a_u\| - \|e^b_u\| \leq 1$, $w^a_u - w^b_u = \frac{1}{\|e^a_u\|} - \frac{1}{\|e^b_u\|} \leq \frac{1}{\|e^a_u\|\|e^b_u\|}$, we can also say that:

$$\|C^a - C^b\| \leq (\frac{\hat{r}^a_u}{\hat{e}^a_u} + \frac{\hat{r}^b_u}{\hat{e}^b_u})\|\hat{r}^a_u - \hat{r}^b_u\| + \frac{\|\hat{r}^a_u\|\|\hat{r}^b_u\|}{\|e^a_u\|\|e^b_u\|} \tag{8}$$

Calculating the Covariance Matrix

- Using $\|\hat{r}_i\| \leq \|\hat{e}_i\| \times B$ and the previous bounds on $\|\hat{r}_u^a - \hat{r}_u^b\|$:

$$\begin{aligned} \|C^a - C^b\|_1 &\leq (B + B)(\alpha + B) + B^2 = 2B\alpha + 3B^2 \\ \|C^a - C^b\|_2 &\leq (B + B)(\sqrt{\frac{\alpha^2}{4\beta_p} + B^2}) + B^2 \\ &= 2B(\sqrt{2B^2}) + B^2 = B^2(1 + 2\sqrt{2}) \end{aligned} \tag{9}$$

where we use $\beta_p = \frac{\alpha^2}{4B^2}$

Calculating the Covariance Weight Matrix

- A similar result holds for the binary e matrix (which indicates which ratings are present)

$$\begin{aligned} \|w_u^a \hat{e}_u^a \hat{e}_u^{aT} - w_u^b \hat{e}_u^b \hat{e}_u^{bT}\|_1 &\leq 3 \\ \|w_u^a \hat{e}_u^a \hat{e}_u^{aT} - w_u^b \hat{e}_u^b \hat{e}_u^{bT}\|_2 &\leq \sqrt{2} \end{aligned} \tag{10}$$

Per-User Privacy

- The claims in this paper are with respect to per-rating privacy
- A stronger guarantee would mask the presence of an entire user
- The only change we need to make is to apply a "more aggressive down-waiting by number of ratings"
- So our ratings vectors are normalized before we do any of the counting operations
- This claim is not entirely clear to me

Cleaning the Covariance Matrix

- Optionally, we can denoise the covariance matrix a little for better performance
- "Shrinking to the average"

$$\bar{C}_{ij} = \frac{C_{ij} + \beta mean(C)}{W_{ij} + \beta mean(W)} \tag{11}$$

- Conduct a rank-k approximation
- The low-rank approximation also compresses it - easier to send to client computers
- Post-processing does not affect privacy

Evaluation

- Netflix Prize dataset: 100M ratings, 17770 movies, 480K people
- Use (ϵ, δ)-DP, parametrizing to one parameter θ
- For each measurement f_i, the magnitude of noise will be

$$\sigma_i = \max_{A \approx B} \frac{\|f_i(A) - f_i(B)\|}{\theta_i} \tag{12}$$

- We will set each θ_i as $\frac{\theta}{K}$ - we can vary θ as our one parameter
- With Laplace noise, this gives us ϵ_i-DP for $\epsilon_i = \theta_i$ on measurement f_i
- With Gaussian noise, we will have (ϵ_i, δ_i)-DP for $\epsilon_i = \theta_i \sqrt{2 \log(\frac{2}{\delta_i})}$
- By composition, our final guarantees will be $\epsilon = \theta$ or $\epsilon = \theta \sqrt{2 \log(\frac{2}{\delta})}$ if we choose a common δ value

Evaluation

- The algorithm measures the data 3 times: global average, per-item average, covariance matrix
- The authors set a different θ_i for each, scaling the global θ by 0.02, 0.19, 0.79 respectively
- The global average receives so much noise because it is contributed to by many ratings, and therefore is very stable
- Apply both kNN and SVD prediction algorithms with ridge regression
- Various parameter settings: $\beta_m = 15, \beta_p = 20, B = 1$
- Evaluated by root mean squared error (RMSE) on a held-out test set

The Big Results Slide

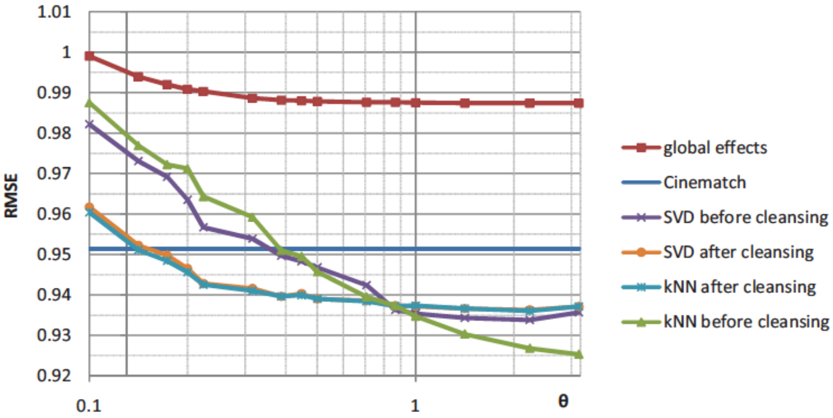


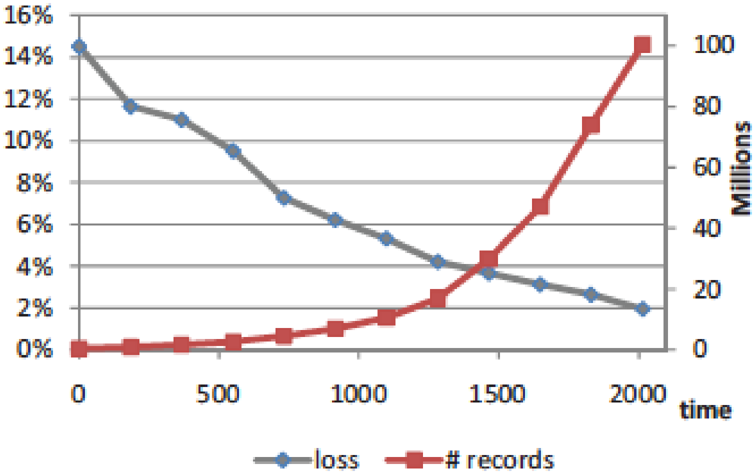
Figure 2: RMSE on prediction for different privacy levels

Results

- As noise (and privacy) increases, accuracy decreases
- Both algorithms cross the Cinematch threshold at $\theta \approx 0.15$
- Covariance matrix cleansing makes the algorithms more accurate without compromising privacy
- It helps most in the high noise domain
  - Could be a consequence of the fact that hyperparameters were optimized for $\theta = 0.15$

Results Over Time

- Also experimented with different dataset sizes - n day window starting from 2000, $n \leq 2000$
- More data helps accuracy (figure is for $\theta = 0.15$)



References

- F. McSherry and I. Mironov, “Differentially Private Recommender Systems: Building Privacy into the Net,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’09, (New York, NY, USA), pp. 627–636, ACM, 2009.
- C. Dwork, “An Ad Omnia Approach to Defining and Achieving Private Data Analysis,” in *Proceedings of the 1st ACM SIGKDD International Conference on Privacy, Security, and Trust in KDD*, PinKDD’07, (Berlin, Heidelberg), pp. 1–13, Springer-Verlag, 2008.
- A. Narayanan and V. Shmatikov, “Robust De-anonymization of Large Sparse Datasets,” in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP ’08, (Washington, DC, USA), pp. 111–125, IEEE Computer Society, 2008.
- J. A. Calandrino, A. Narayanan, E. W. Felten, and V. Shmatikov, “Don’t review that book: Privacy risks of collaborative filtering,” 2009.