18734: Foundations of Privacy

Differentially Private Recommendation Systems

Anupam Datta

Fall 2015

Netflix $1,000,000 Prize Competition

User/Movie	….	300	The Notebook	….
…	…	…	…	…
John		4	Unrated	
Mary		Unrated	Unrated	
Sue		2	5	
Joe		5	1	
…	…	…	…	…

Queries: On a scale of 1 to 5 how would John rate "The Notebook" if he watched it?

Netflix Prize Competition

User/Movie	….	13,537	13,538	….
…	…	…	…	…
258,964		(4, 10/11/2005)	Unrated	
258,965		Unrated	Unrated	
258,966		(2, 6/16/2005)	(5, 6/18/2005)	
258,967		(5, 9/15/2005)	(1,4/28/2005)	
…	…	…	…	…

Note: N x M table is very sparse (M = 17,770 movies, N = 500,000 users)

To Protect Privacy:

- Each user was randomly assigned to a globally unique ID
- Only 1/10 of the ratings were published
- The ratings that were published were perturbed a little bit

Root Mean Square Error

$$RMSE(P) = \sqrt{\frac{\sum_{i=1}^{k}(p_i - a_i)^2}{k}}$$

$p_i \in [1,5]$ - predicted ratings

$a_i \in [1,5]$ - actual ratings

Netflix Prize Competition

Goal: Make accurate predictions as measured by Root Mean Squared Error (RMSE)

$$RMSE(\vec{P}) = \sqrt{\frac{\sum_{i=1}^{k}(p_i - a_i)^2}{k}}$$

$p_i \in [1,5]$ - predicted ratings

$a_i \in [1,5]$ - actual ratings

Algorithm	RMSE
BellKor's Pragmatic Chaos	0.8567 < 0.8572
Challenge: 10% Improvement	0.8572
Netflix's Cinematch (Baseline)	0.9525

Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 leaders.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8642	9.27	2009-07-15 14:53:22
14	Gravity	0.8643	9.26	2009-04-22 18:31:32
15	Ces	0.8651	9.18	2009-06-21 19:24:53
16	Invisible Ideas	0.8653	9.15	2009-07-15 15:53:04
17	Just a guy in a garage	0.8662	9.06	2009-05-24 10:02:54
18	J Dennis Su	0.8666	9.02	2009-03-07 17:16:17
19	Craig Carmichael	0.8666	9.02	2009-07-25 16:00:54
20	acmehill	0.8668	9.00	2009-03-21 16:20:50
Progress Prize 2007 - RMSE = 0.8723 - Winning Team: KorBell				
Cinematch score - RMSE = 0.9525				

Netflix Privacy Woes

3/12/2010 @ 12:35PM | 2,590 views

Netflix Settles Privacy Lawsuit, Cancels Prize Sequel

Taylor Buley , Contributor

+ Comment Now + Follow Comments

On Friday, Netflix announced on its corporate blog that it has settled a lawsuit related to its Netflix Prize, a $1 million contest that challenged machine learning experts to use Netflix's data to produce better recommendations than the movie giant could serve up themselves.

The lawsuit called attention to academic research that suggests that Netflix indirectly exposed the movie preferences of its users by publishing anonymized customer data. In the suit, plaintiff Paul Navarro and others sought an injunction preventing Netflix from going through the so-called "Netflix Prize II," a follow-up challenge that Netflix promised would offer up even more personal data such as genders and zipcodes.

Outline

- Recap: *Differential Privacy* and define *Approximate Differential Privacy*
- Prediction Algorithms
- Privacy Preserving Prediction Algorithms
- Remaining Issues

Privacy in Recommender Systems

- Netflix might base its recommendation to me on both:
  - My own rating history
  - The rating history of other users
- Goal: not leak other users' ratings to me
- Basic recommendation systems leak other users' information
  - Calandrino, et al. Don't review that book: Privacy risks of collaborative filtering, 2009.

Recall Differential Privacy [Dwork et al 2006]

Randomized sanitization function κ has ε-differential privacy if for all data sets *D1* and *D2* differing by at most one element and all subsets *S* of the range of κ,

$$\Pr[\kappa(D1) \in S] \leq e^{\varepsilon} \Pr[\kappa(D2) \in S]$$

Review: Laplacian Mechanism

$K(D) = f(D) + \mathrm{Lap}(GSf/\varepsilon)$

Thm: K

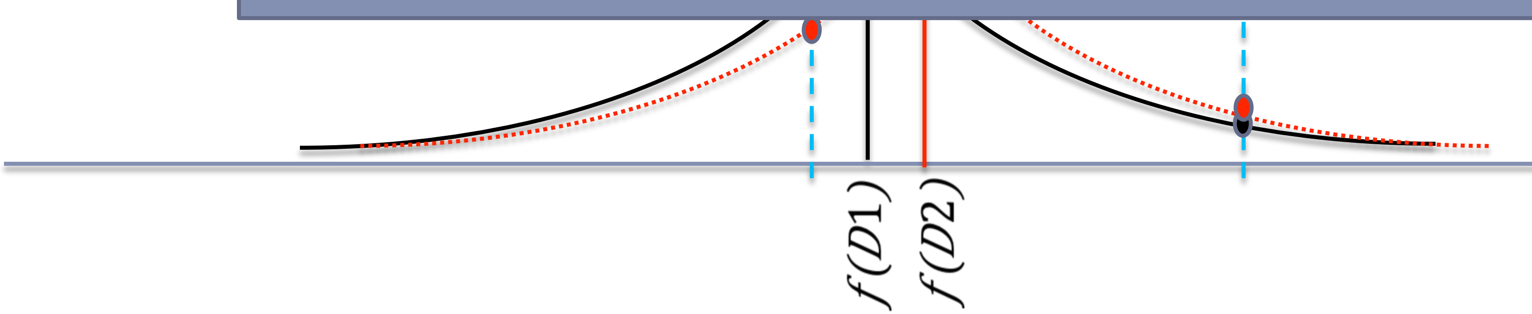
Probability Density Function

$x|/\sigma)$

Picture P

$e^{-\varepsilon}$

Question: The Gaussian (Normal) distribution is nicer because it is more tightly concentrated around its mean. Can we use that distribution instead?



Gaussian Mechanism

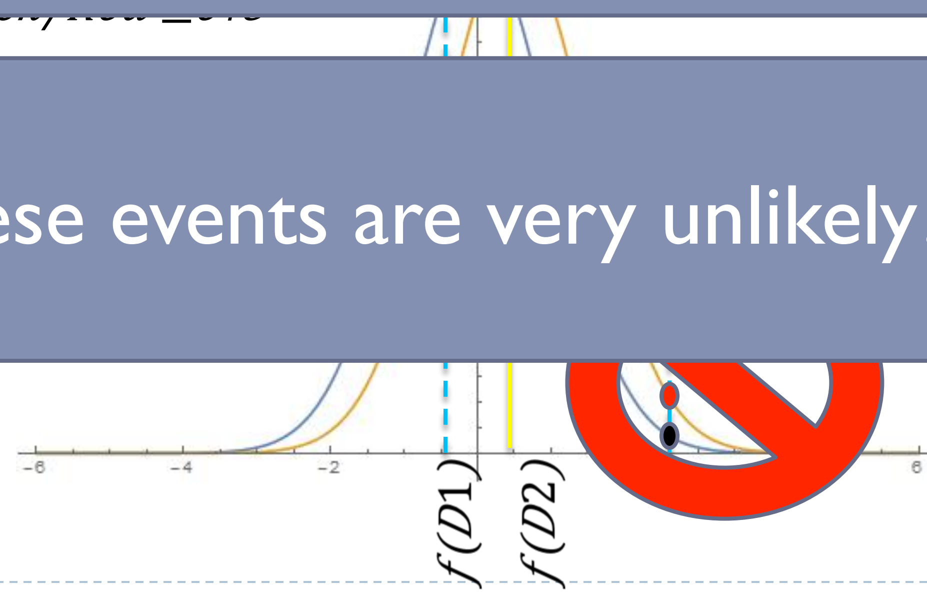
$\kappa(D) = f(D) + N(GS_f/\varepsilon)$

Thm? K is ε-differentially private?

Probability Density Function

$N(x, 0, \sigma) \propto \frac{1}{\sigma\sqrt{2\pi}} \exp(-x^2/2\sigma^2)$

Problem: The ratio can be huge at the tails!



But these events are very unlikely…

Approximate Differential Privacy

Randomized sanitization function κ has *(ε, δ)-differential privacy* if for all data sets *D1* and *D2* differing by at most one element and all subsets *S* of the range of κ,

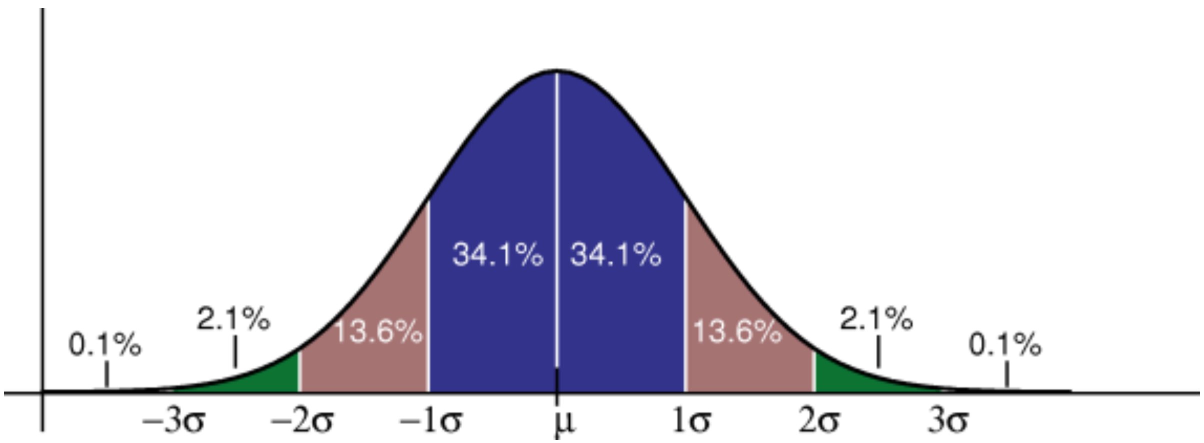
$$\Pr[\kappa(D1) \in S] \leq e^{\varepsilon} \Pr[\kappa(D2) \in S] + \delta$$

Gaussian Mechanism

$K(D) = f(D) + N(\sigma 2)$

Thm K is ($\boldsymbol{\varepsilon}$, δ)-differentially private as long as $\sigma \geq \sqrt{2\ln(2/\delta)} / \varepsilon \times GSf$

Idea Use δ to exclude the tails of the gaussian distribution



Multivariate Gaussian Mechanism

Suppose that f outputs a length d vector instead of a number

$\mathrm{K}(D) = f(D) + \mathrm{N}(\sigma 2)d$

Thm K is ($\boldsymbol{\varepsilon}$, δ)-differentially private as long as
$\sigma \geq \sqrt{2\ln(2/\delta)} / \varepsilon \times \max_{D1 \approx D2} \| f(D1) - f(D2) \| 2$

Remark: Similar results would hold with the Laplacian Mechanism, but we would need to add more noise (proportional to the larger L1 norm)

Approximate Differential Privacy

- Key Difference
  - Approximate Differential Privacy does NOT require that:

$$\text{Range}(\kappa(D1)) = \text{Range}(\kappa(D2))$$

- The privacy guarantees made by *(ε,δ)-differential privacy* are not as strong as ε-differential privacy, but less noise is required to achieve *(ε,δ)-differential privacy*.

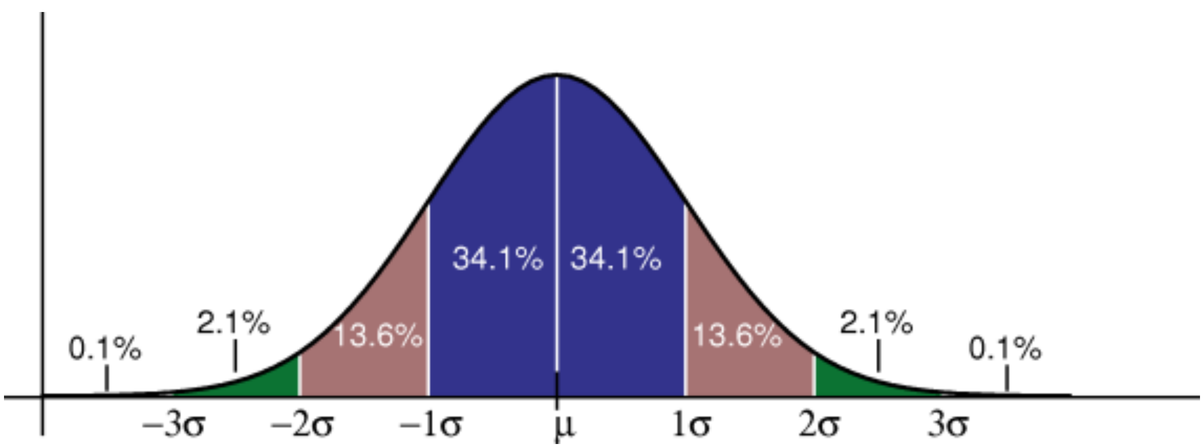
Achieving Approximate Differential Privacy

Key Differences:

- Use of the L2 norm instead of L1 norm to define the sensitivity of Δf

$$\max_{A \approx B} \|f(A) - f(B)\|_2$$

- Use of Gaussian Noise instead of Laplace Noise



Differential Privacy for Netflix Queries

- What level of granularity to consider? What does it mean for databases D1 and D2 to differ on at most one element?
  - One user (column) is present in D1 but not in D2
  - One rating (cell) is present in D1 but not in D2

- Issue 1: Given a query "how would user i rate movie j?" Consider: K(D-u[i]) - how can it possibly be accurate?

- Issue 2: If the definition of differing in at most one element is taken over cells, then what privacy guarantees are made for a user with many data points?

Netflix Predictions – High Level

- Q(i,j) – "How would user i rate movie j?"

- Predicted rating may typically depend on
  - Global average rating over all movies and all users
  - Average movie rating of user i
  - Average rating of movie j
  - Ratings user i gave to *similar* movies
  - Ratings *similar users* gave to movie j

- Sensitivity may be small for many of these queries

Personal Rating Scale

- For Alice a rating of 3 might mean the movie was really terrible.
- For Bob the same rating might mean that the movie was excellent.
- How do we tell the difference?

$$r_{im} - \bar{r}_i > 0?$$

How do we tell if two users are similar?

Pearson's Correlation is one metric for similarity of users i and j

- Consider all movies rated by both users
- Negative value whenever i likes a movie that j dislikes
- Positive value whenever i and j agree

$$S(i,j) = \sum_{m \in L_i \bigcap L_j} (r_{im} - \bar{r}_i)(r_{jm} - \bar{r}_j)$$

We can use similar metrics to measure the similarity between two movies.

Netflix Predictions Example

- Collaborative Filtering
  - Find the k-nearest neighbors of user i who have rated movie j by Pearson's Correlation:

$$S(i, j)$$

similarity of users i and j

$$N_i(k, j) = \{u1, \ldots, uk\}$$

k most similar users

  - Predicted Rating

$$p_{ij} = \bar{r}_i + \frac{1}{k} \sum_{u \in N_i(k,j)} (r_{uj} - \bar{r}_u)$$

Netflix Prediction Sensitivity Example

$$p_{ij} = \bar{r}_i + \tfrac{1}{k} \sum_{u \in N_i(k,j)} (\bar{r}_{uj} - \bar{r}_u)$$

- Pretend the query Q(i,j) included user i's rating history
- At most one of the neighbors ratings changes, and the range of ratings is 4 (since ratings are between 1 & 5). The L1 sensitivity of the prediction is:

$$\Delta p = 4/k$$

Similarity of Two Movies

- Let U be the set of all users who have rated both movies i and j then

$$S(i,j) = \sum_{u \in U} (r_{uj} - \bar{r}_u) \times (r_{ui} - \bar{r}_u)$$

K-Nearest Users or K-Nearest Movies?

Find k most similar *users* to i that have also rated movie j?

Find k most similar *movies* to j that user i has rated?

Either way, after some pre-computation, we need to be able to find the k-nearest users/movies quickly!

Covariance Matrix

Movie-Movie Covariance Matrix

- (MxM) matrix
- Cov[i][j] measures similarity between movies i and j
- M ≈ 17,000
- More accurate

User-User Covariance Matrix?

- (NxN) Matrix to measure similarity between users
- N ≈ 500,000
- Less accurate

What do we need to make predictions?

For a large class of prediction algorithms it suffices to have:

- Gavg – average rating for all movies by all users
- Mavg – average rating for each movie by all users
- Average Movie Rating for each user
- Movie-Movie Covariance Matrix (COV)

Differentially Private Recommender Systems (High Level)

To respect approximate differential privacy publish

- Gavg + NOISE
- Mavg + NOISE
- COV + NOISE

- ΔGavg, ΔMavg are very small so they can be published with little noise
- ΔCOV requires more care (our focus)

- Don't publish average ratings for users (used in per-user prediction phase using k-NN or other algorithms)

Movie-Movie Covariance Matrix

$$Cov=\sum_{u} (r_{u})(r_{u})^{T}$$

$$r_{u} = r_{u} - r$$

User u's rating for each movie

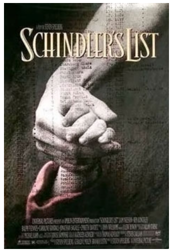
Average rating for each movie

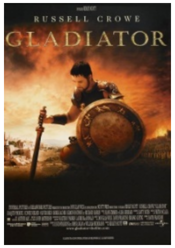
Movie-Movie Covariance Matrix

$$Cov=\sum_{u} (r_{u})(r_{u})^{T}$$

$$r=\left\langle \begin{matrix}3.2\\2\\3\end{matrix} \right\rangle$$

$$r_{u1}=\left\langle \begin{matrix}4.2\\2\\3\end{matrix} \right\rangle \qquad r_{u2}=\left\langle \begin{matrix}1.5\\4.5\\2\end{matrix} \right\rangle$$





Movie-Movie Covariance Matrix

$$Cov=\sum_{u} (r_u)(r_u)^T$$

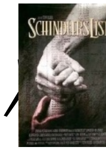
$$r=\begin{pmatrix}3.2\\2\\3\end{pmatrix}$$

$$r_{u1}=\begin{pmatrix}1\\0\\0\end{pmatrix}$$

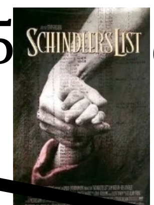
$$r_{u2}=\begin{pmatrix}-1.7\\2.5\\-1\end{pmatrix}$$

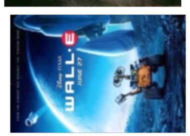
Example

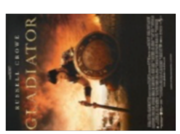
$r\downarrow u1 \;\; (r\downarrow u1\;)\uparrow T = \langle ■-1.7 @2.5@-1 \;\rangle -1.7$ [illegible] $.5\&$ [illegible] $\rangle$



$\mathbf{-4.25 = -1.7 \times 2.5}$ $[■2.89\&-4.25$ $@-4.25\&$ [illegible] $\&-2.5$ [illegible] $\&$
$-2.5\&1\;]$

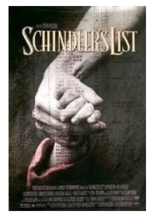




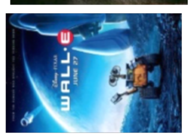


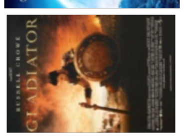
Example

$$Cov = r_{u1}(r_{u1})^T + r_{u2}(r_{u2})^T$$



$$= \begin{bmatrix} \blacksquare & -4.25 & 1.7 \\ -4.25 & 6.25 & -2.5 \\ 1.7 & -2.5 & 1 \end{bmatrix}$$





Covariance Matrix Sensitivity

$$\mathrm{Cov} = \sum_u r_u r_u^T$$

$$\begin{aligned} \|\mathrm{Cov}^a - \mathrm{Cov}^b\| &= \|r_u^a r_u^{aT} - r_u^b r_u^{bT}\| \\ &\leq \|r_u^a - r_u^b\| \times (\|r_u^a\| + \|r_u^b\|) \end{aligned}$$

- Could be large if a user's rating has large spread or if a user has rated many movies

Covariance Matrix Trick I

- Center and clamp all ratings around averages. If we use clamped ratings then we reduce the sensitivity of our function.

$$\widehat{r}_{ui} = \begin{cases} -B, & \text{if } r_{ui} - \overline{r}_u < -B, \\ r_{ui} - \overline{r}_u, & \text{if } -B \leq r_{ui} - \overline{r}_u < B \\ B, & \text{if } B \leq r_{ui} - \overline{r}_u. \end{cases}$$

Example (B = 1)

User 1:

$$r_{u1} = \langle 4.2 \;\; 2 \;\; 3 \rangle$$

$$\bar{r}_{u1} = (4.2+2+3)/3 \approx 3.07$$

$$r_{u1} = \langle 1 \;\; -1 \;\; -.07 \rangle$$

$$\min_{T} \{B, 4.2-3.07\}$$

$$\max_{T} \{-B, 2-3.07\}$$

Covariance Matrix Trick II

- Carefully weight the contribution of each user to reduce the sensitivity of the function. Users who have rated more movies are assigned lower weight.

$$\mathrm{Cov} \quad = \quad \sum_u w_u \widehat{r}_u \widehat{r}_u^T + \mathrm{Noise}^{d \times d}$$

- Where e_{ui} is 1 if user u rated movie i

  and $w_u = 1/||e_u||_2$

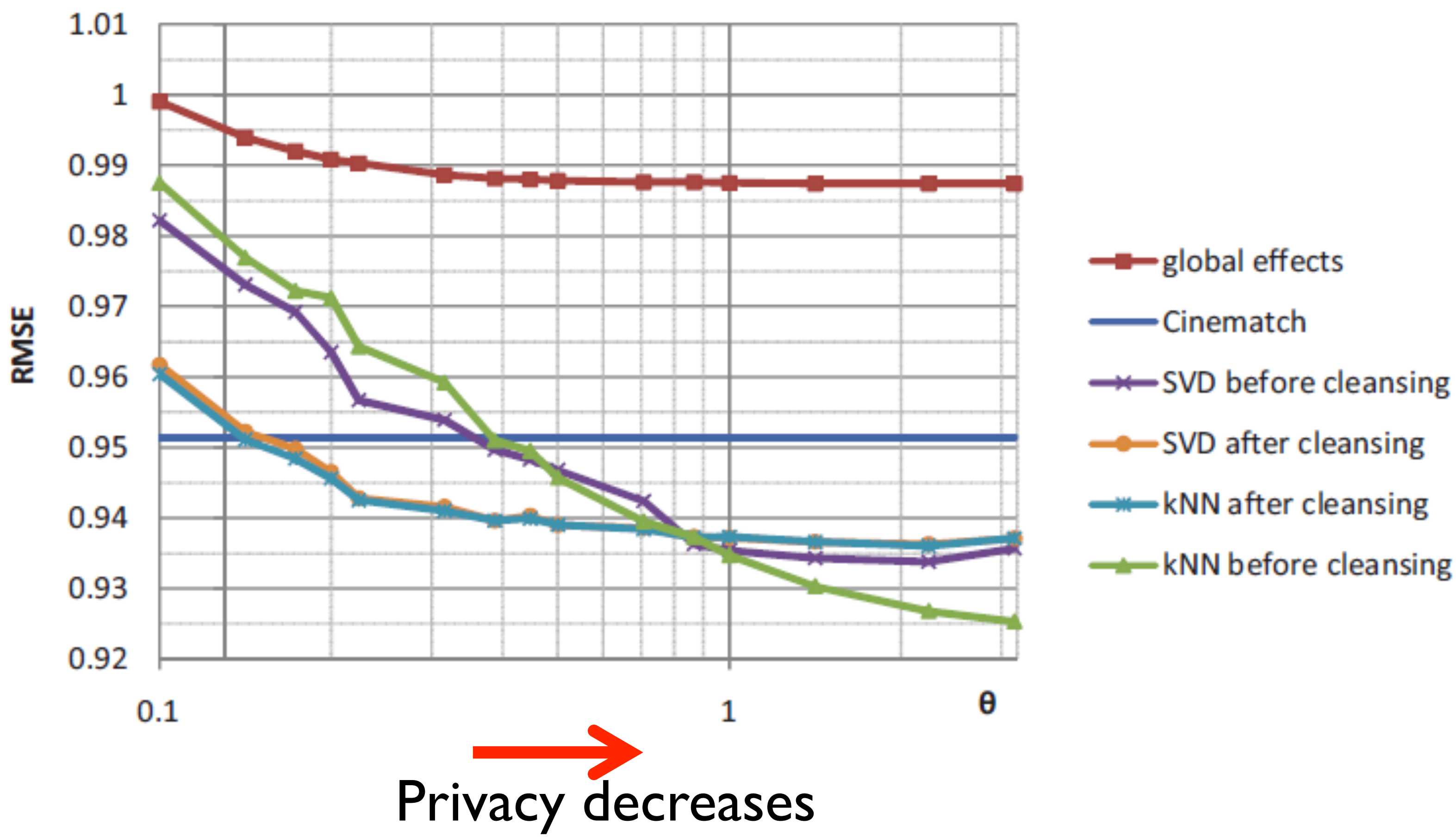
Publishing the Covariance Matrix

- Theorem (roughly):

$$\|w_u^a \widehat{r}_u^a \widehat{r}_u^{aT} - w_u^b \widehat{r}_u^b \widehat{r}_u^{bT}\|_2 \quad \leq \quad (1 + 2\sqrt{2})B^2$$

- Add independent Gaussian noise proportional to this sensitivity bound to each entry in covariance matrix

Experimental Results



Note About Results

- Granularity: One *rating* present in D1 but not in D2
  - Accuracy is much lower when one user is present in D1 but not in D2
  - Intuition: Given query Q(i,j) the database D-u[i] gives us no history about user i.
- Approximate Differential Privacy
  - Gaussian Noise added according to L2 Sensitivity
  - Clamped Ratings (B =1) to further reduce noise

Acknowledgment

- A number of slides are from Jeremiah Blocki

Global Averages

$$
\begin{aligned}
\mathrm{GSum} &= \sum_{u,i} r_{ui} + \mathrm{Noise}, \\
\mathrm{GCnt} &= \sum_{u,i} e_{ui} + \mathrm{Noise}.
\end{aligned}
$$

$$\mathrm{G} = \mathrm{GSum}/\mathrm{GCnt}$$

$$
\begin{aligned}
\mathrm{MSum} &= \sum_{u} r_{u} + \mathrm{Noise}^d, \\
\mathrm{MCnt} &= \sum_{u} e_{u} + \mathrm{Noise}^d.
\end{aligned}
$$

$$\mathrm{MAvg}_i = \frac{\mathrm{MSum}_i + \beta_m \mathrm{G}}{\mathrm{MCnt}_i + \beta_m}.$$

Theorem

THEOREM 4. *Let r^a and r^b differ on one rating, present in r^b. Let α be the maximum possible difference in ratings*[2]*. For centered and clamped ratings $\widehat{r}^a$ and $\widehat{r}^b$, we have*

$$\begin{aligned} \|\widehat{r}^a - \widehat{r}^b\|_1 &\leq \alpha + B, \\ \|\widehat{r}^a - \widehat{r}^b\|_2^2 &\leq \frac{\alpha^2}{4\beta_p} + B^2. \end{aligned}$$

[2]For the Netflix Prize data set $\alpha = 4$.