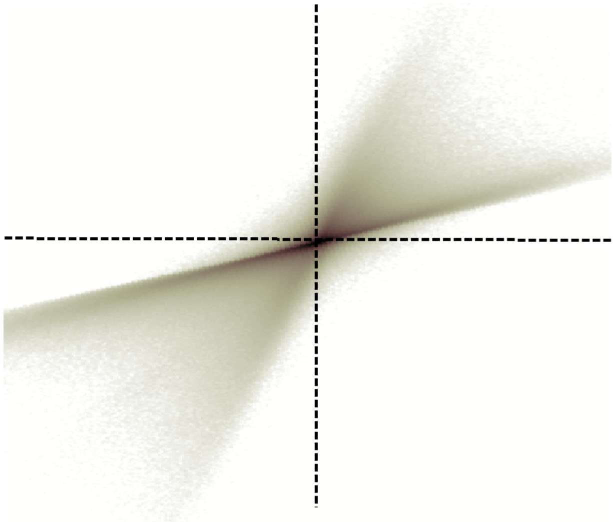
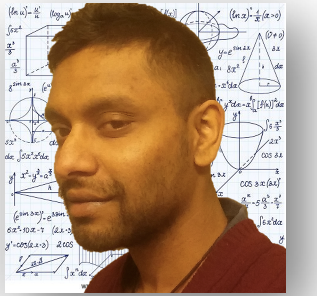
1	5			
		5	3	5
		5		5
3		5	4	5
	5			
4		3		
	1	1	4	



Deconvolving Feedback Loops in Recommender Systems

Paper `arXiv:1703.01049`
Code `github.com/sinhayan/Deconvolving_Feedback_Loops`

Karthik Ramani

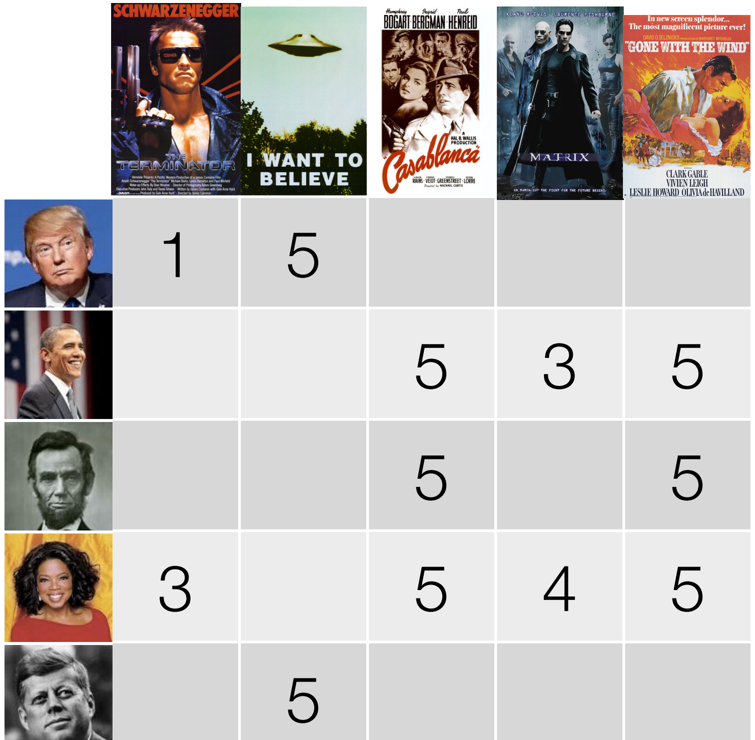

Ayan Sinha

David F. Gleich
Purdue University

Summary of results

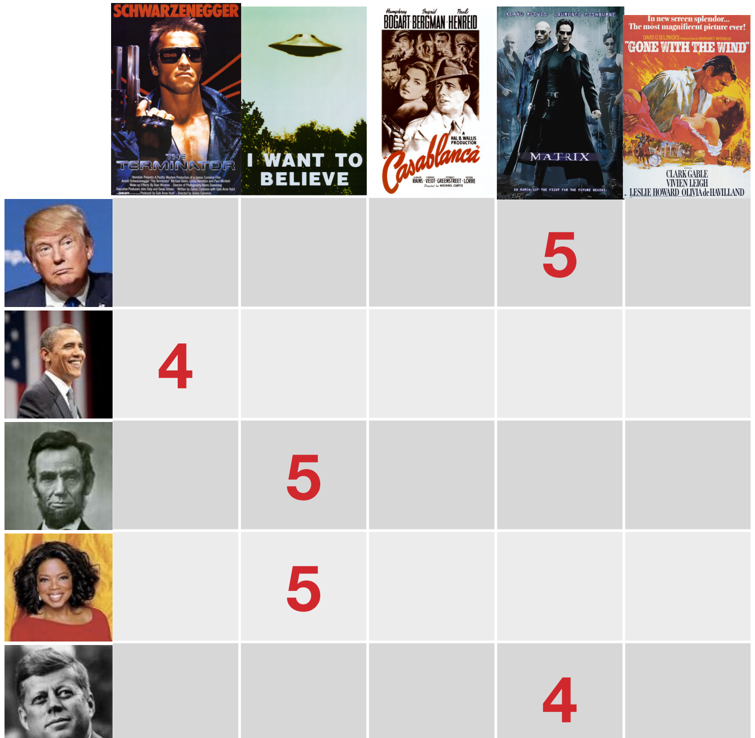
Recommender systems introduce feedback into the ratings matrix.

Last weekend



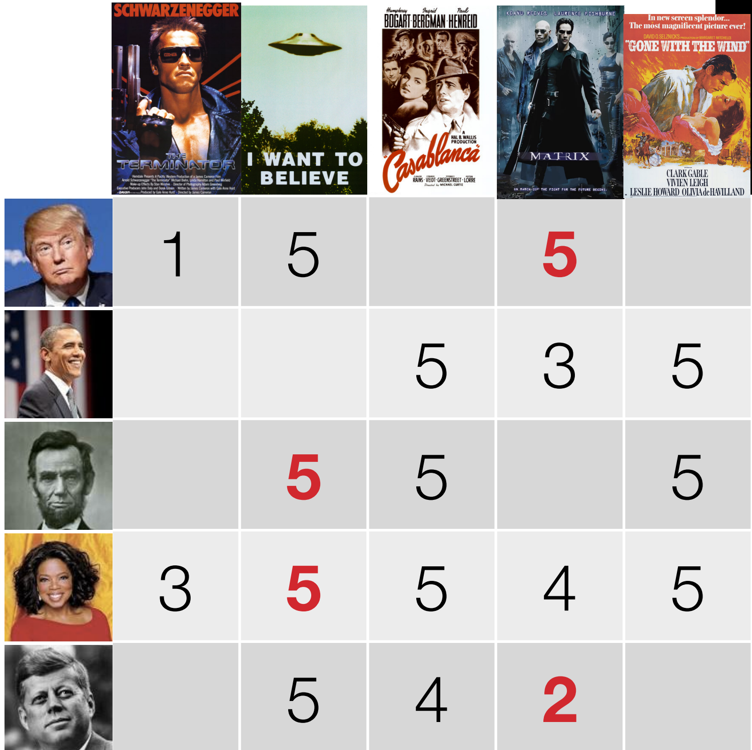
	TERMINATOR	I WANT TO BELIEVE	Casablanca	MATRIX	GONE WITH THE WIND
	1	5			
			5	3	5
			5		5
	3		5	4	5
		5			

Recommended



	TERMINATOR	I WANT TO BELIEVE	Casablanca	MATRIX	GONE WITH THE WIND
				5	
	4				
		5			
		5			
				4	

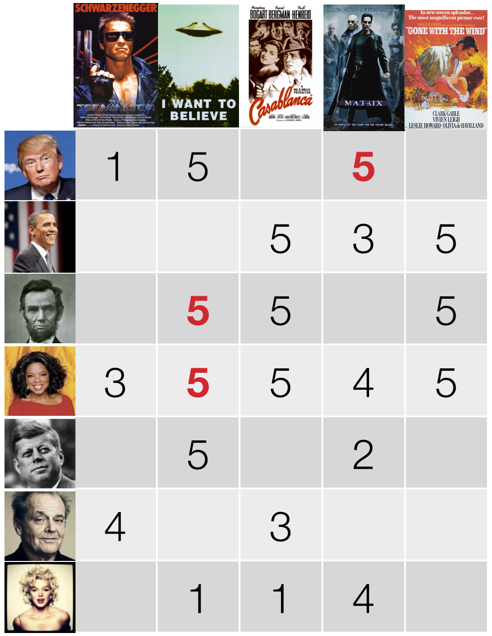
This weekend



	TERMINATOR	I WANT TO BELIEVE	Casablanca	MATRIX	GONE WITH THE WIND
	1	5		5	
			5	3	5
		5	5		5
	3	5	5	4	5
		5	4	2	

Summary of results

Recommender systems introduce feedback into the ratings matrix.

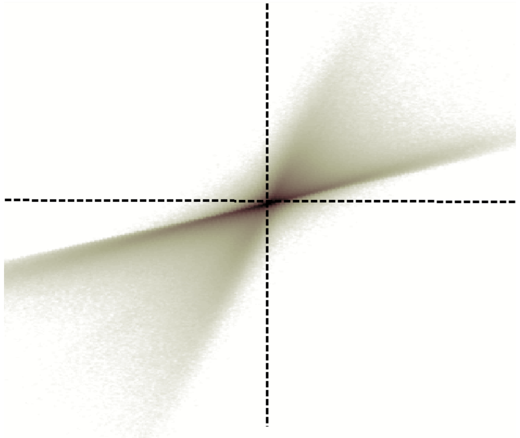


	1	5		**5**	
			5	3	5
		5	5		5
	3	**5**	5	4	5
		5		2	
	4		3		
		1	1	4	

We propose an SVD-based method that deconvolves that effect* with one matrix.

ALG

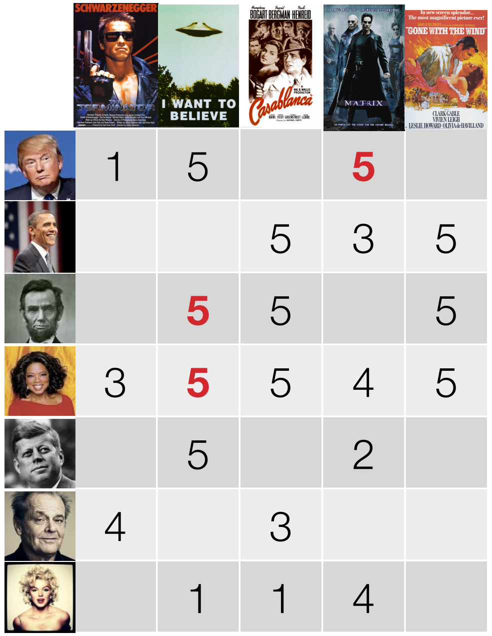
Skew in deconvolved vs. given ratings scatter-plot scores rec. effects.



* Will be explained soon!

Summary of results

Recommender systems introduce feedback into the ratings matrix.



1	5		5	
		5	3	5
	5	5		5
3	5	5	4	5
	5		2	
4		3		
	1	1	4	

We propose an SVD-based method that deconvolves that effect* with one matrix.

ALG

Skew in deconvolved vs. given ratings scatter-plot scores rec. effects.

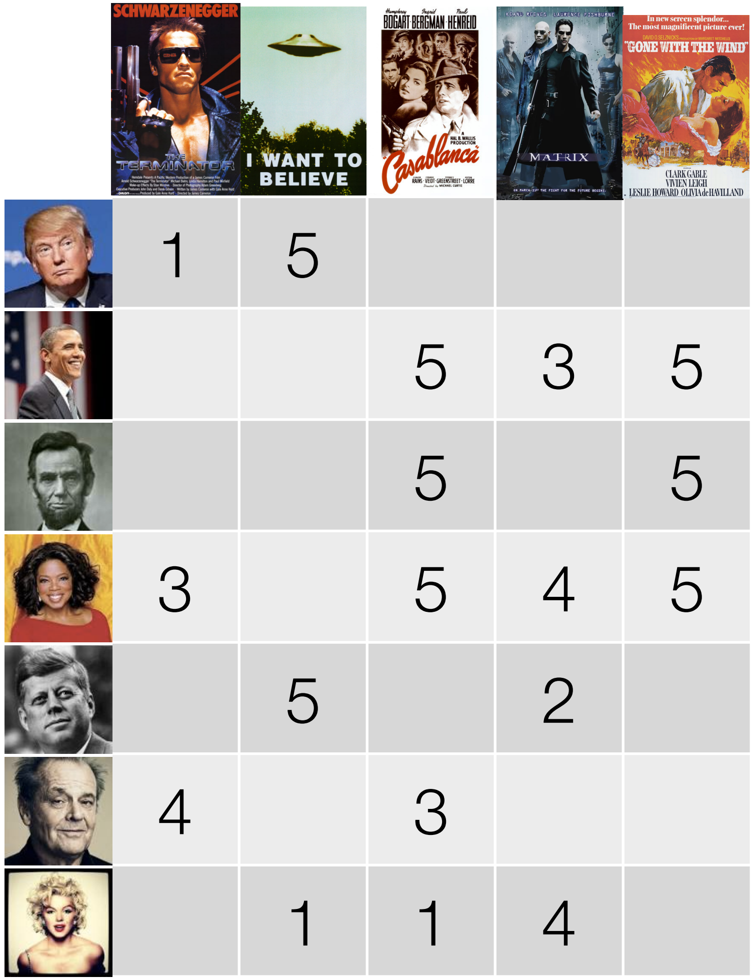
It also scores system rec. effects

Dataset	Score
Jester	0.0487
MusicLab-Weak	0.1073
MusicLab-Strong	0.1509
MovieLens-10M	0.3821
BeerAdvocate	0.2223
Fine Foods	0.1209
Netflix	0.2661

There's just one problem.

Doing this is impossible.

A new rating can come from more than just a user or via the recommender system.



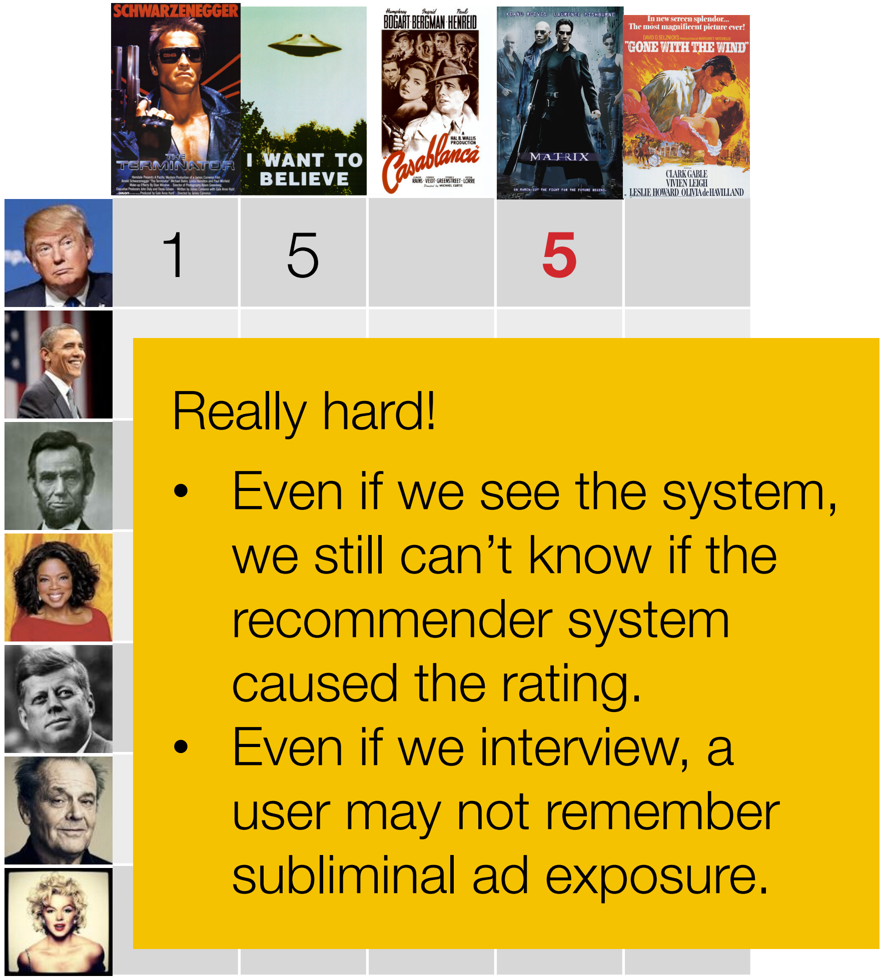
	THE TERMINATOR	I WANT TO BELIEVE	Casablanca	MATRIX	GONE WITH THE WIND
	1	5			
			5	3	5
			5		5
	3		5	4	5
		5		2	
	4		3		
		1	1	4	

We predict *Thumbs Up*

Recommendation

TV Advertisement

Friend



	THE TERMINATOR	I WANT TO BELIEVE	Casablanca	MATRIX	GONE WITH THE WIND
	1	5		**5**	

Really hard!

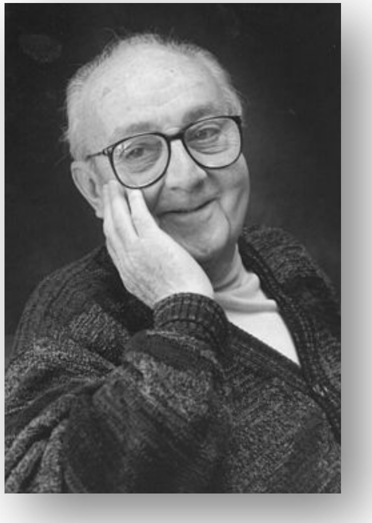
- Even if we see the system, we still can't know if the recommender system caused the rating.
- Even if we interview, a user may not remember subliminal ad exposure.

Time

That's okay!

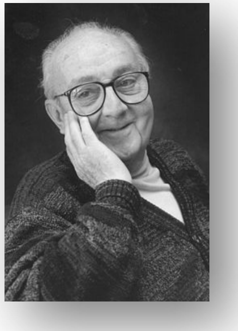
We do impossible stuff all the time.
(e.g. models of the universe)

But we need strong models.



“All models are wrong but some are useful.”

Our goal is a model of recommender systems that we can invert to understand the effects.



Assumption 0. The observed ratings are a mixture of

true ratings and *recommended items*

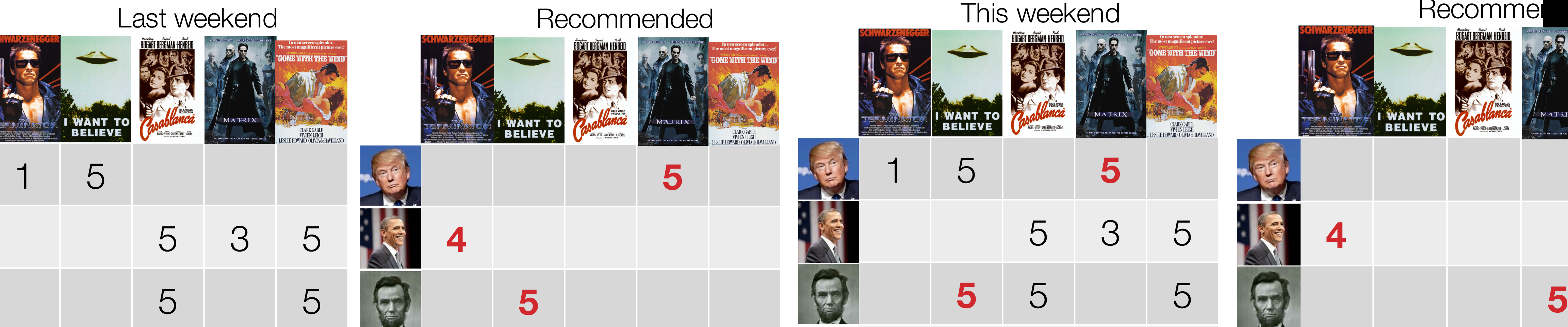
$$\boldsymbol{R}_{\text{obs}} = \boldsymbol{R}_{\text{true}} + \boldsymbol{R}_{\text{recom}}$$

Exogenous effects are either true or recommended ☺

Assumption 1. The recommender system uses an item-item similarity matrix $\boldsymbol{S}$ and feedback occurs through this.

$$\boldsymbol{R}_{\text{obs}} = \boldsymbol{R}_{\text{true}} + \boldsymbol{H} \odot (\boldsymbol{R}_{\text{obs}}\boldsymbol{S})$$

$\boldsymbol{H}$ gives the actual selections via an element-wise prod.



Last weekend

Terminator	I want to believe	Casablanca	Matrix	Gone with the wind
1	5			
		5	3	5
		5		5

Recommended

	Terminator	I want to believe	Casablanca	Matrix	Gone with the wind
Trump				5	
Obama	4				
Lincoln		5			

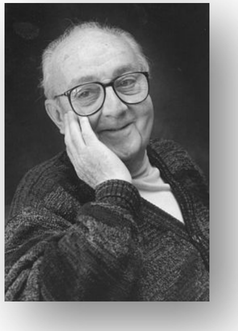
This weekend

	Terminator	I want to believe	Casablanca	Matrix	Gone with the wind
Trump	1	5		5	
Obama			5	3	5
Lincoln		5	5		5

Recommen

	Terminator	I want to believe	Casablanca	Matrix
Trump				
Obama	4			
Lincoln				5

Our goal is a model of recommender systems that we can invert to understand the effects.



Assumption 0. The observed ratings are a mixture of

true ratings and *recommended items*

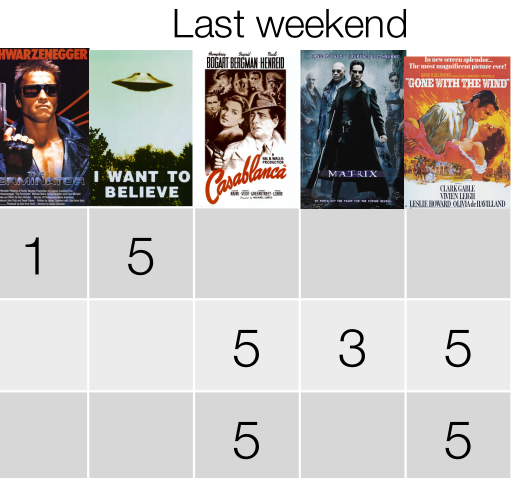
$$\boldsymbol{R}_{\text{obs}} = \boldsymbol{R}_{\text{true}} + \boldsymbol{R}_{\text{recom}}$$

Exogenous effects are either true or recommended ☺

Assumption 1. The recommender system uses an item-item similarity matrix $\boldsymbol{S}$ and feedback occurs through this.

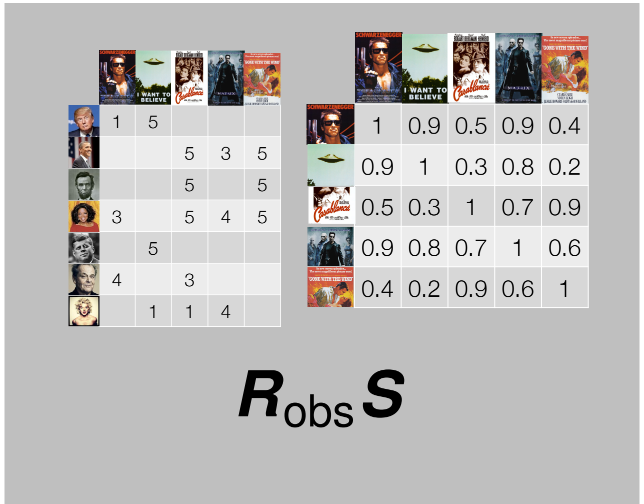
$$\boldsymbol{R}_{\text{obs}} = \boldsymbol{R}_{\text{true}} + \boldsymbol{H} \odot (\boldsymbol{R}_{\text{obs}}\boldsymbol{S})$$

$\boldsymbol{H}$ gives the actual selections via an element-wise prod.



Last weekend

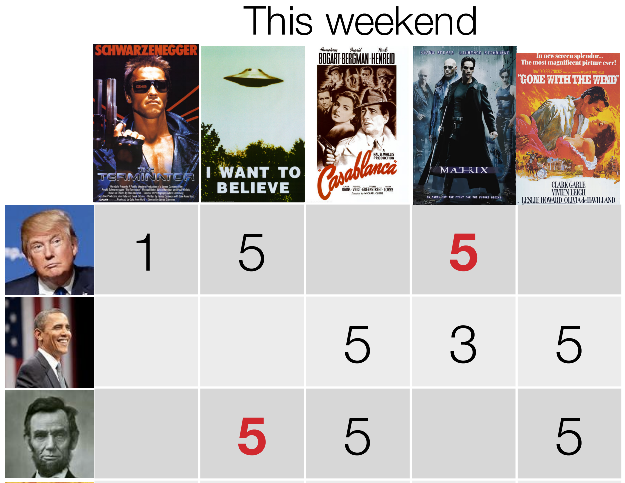
1	5			
		5	3	5
		5		5



1	5			
		5	3	5
		5		5
3		5	4	5
	5			
4		3		
	1	1	4	

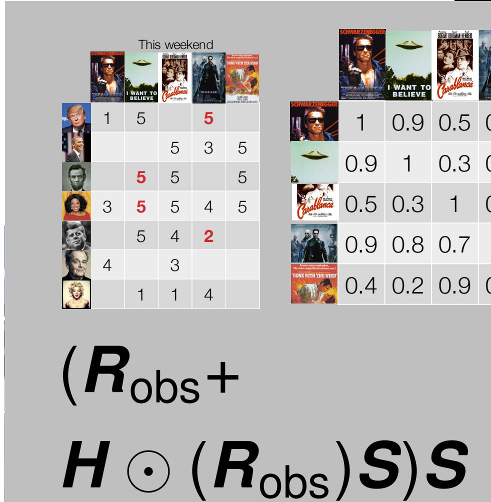
1	0.9	0.5	0.9	0.4
0.9	1	0.3	0.8	0.2
0.5	0.3	1	0.7	0.9
0.9	0.8	0.7	1	0.6
0.4	0.2	0.9	0.6	1

$$\boldsymbol{R}_{\text{obs}}\boldsymbol{S}$$



This weekend

1	5		**5**	
		5	3	5
	5	5		5



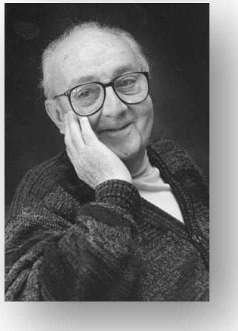
This weekend

1	5		**5**	
		5	3	5
	5	5		5
3	**5**	5	4	5
	5	4	**2**	
4		3		
	1	1	4	

1	0.9	0.5
0.9	1	0.3
0.5	0.3	1
0.9	0.8	0.7
0.4	0.2	0.9

$$(\boldsymbol{R}_{\text{obs}} + \boldsymbol{H} \odot (\boldsymbol{R}_{\text{obs}})\boldsymbol{S})\boldsymbol{S}$$

Our goal is a model of recommender systems that we can invert to understand the effects.



Assumption 0. The observed ratings are a mixture of

true ratings and *recommended items*

$$\boldsymbol{R}_{\text{obs}} = \boldsymbol{R}_{\text{true}} + \boldsymbol{R}_{\text{recom}}$$

Exogenous effects are either true or recommended ☺

Assumption 1. The recommender system uses an item-item similarity matrix ***S*** and feedback occurs through this.

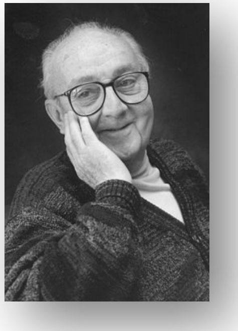
$$\boldsymbol{R}_{\text{obs}} = \boldsymbol{R}_{\text{true}} + \boldsymbol{H} \odot (\boldsymbol{R}_{\text{obs}}\boldsymbol{S})$$

H gives the actual selections via an element-wise prod.

$$\begin{aligned}\boldsymbol{R}_{\text{obs}} &= \boldsymbol{R}_{\text{true}} + \boldsymbol{H}^{(2)} \odot \Big(\big(\boldsymbol{R}_{\text{true}} + \boldsymbol{H}^{(1)} \odot (\boldsymbol{R}_{\text{obs}}\boldsymbol{S}^{(1)})\big) \boldsymbol{S}^{(2)} \Big) \\ &= \boldsymbol{R}_{\text{true}} + \boldsymbol{H}^{(2)} \odot \boldsymbol{R}_{\text{true}}\boldsymbol{S}^{(2)} + \boldsymbol{H}^{(2)} \odot (\boldsymbol{H}^{(1)} \odot (\boldsymbol{R}_{\text{true}}\boldsymbol{S}^{(1)}))\boldsymbol{S}^{(2)} + \cdots\end{aligned}$$

This process fills in the matrix, but we have no control over ***H***.

Our goal is a model of recommender systems that we can invert to understand the effects.



Assumption 2. We model the effect of ***H*** in expectation with independent coin-tosses on accepting recommendation.

$$\boldsymbol{E}[\boldsymbol{H} \odot \boldsymbol{R}_{\text{recom}}] = \alpha \boldsymbol{R}_{\text{recom}}$$

Each recommendation is accepted with prob. α

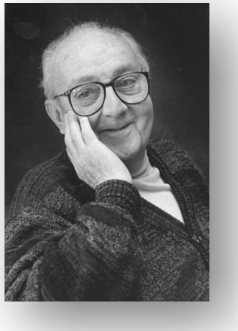
$$\begin{aligned} \boldsymbol{R}_{\text{obs}} &= \boldsymbol{R}_{\text{true}} + \boldsymbol{H}^{(2)} \odot \boldsymbol{R}_{\text{true}}\boldsymbol{S} + \boldsymbol{H}^{(2)} \odot (\boldsymbol{H}^{(1)} \odot (\boldsymbol{R}_{\text{true}}\boldsymbol{S}))\boldsymbol{S} + \cdots \\ &= \boldsymbol{R}_{\text{obs}} = \boldsymbol{R}_{\text{true}} + \alpha \boldsymbol{R}_{\text{true}}\boldsymbol{S} + \alpha^2 \boldsymbol{R}_{\text{true}}\boldsymbol{S}^2 + \alpha^3 \boldsymbol{R}_{\text{true}}\boldsymbol{S}^3 + \cdots \\ &= \boldsymbol{R}_{\text{true}}(\mathrm{I} + \alpha \boldsymbol{S} + \alpha^2 \boldsymbol{S}^2 + \alpha^3 \boldsymbol{S}^3 + \cdots) \end{aligned}$$

For simplicity, we use const. ***S***, see paper to avoid.

This means we are modeling *expected* behavior vs. actual behavior.

This gives us a nice expression, but what is ***S*** ?

Our goal is a model of recommender systems that we can invert to understand the effects.



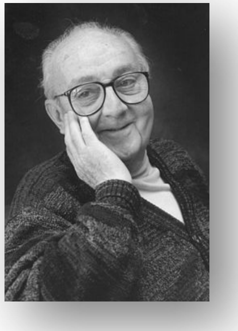
Assumption 3. The user means and item norms of $\boldsymbol{R}_{\text{true}}$ and $\boldsymbol{R}_{\text{obs}}$ are close enough that we consider them the same.

Assumption 4. The item-item similarity matrix $\boldsymbol{S}$ is induced by $\boldsymbol{R}_{\text{true}}$.
This can be avoided (see the paper) but the presentation is more obscure.

Together, these assumptions can be interpreted as

- the recommender system is a second-order effect
- it isn't powerful enough to "change the world"
- it's being used in a time-span where big changes don't occur
- we care about relative rankings

Our goal is a model of recommender systems that we can invert to understand the effects.



Assumption 3. The user means and item norms of $\boldsymbol{R}_{\text{true}}$ and $\boldsymbol{R}_{\text{obs}}$ are close enough that we consider them the same.

Assumption 4. The item-item similarity matrix $\boldsymbol{S}$ is induced by $\boldsymbol{R}_{\text{true}}$.
This can be avoided (see the paper) but the presentation is more obscure.

$$S(i,j) = \frac{\sum_{u \in U}(\boldsymbol{R}_{u,i} - \bar{\boldsymbol{R}}_u)(\boldsymbol{R}_{u,j} - \bar{\boldsymbol{R}}_u)}{\sqrt{\sum_{u \in U}(\boldsymbol{R}_{u,i} - \bar{\boldsymbol{R}}_u)^2}\sqrt{\sum_{u \in U}(\boldsymbol{R}_{u,j} - \bar{\boldsymbol{R}}_u)^2}}$$

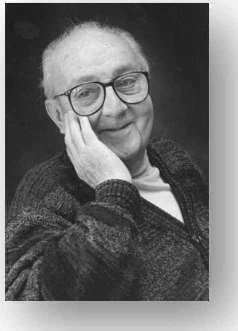
Adjusted or centered cosine similarity

$$\tilde{\boldsymbol{R}}_{u,i} = \boldsymbol{R}_{u,i} - \bar{\boldsymbol{R}}_u; \text{and}, \hat{\boldsymbol{R}}_{u,i} = \frac{\tilde{\boldsymbol{R}}_{u,i}}{\|\tilde{\boldsymbol{R}}_i\|} = \frac{\boldsymbol{R}_{u,i} - \bar{\boldsymbol{R}}_u}{\sqrt{\sum_{u \in U}(\boldsymbol{R}_{u,i} - \bar{\boldsymbol{R}}_u)^2}}$$

Centering and normalizing

$$\hat{\boldsymbol{R}}_{\text{obs}} = \hat{\boldsymbol{R}}_{\text{true}}(\mathrm{I} + \alpha \hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}} + \alpha^2 (\hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}})^2 + \alpha^3 (\hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}})^3 + \cdots)$$

Our goal is a model of recommender systems that we can invert to understand the effects.



Assumption 5. The spectral radius of $\alpha \hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}} \leq 1$
This is a technical scaling assumption. We don't *need* it as we could pick a different scaling for $\hat{\boldsymbol{R}}_{\text{true}}$

$$\boldsymbol{I} + \alpha \hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}} + \alpha^2 (\hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}})^2 + \ldots = (\boldsymbol{I} - \alpha \hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}})^{-1}$$

$$\hat{\boldsymbol{R}}_{\text{obs}} = \hat{\boldsymbol{R}}_{\text{true}} (\boldsymbol{I} + \alpha \hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}} + \alpha^2 (\hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}})^2 + \alpha^3 (\hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}})^3 + \cdots)$$

$$\hat{\boldsymbol{R}}_{\text{obs}} = \hat{\boldsymbol{R}}_{\text{true}} (\boldsymbol{I} - \alpha \hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}})^{-1}$$

The driving equation for the feedback

These assumptions are strong, but (we argue) not entirely unreasonable

Assumption 1. The recommender system uses an item-item similarity matrix $\boldsymbol{S}$ and feedback occurs through this.

- Reasonable for “early” recommenders.

Assumption 2. We model the effect of **H** in expectation with independent coin-tosses on accepting recommendation.

- Reasonable for a model.

Assumption 3, 4. The user means and item means of $\boldsymbol{R}_{\text{true}}$ and $\boldsymbol{R}_{\text{obs}}$ are close enough that we consider them the same. The item-item similarity matrix $\boldsymbol{S}$ is induced by $\boldsymbol{R}_{\text{true}}$.

- Strong, and they can be replaced with some equally strong but less *wrong* variants.

Assumption 5. The spectral radius of

$$\alpha \hat{\boldsymbol{R}}_{\text{true}}^T \hat{\boldsymbol{R}}_{\text{true}} \leq 1$$

- Relatively incidental. Just governs the scaling constant of the final numbers.

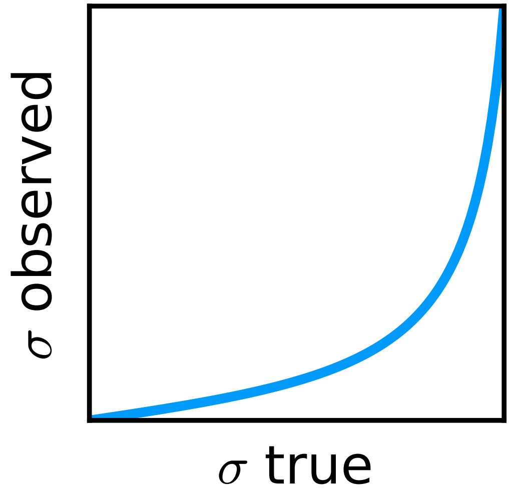
Our recommender inversion theorem gives an algorithm to deconvolve a ratings matrix.

Assuming the RS follows the driving equation,

$$\hat{\boldsymbol{R}}_{\text{obs}} = \hat{\boldsymbol{R}}_{\text{true}}(\boldsymbol{I} - \alpha \hat{\boldsymbol{R}}_{\text{true}}^{T} \hat{\boldsymbol{R}}_{\text{true}})^{-1}$$

α is between 0 and 1, and the singular value decomposition of the observed rating matrix is, $\hat{\boldsymbol{R}}_{\text{obs}} = \boldsymbol{U}\boldsymbol{\Sigma}_{\text{obs}}\boldsymbol{V}^{T}$, the deconvolved matrix $\boldsymbol{R}_{\text{true}}$ of true ratings is given as $\boldsymbol{U}\boldsymbol{\Sigma}_{\text{true}}\boldsymbol{V}^{T}$, where the $\boldsymbol{\Sigma}_{\text{true}}$ is a diagonal matrix with elements:

$$\sigma_i^{\text{true}} = \frac{-1}{2\alpha\sigma_i^{\text{obs}}} + \sqrt{\frac{1}{4\alpha^2(\sigma_i^{\text{obs}})^2} + \frac{1}{\alpha}}$$



Proof. Write out the SVD of $\boldsymbol{R}_{\text{true}}$ and then we just get a polynomial expression in the SVD of $\boldsymbol{R}_{\text{true}}$ that we can solve.

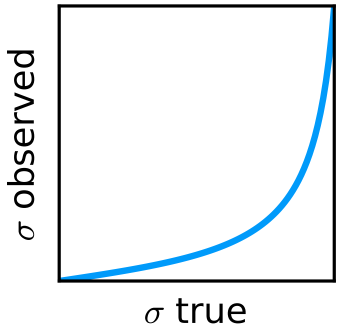
Our recommender inversion theorem gives an algorithm to deconvolve a ratings matrix.

Input. $\boldsymbol{R}_{\text{obs}}$, α, k, where $\boldsymbol{R}_{\text{obs}}$ is observed ratings matrix, α is parameter governing feedback loops and k is number of singular values

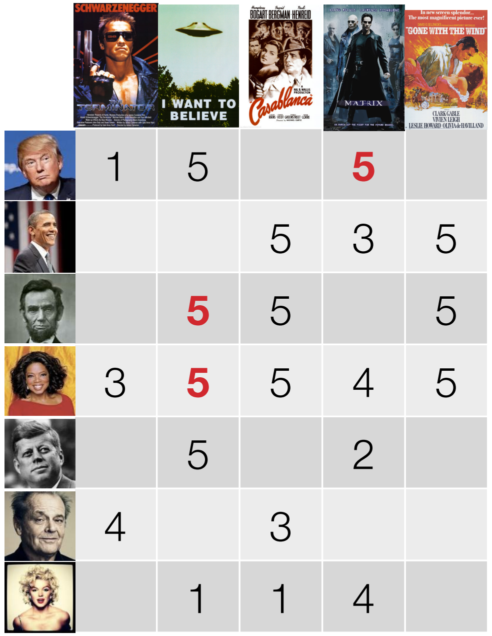
Output. $\hat{\boldsymbol{R}}_{\text{true}}$, True rating matrix

1. Compute $\tilde{\boldsymbol{R}}_{\text{obs}}$ given $\boldsymbol{R}_{\text{obs}}$, where $\tilde{\boldsymbol{R}}_{\text{obs}}$ is user centered observed matrix
2. Compute $\hat{\boldsymbol{R}}_{\text{obs}} \leftarrow \tilde{\boldsymbol{R}}_{\text{obs}} D_N^{-1}$, where $\hat{\boldsymbol{R}}_{\text{obs}}$ is item-normalized rating matrix, and D_N^{-1} is diagonal matrix of item-norms $D_N(i,i) = \sqrt{\sum_{u \in U}(\boldsymbol{R}_{u,i} - \bar{\boldsymbol{R}}_u)^2}$
3. Solve $\boldsymbol{U}\boldsymbol{\Sigma}_{\text{obs}}\boldsymbol{V}^T \leftarrow SVD(\hat{\boldsymbol{R}}_{\text{obs}}, k)$, the truncated SVD corresponding to k largest singular values.
4. Perform $\sigma_i^{\text{true}} \leftarrow \left(\frac{-1}{2\alpha\sigma_i^{\text{obs}}} + \sqrt{\frac{1}{4\alpha^2(\sigma_i^{\text{obs}})^2} + \frac{1}{\alpha}}\right)$ for all i
5. **return** $\boldsymbol{U}, \boldsymbol{\Sigma}_{\text{true}}, \boldsymbol{V}^T$

We approximate by truncating the SVD.



Summary of results


SCHWARZENEGGER
I WANT TO BELIEVE
BOGART BERGMAN HENREID
Casablanca
MATRIX
GONE WITH THE WIND

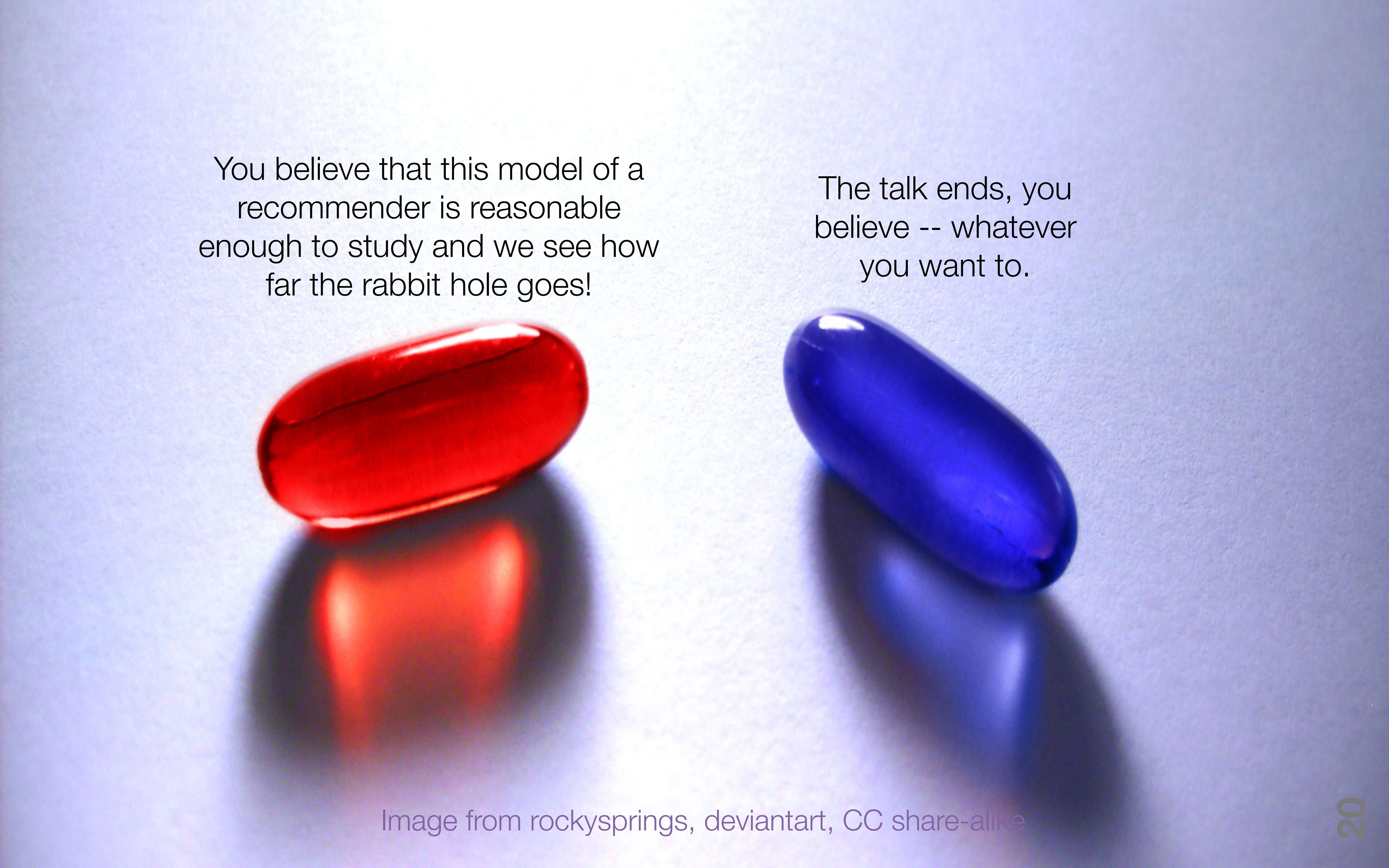
1	5		**5**	
		5	3	5
	5	5		5
3	**5**	5	4	5
	5		2	
4		3		
	1	1	4	

Recommender systems introduce feedback into the ratings matrix.

We propose an SVD-based method that deconvolves that effect* with one matrix.

ALG

* These are our assumptions!

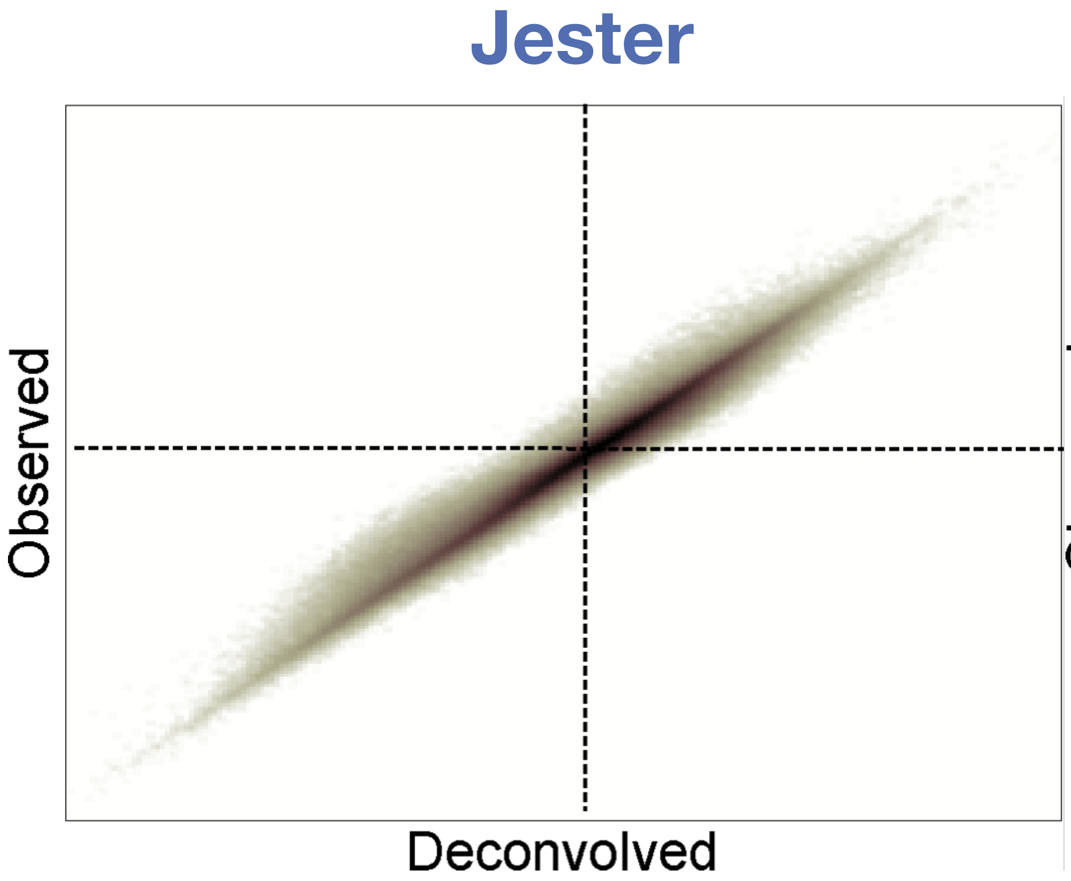


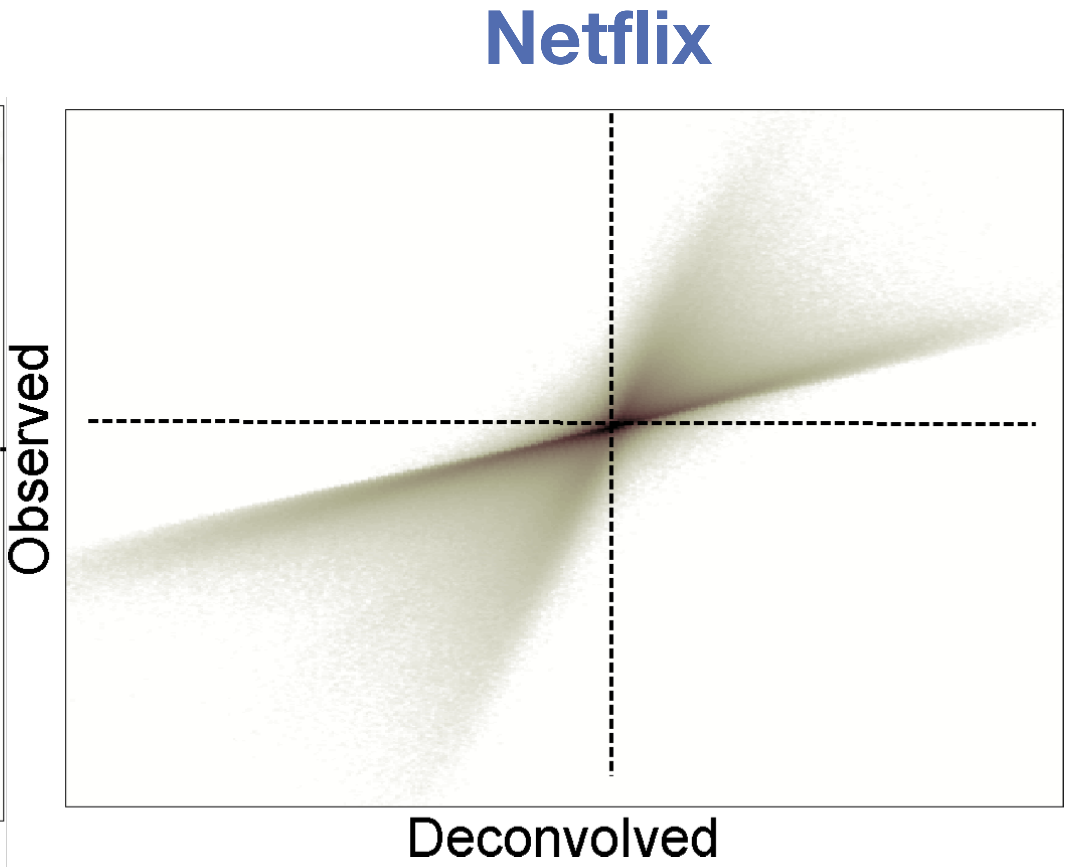
You believe that this model of a recommender is reasonable enough to study and we see how far the rabbit hole goes!

The talk ends, you believe -- whatever you want to.

Image from rockysprings, deviantart, CC share-alike

Real data shows two very different things for systems with recommenders and without.





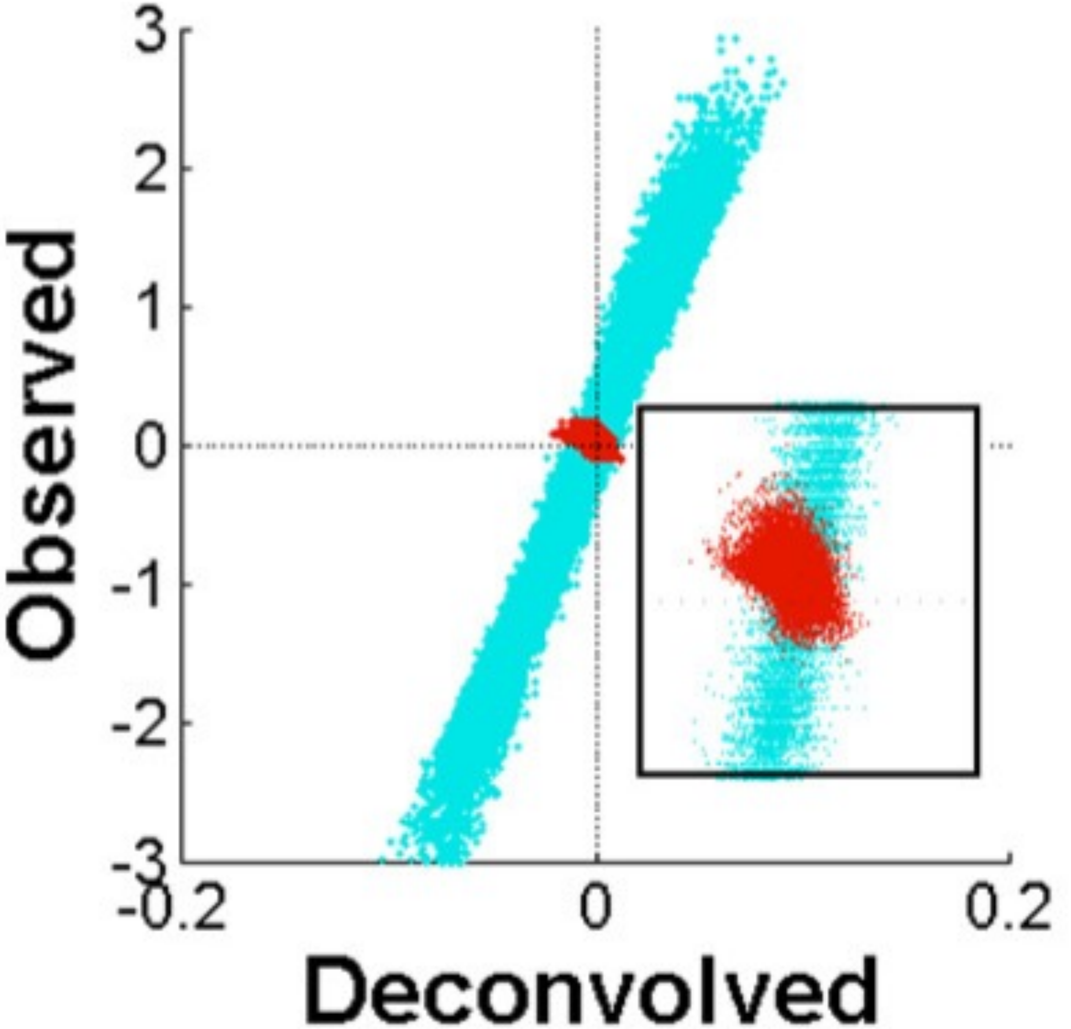
k = 1500

We create a synthetic recommender system to understand the impact of the feedback.

This is a synthetic model in the spirt of an item-response theory model

- A chosen set of true ratings are sampled initially.
- We randomly select from these for the initial observed matrix.
- We do 10 rounds of an item-similarity feedback recommender based on cosine similarity. At each step, users rate top-10 recommendations based on true values or recommended ratings.
- This allows us to track which entries were *caused* by the recommender vs. were true ratings.
- This was a few hundred users and a few hundred items.

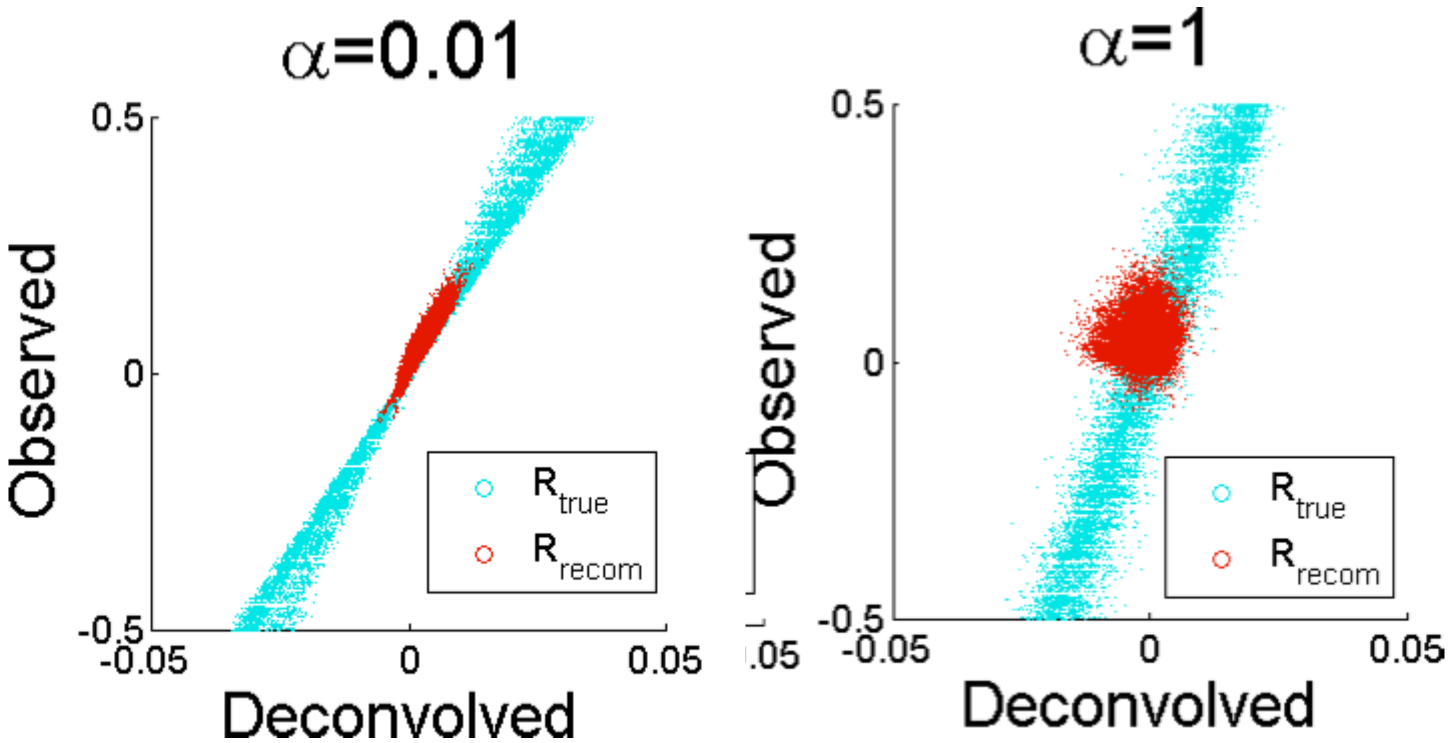
We create a synthetic recommender system to understand the impact of the feedback.



When we plot the deconvolved ratings matrix for the synthetic case, we see clear dispersion around the ratings that arose via the recommender system compared with those that were true.

Full SVD.

Using a large value of alpha highlights the recommender effects more clearly.

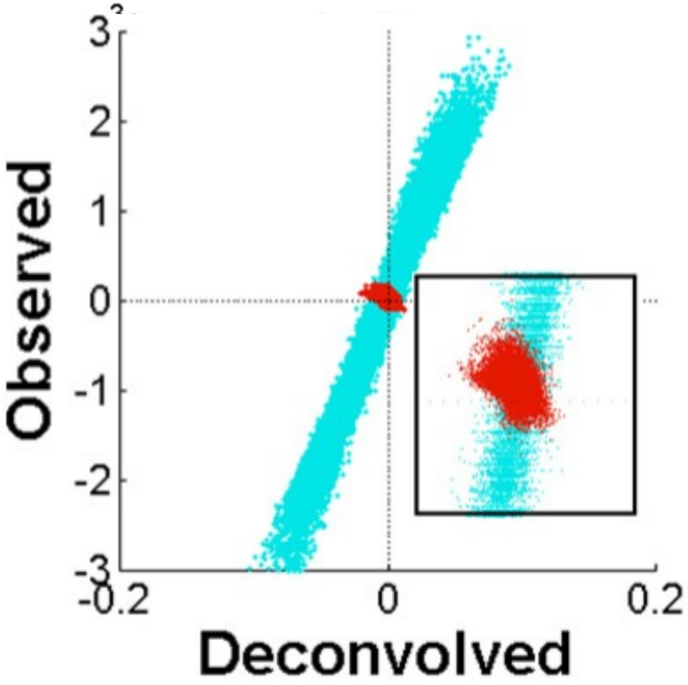


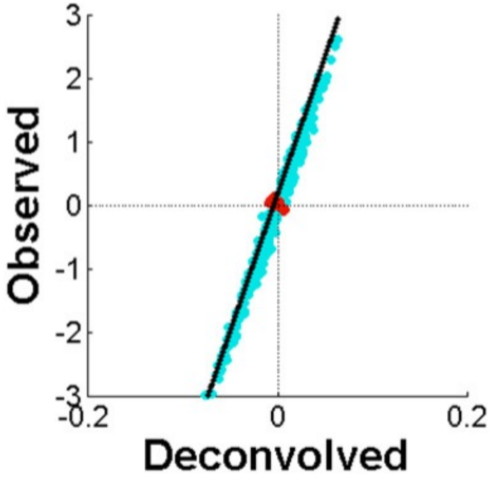
We are modeling the strength of the recommender system in α, so when we invert, we see the effect most strongly illustrated when α is large.

We always use $\alpha = 1$

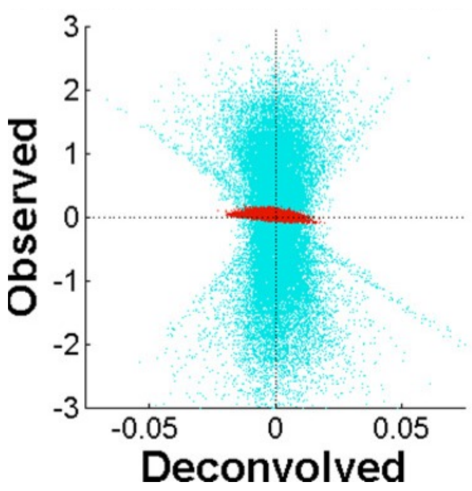
By cooking up a heuristic scoring scheme, we can identify these “skewed” items!

We transform the data to emphasize the skew.





For each item, we estimate a best fit line in the presence of outliers using the RANSAC method.

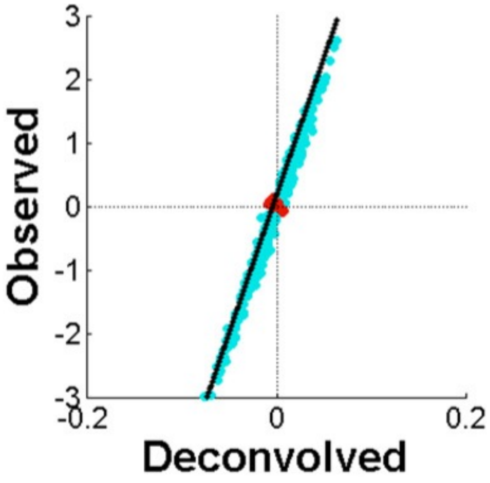


We translate the ratings so the line is the “y” axis.

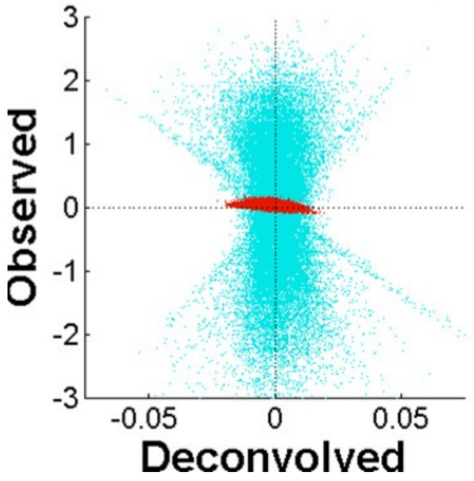
Deviations now show as projects on the “x” axis

By cooking up a heuristic scoring scheme, we can identify these “skewed” items!

We transform the data to emphasize the skew.

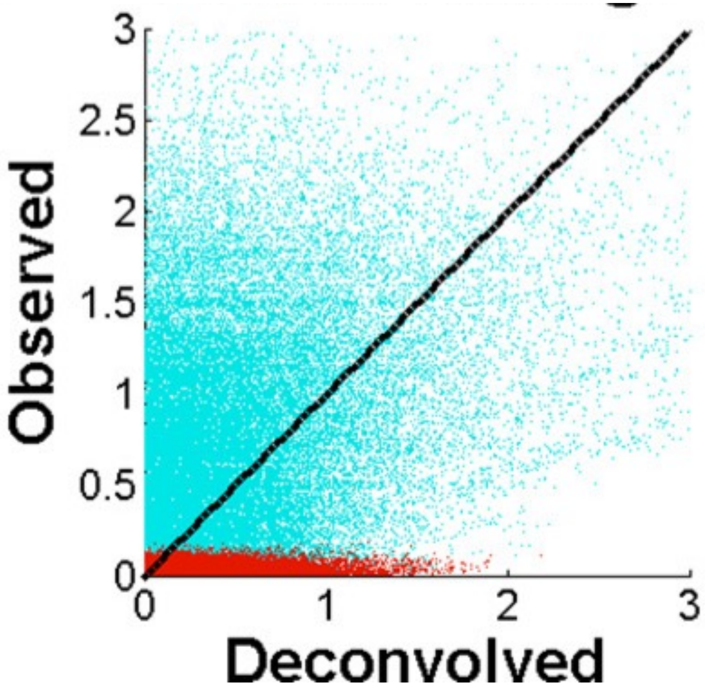


For each item, we estimate a best fit line in the presence of outliers using the RANSAC method.



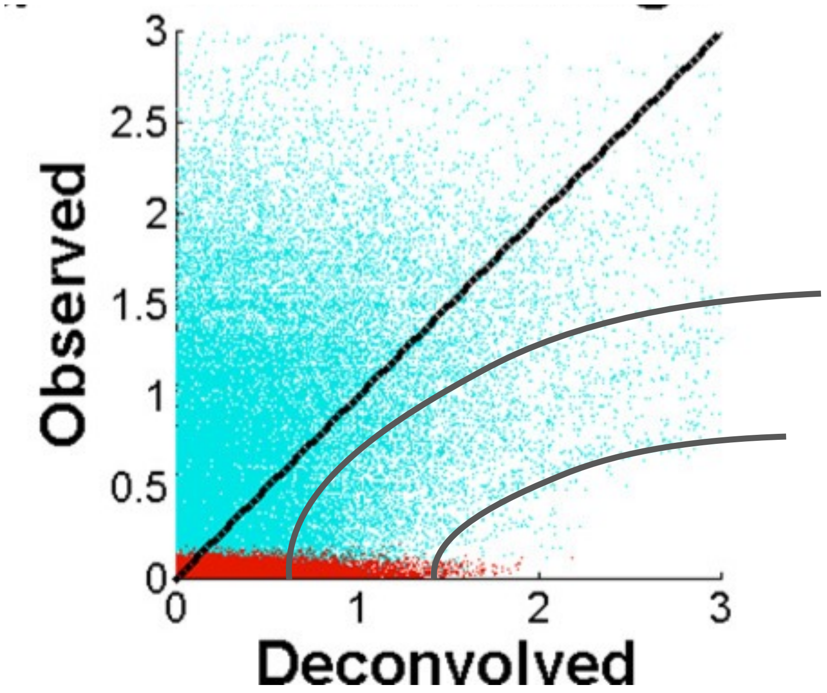
We translate the ratings so the line is the “y” axis.

Deviations now show as projects on the “x” axis

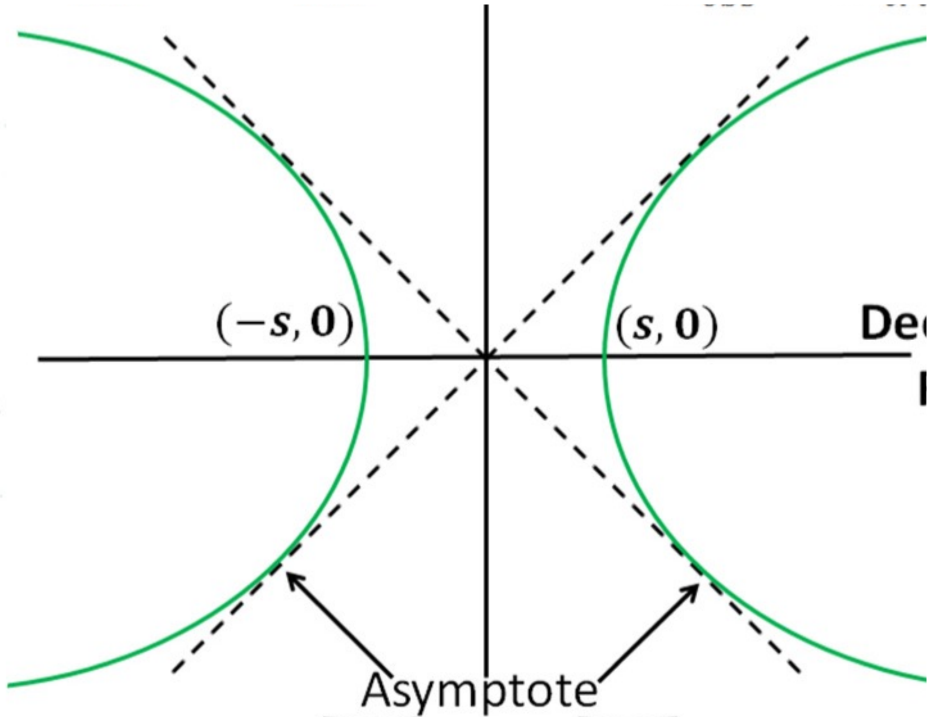


Finally, we take absolute values and scale to the same range

By cooking up a heuristic scoring scheme, we can identify these “skewed” items!



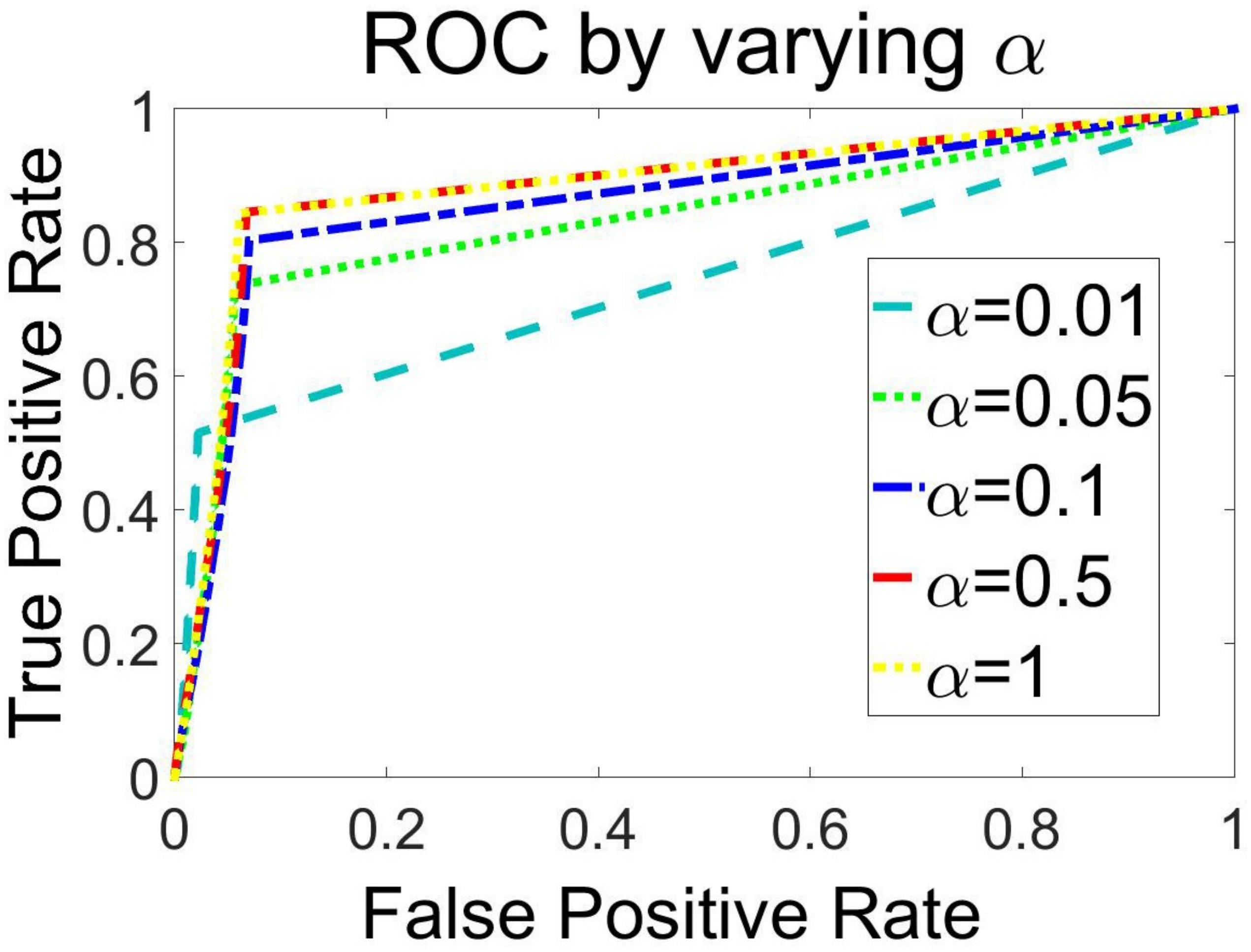
We look at fitting a hyperbola through the each rating with a unit slope at the fitting point.



$$\frac{(\overbrace{R_{true}})^2}{s^2} - \frac{(\overbrace{R_{obs}})^2}{(s\theta)^2} = 1$$

$$s(\breve{\boldsymbol{R}}_{\text{true}}, \breve{\boldsymbol{R}}_{\text{obs}}) = \text{real}(\sqrt{\breve{\boldsymbol{R}}_{\text{true}}^2 - \breve{\boldsymbol{R}}_{\text{obs}}^2})$$

The resulting score does pretty well at finding the influenced ratings.

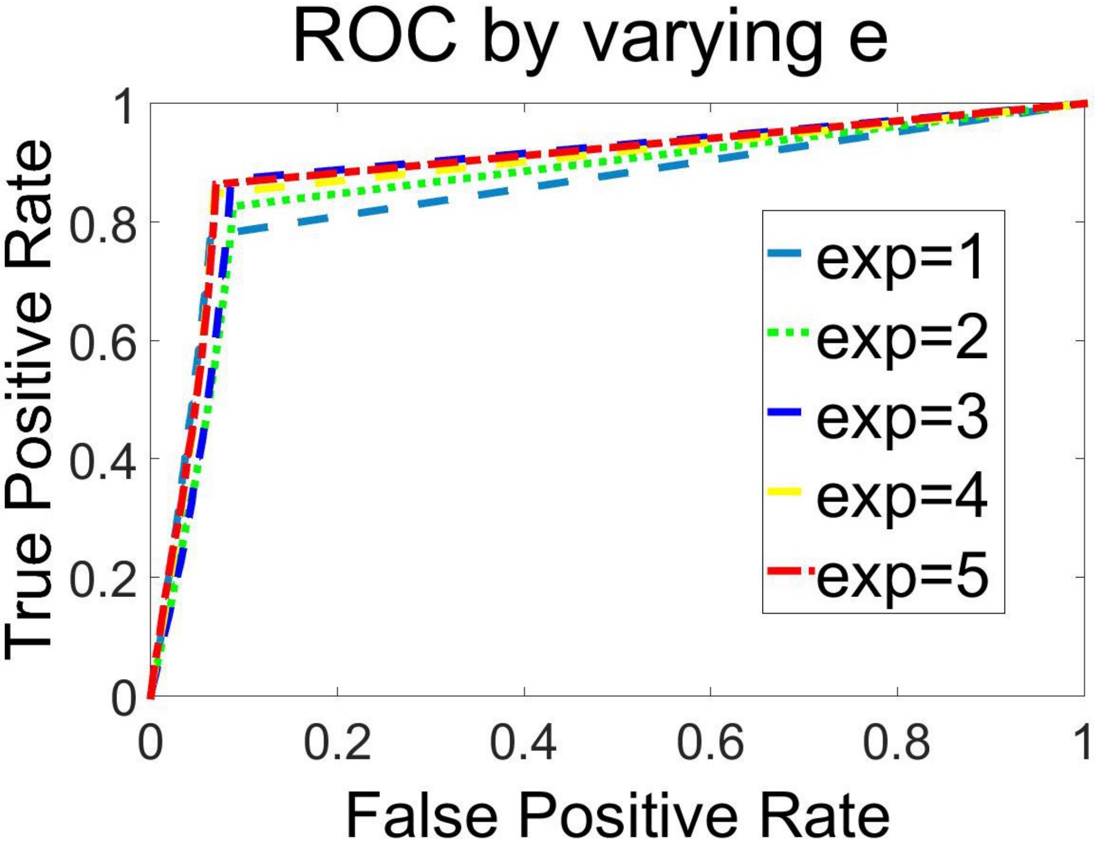


If the point in inside the hyperbola with intercept 0, then we give it score 0. Hence, the kink.

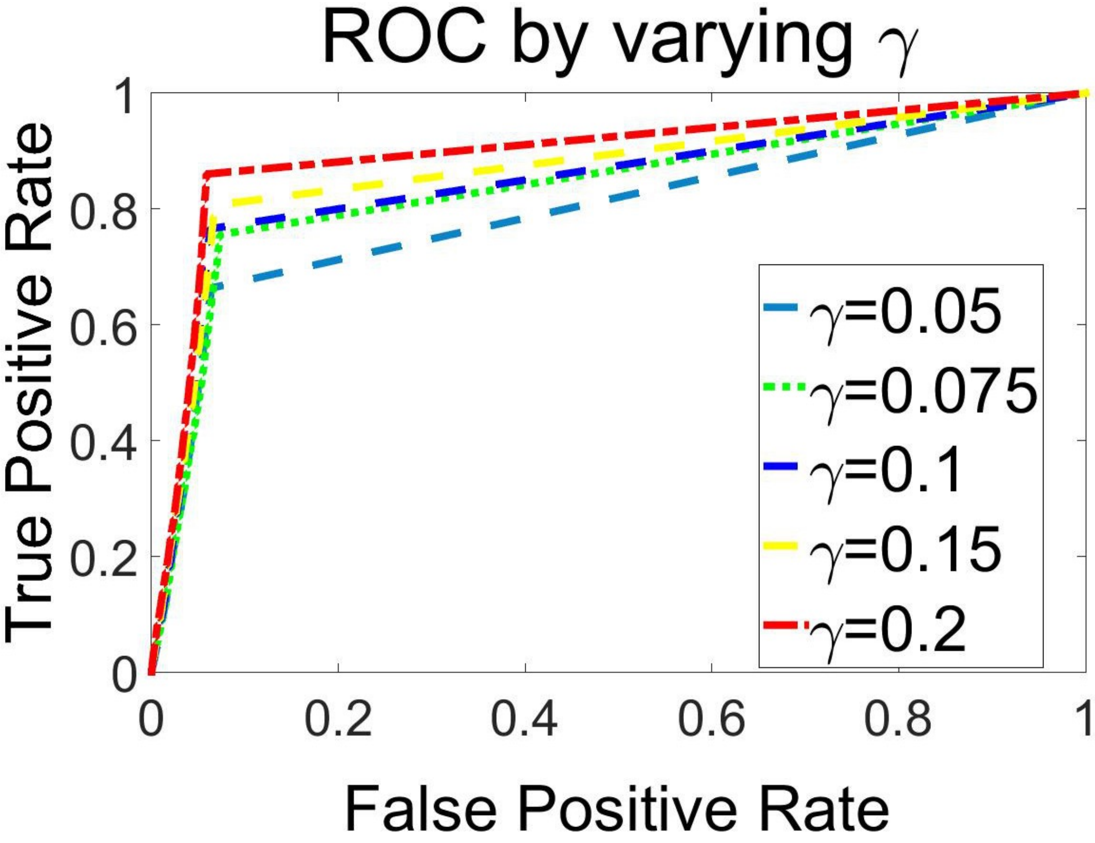
We again see better results with larger α

There results are largely the same if we vary parameters of the synthetic recommender.

e is the propensity to accept a recommendation



γ is the fraction of initial ratings

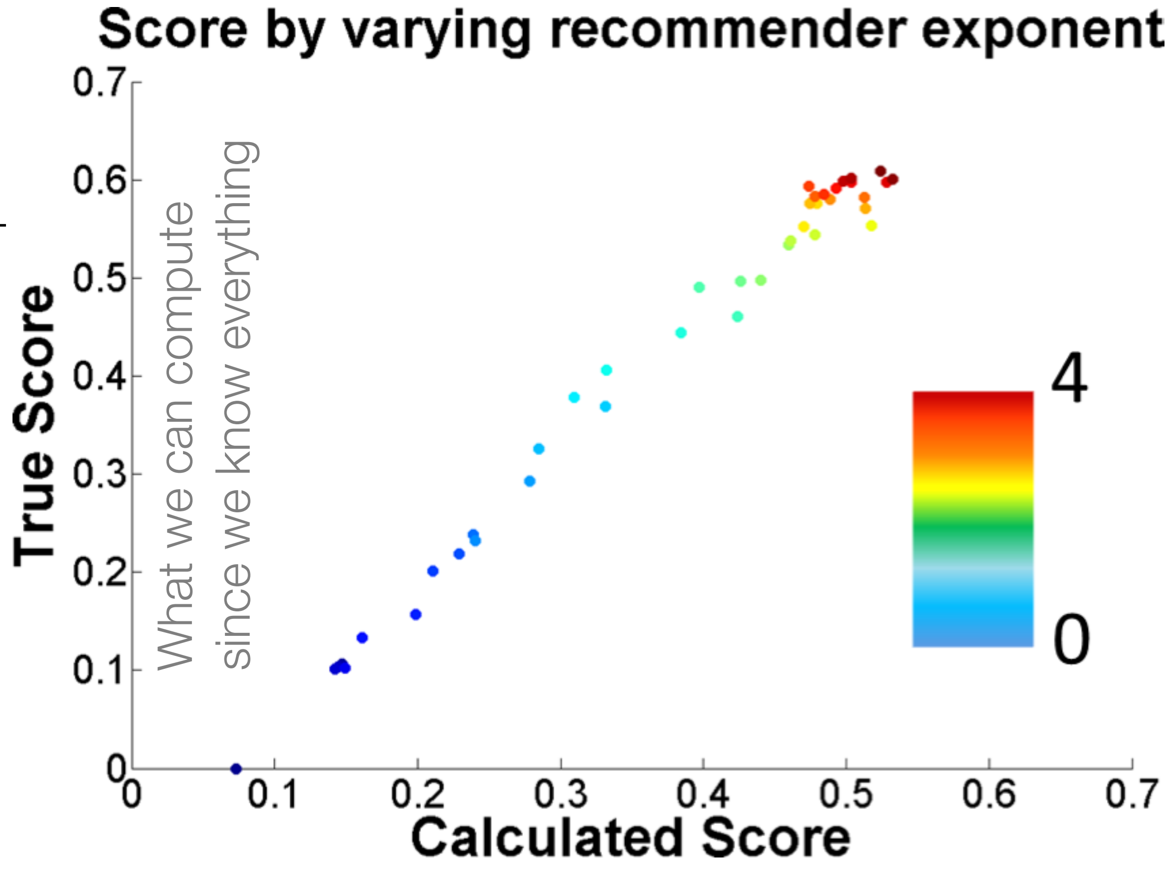


We can get an overall estimate of the effect of the recommender by summing these scores.

$$\frac{\displaystyle\sum_{r \in \text{Obs. Ratings}} \begin{cases} 1 & s(r_{\text{true}}, r_{\text{obs}}) > 0 \\ 0 & \text{otherwise.} \end{cases}}{\text{num obs. ratings}}$$

Our system score is the fraction of ratings where we see recommender effects to any degree.

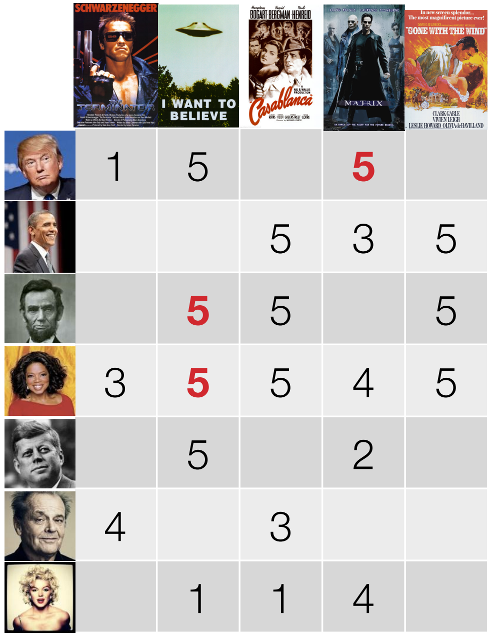
For the synthetic case, as we vary the recommender strength e, we can look at the scores.



What we could compute on a real system

Summary of results

Recommender systems introduce feedback into the ratings matrix.

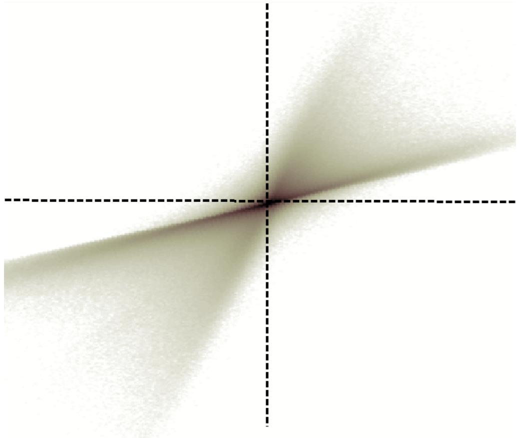

SCHWARZENEGGER
I WANT TO BELIEVE
BOGART BERGMAN HENREID
Casablanca
MATRIX
"GONE WITH THE WIND"

1	5		5	
		5	3	5
	5	5		5
3	5	5	4	5
	5		2	
4		3		
	1	1	4	

We propose an SVD-based method that deconvolves that effect* with one matrix.

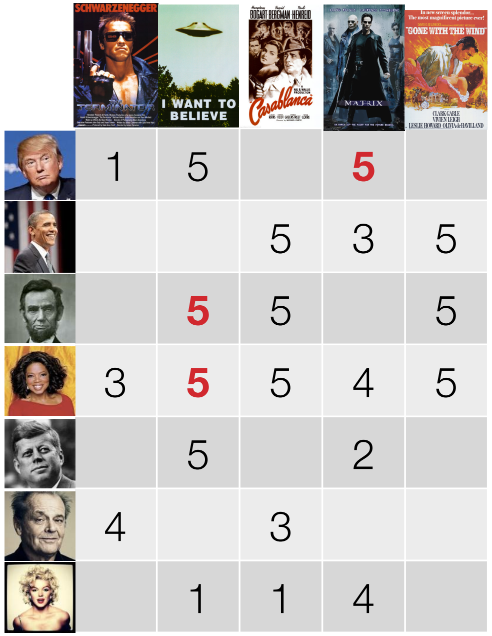
ALG

Skew in deconvolved vs. given ratings scatter-plot scores rec. effects.



Summary of results

Recommender systems introduce feedback into the ratings matrix.



1	5		5	
		5	3	5
	5	5		5
3	5	5	4	5
	5		2	
4		3		
	1	1	4	

We propose an SVD-based method that deconvolves that effect* with one matrix.

ALG

Skew in deconvolved vs. given ratings scatter-plot scores rec. effects.

It also scores system rec. effects

Dataset	Score
Jester	0.0487
MusicLab-Weak	0.1073
MusicLab-Strong	0.1509
MovieLens-10M	0.3821
BeerAdvocate	0.2223
Fine Foods	0.1209
Netflix	0.2661

Case studies with our method on real data!

Dataset	Users	Items	Rating
Jester-1	24.9K	100	615K
Jester-2	50.6K	140	1.72M
MusicLab-Weak	7149	48	25064
MusicLab-Strong	7192	48	23386
MovieLens-100K	943	603	83.2K
MovieLens-1M	6.04K	2514	975K
MovieLens-10M	69.8K	7259	9.90M
Netflix	480K	16795	100M
BeerAdvocate	31.8K	9146	1.35M
RateBeer	28.0K	20129	2.40M
Fine Foods	130K	5015	329K
Wine Ratings	21.0K	8772	320K

Joke ratings collected with an experimental design

Music ratings collected with varying system feedback effets (but no recommender system)

Music ratings collected with varying system feedback effets (but no recommender system)

The 100M netflix data

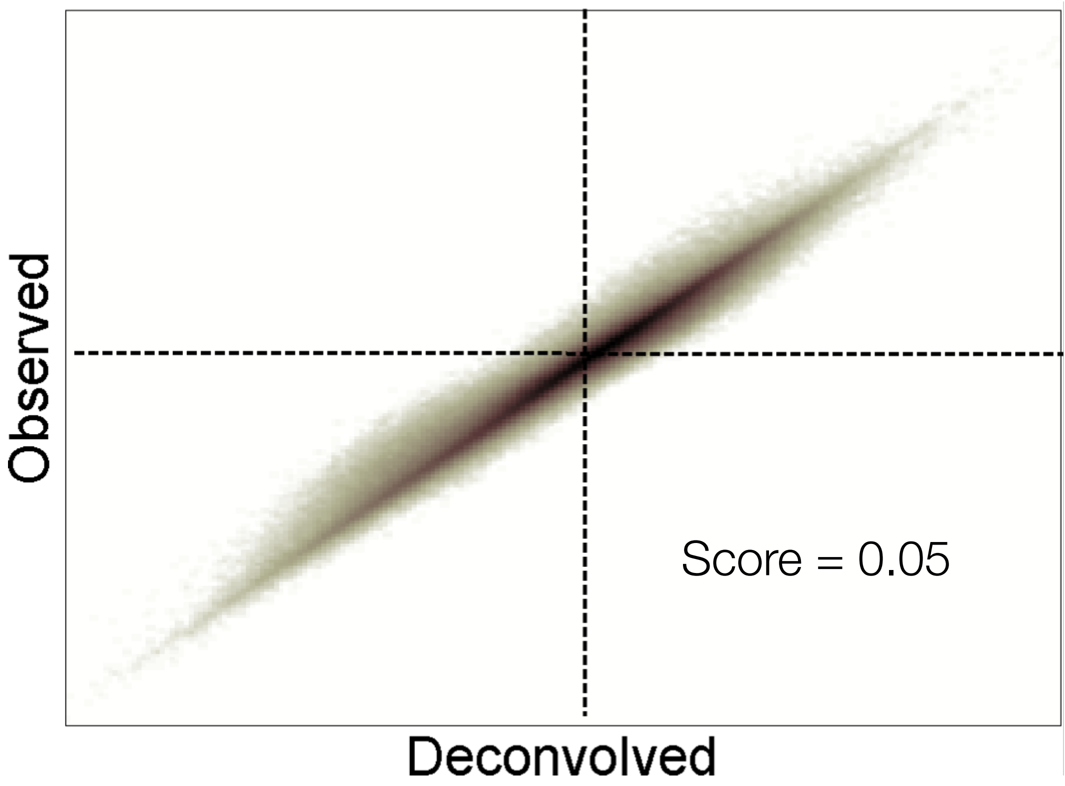
Another large set of recommender system data from SNAP and various website with no explicit recommenders

Dataset	Users	Items	Rating
Jester-1	24.9K	100	615K
Jester-2	50.6K	140	1.72M
MusicLab-Weak	7149	48	25064
MusicLab-Strong	7192	48	23386
MovieLens-100K	943	603	83.2K
MovieLens-1M	6.04K	2514	975K
MovieLens-10M	69.8K	7259	9.90M
Netflix	480K	16795	100M
BeerAdvocate	31.8K	9146	1.35M
RateBeer	28.0K	20129	2.40M
Fine Foods	130K	5015	329K
Wine Ratings	21.0K	8772	320K

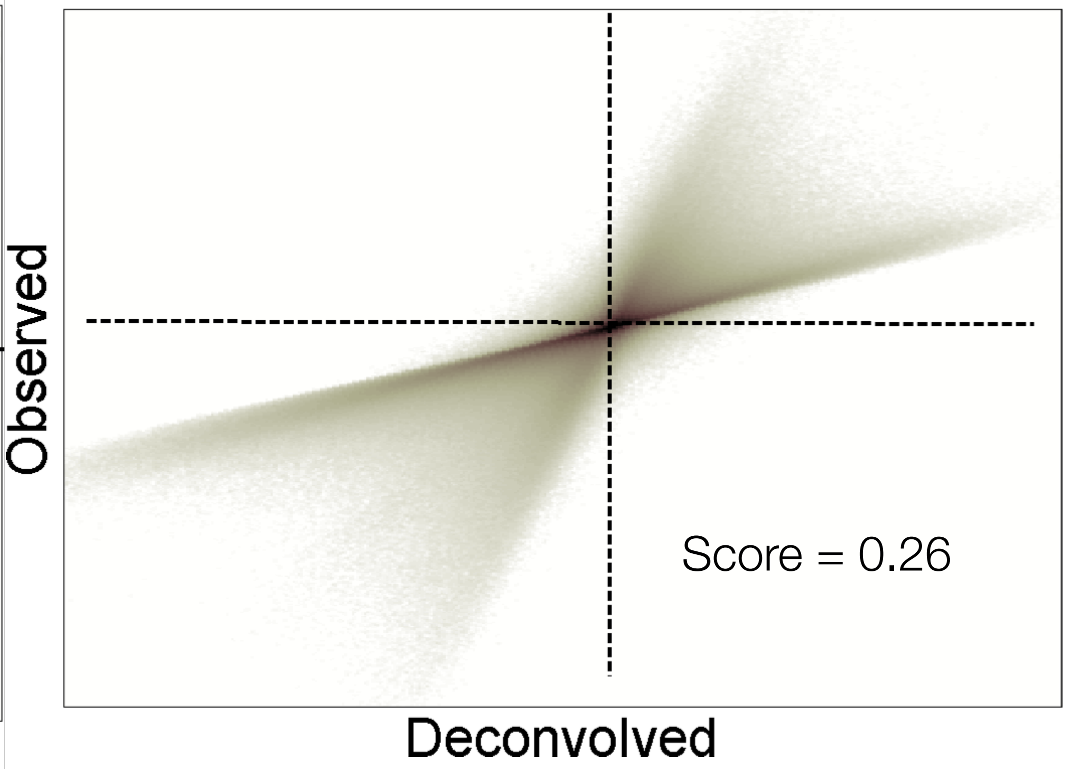
* We do remove users with few ratings.

Real data shows two very different things for systems with recommenders and without.

Jester

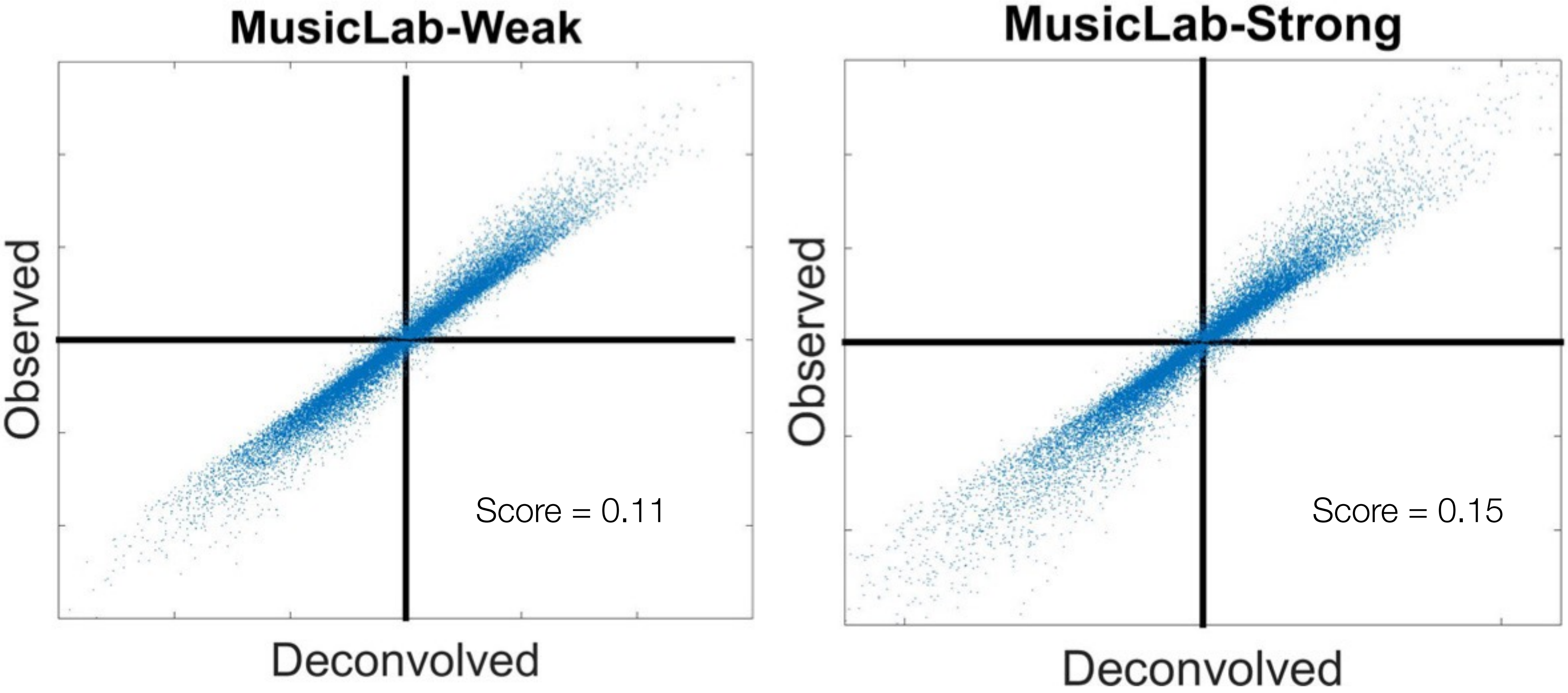


Netflix



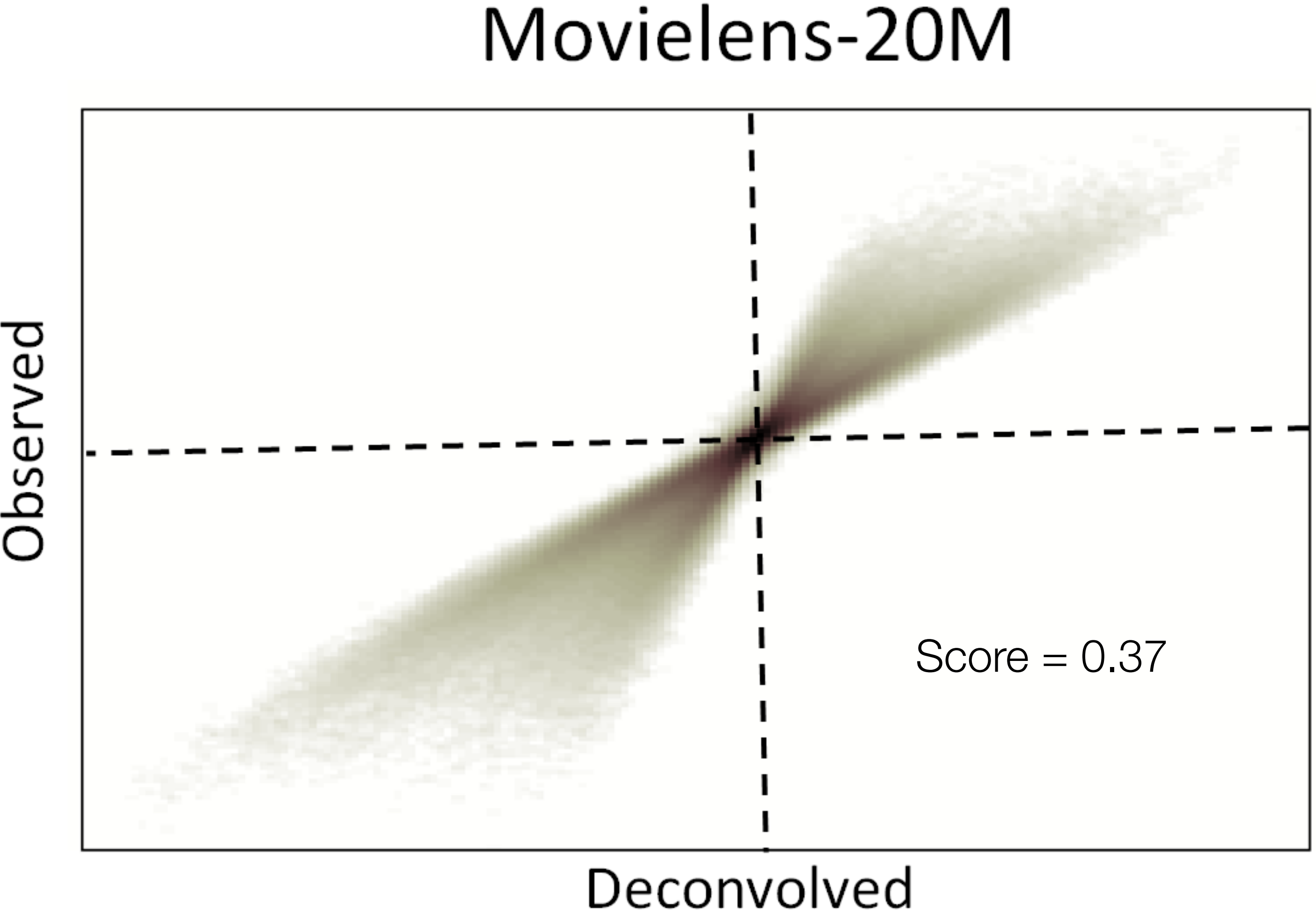
k = 1500

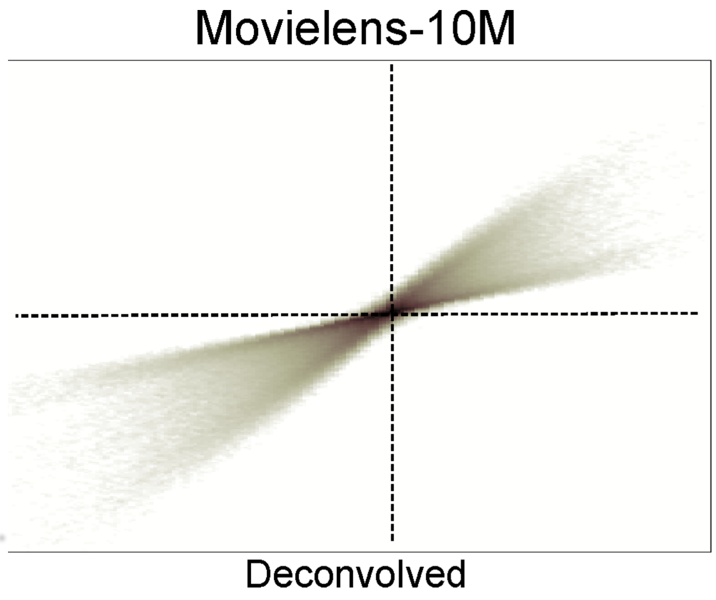
In the MusicLab experiment, we see more dispersion with the feedback scenario.



We used the full SVD

We see increasing effects for MovieLens over time as the number of ratings grows.



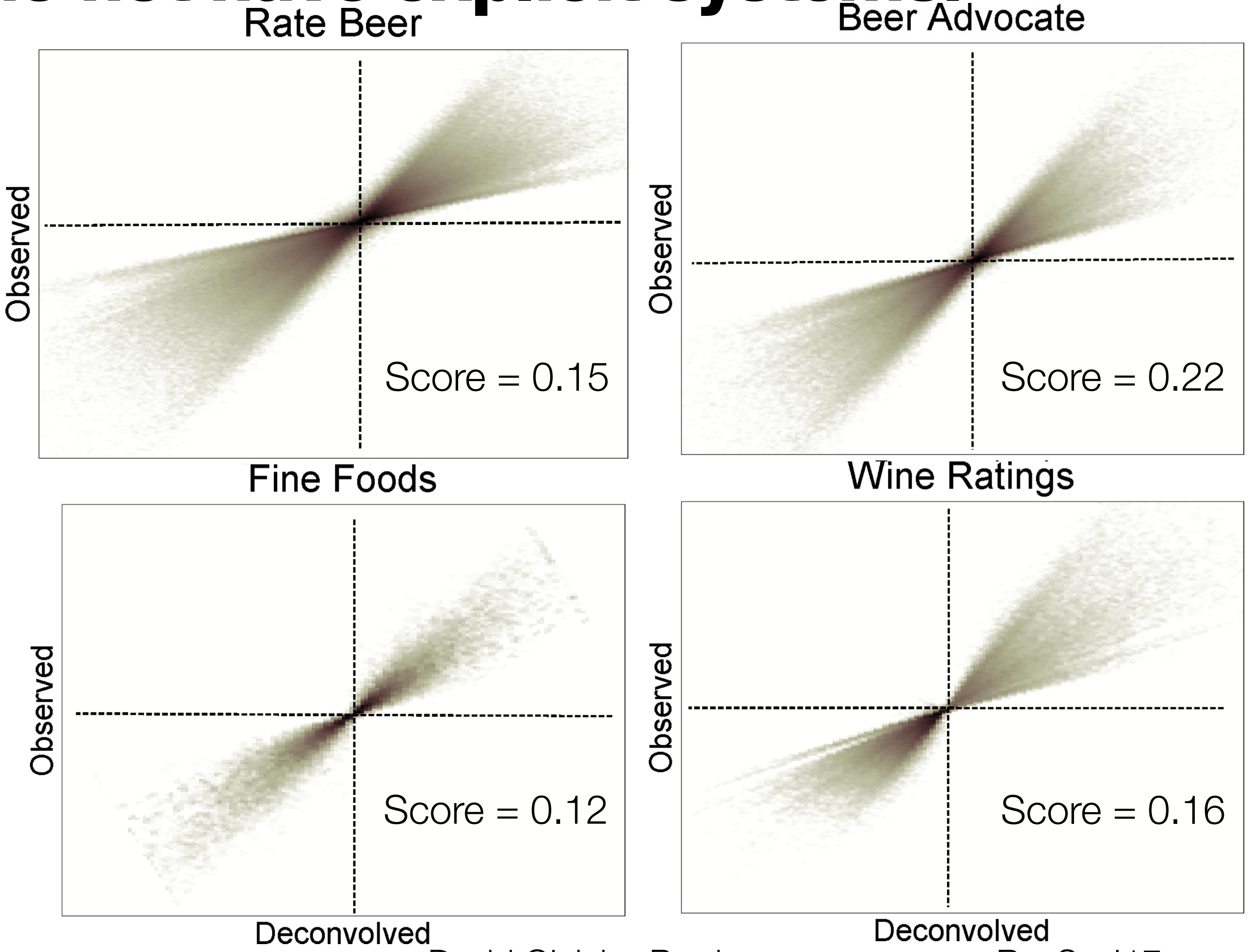


Score = 0.38

We see varying recommender effects even in systems that do not have explicit systems.

Recall that our model is recommended + true. So any feedback effects will be called recommender effects.

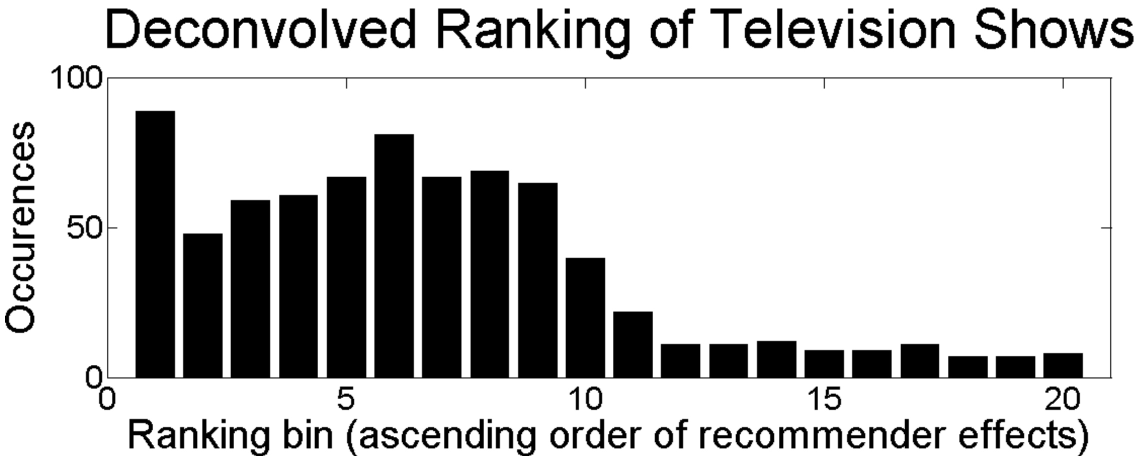
These systems may have other forms of feedback that we are sensitive too.



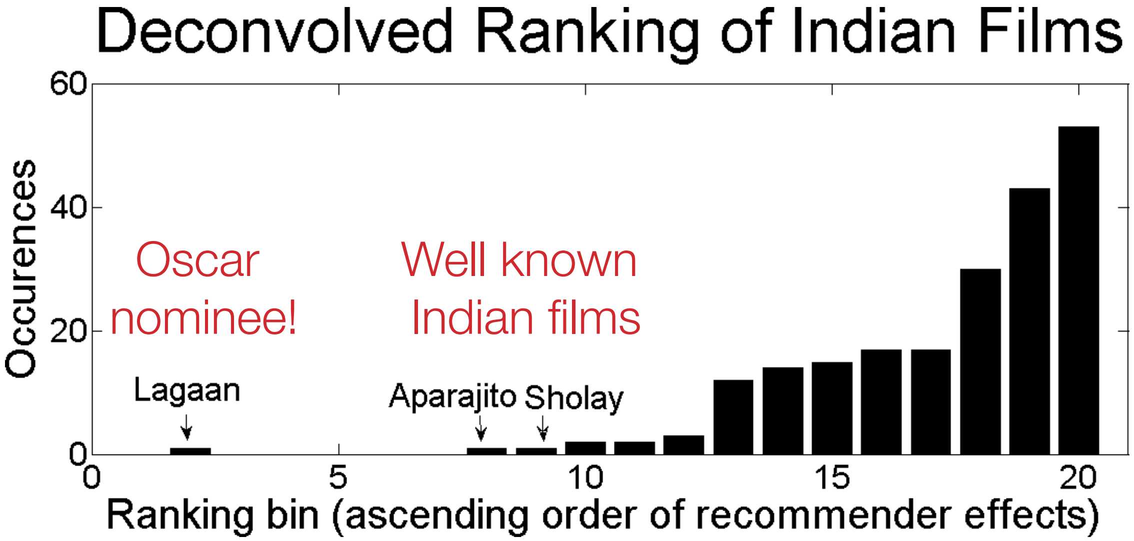
Looking at the individual scores shows that obscure movies are subject to feedback.

In the Netflix data, TV shows show low recommender effects.

Within TV, Season 1 is more recommended.


Deconvolved Ranking of Television Shows

Obscure Indian movies show high recommender system effects.


Deconvolved Ranking of Indian Films

There is a ton of future work if people want to follow up on this!

Questions for real data

- Are the deconvolved ratings were more *useful* in producing recommendations.
- How accurate are we at detecting these feedback loops based on logs of which items are recommended?

Tractable theory & practice relaxations

- What if we are given a similarity matrix ***S***?
- Can we quantify how similar the norms need to be?
- Can this same thing be done for a low-rank model of a recommender?
- What about for general active learning scenarios?

Paper

Sinha, Gleich, Ramani.
Deconvolving Feedback Loops in Recommender Systems, NIPS 2016

`arXiv:1703.01049`

Code

```
https://github.com/sinhayan/
Deconvolving_Feedback_Loops
```

Thanks!
(and ask about Music!)