Demos for the next session

https://s3.amazonaws.com/GraphLab-Datasets/demos/recommendation-systems.ipynb

https://s3.amazonaws.com/GraphLab-Datasets/demos/matrix-factorization-demo.ipynb

https://s3.amazonaws.com/GraphLab-Datasets/demos/text-analysis.ipynb

Survey:

https://www.surveymonkey.com/s/GraphLab2014TrainingDay

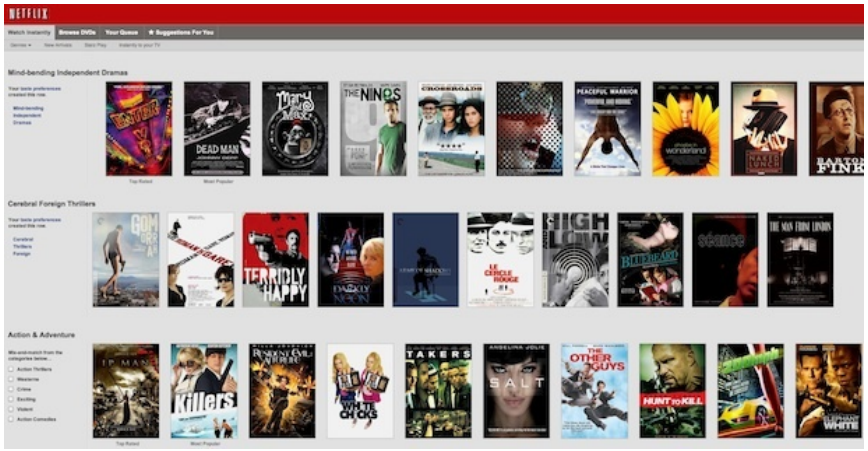
Recommendation systems and text analysis with GraphLab Create

Outline

- Recommendation systems
  - Background
  - Computing item similarities
  - Matrix factorization methods
- Text analysis
  - Munging and preprocessing
  - Finding similar documents
  - Topic modeling

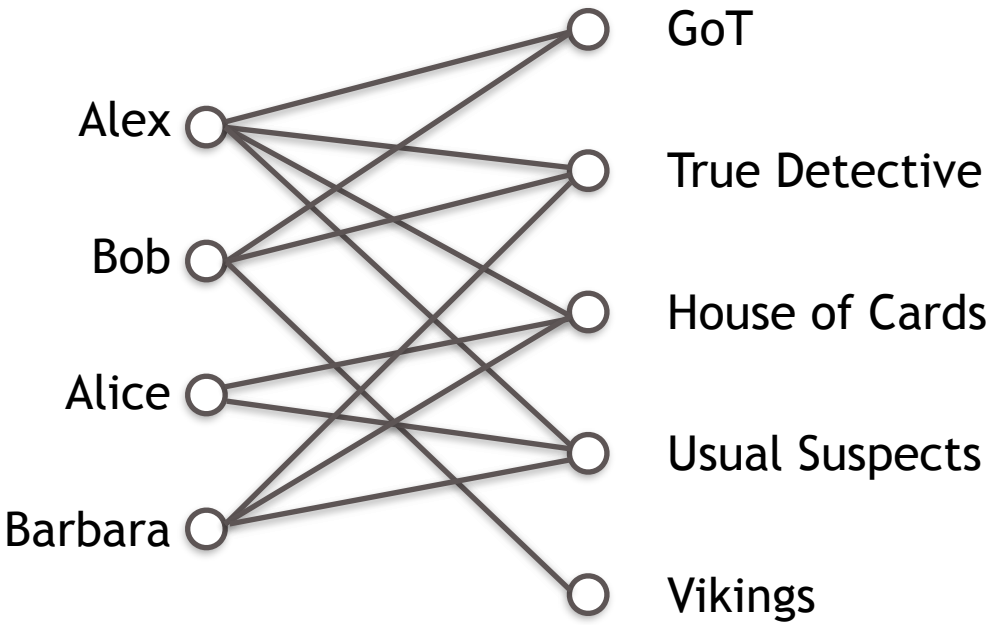
Recommendation systems with GraphLab Create

Why recommendation systems?



user_id	item_id
Alex	*Game of Thrones*
Alex	*True Detective*
Alex	*House of Cards*
Alex	*Usual Suspects*
Bob	*Game of Thrones*
Bob	*True Detective*
Bob	*Vikings*
Alice	*Game of Thrones*
Alice	*True Detective*
…	…

user_id	item_id
Alex	*Game of Thrones*
Alex	*True Detective*
Alex	*House of Cards*
Alex	*Usual Suspects*
Bob	*Game of Thrones*
Bob	*True Detective*
Bob	*Vikings*
Alice	*Game of Thrones*
Alice	*True Detective*
…	…

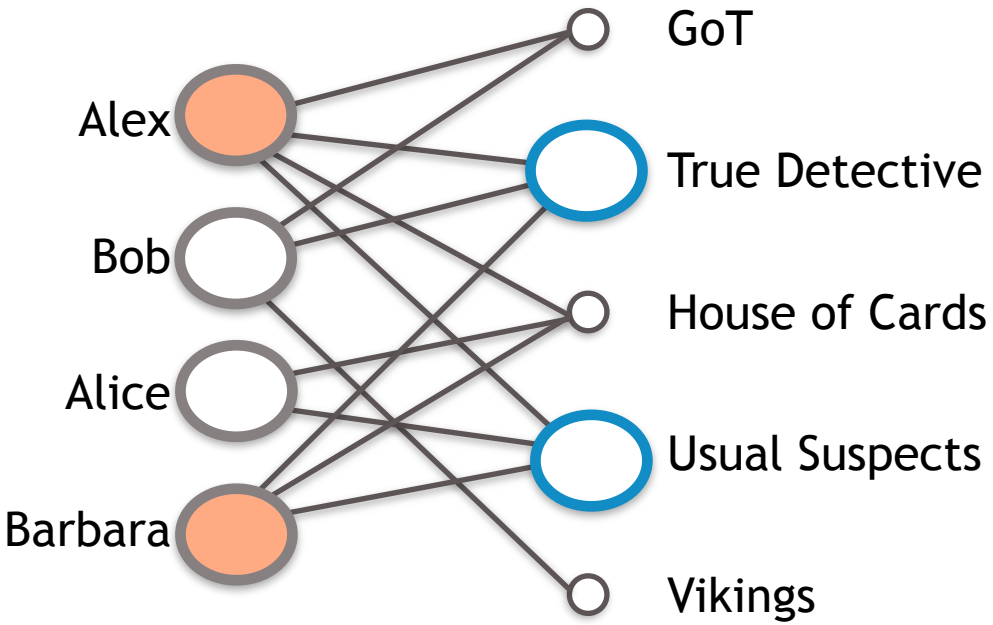


SFrame ⟷ SGraph

user_id	item_id
Alex	*Game of Thrones*
Alice	*True Detective*
…	…

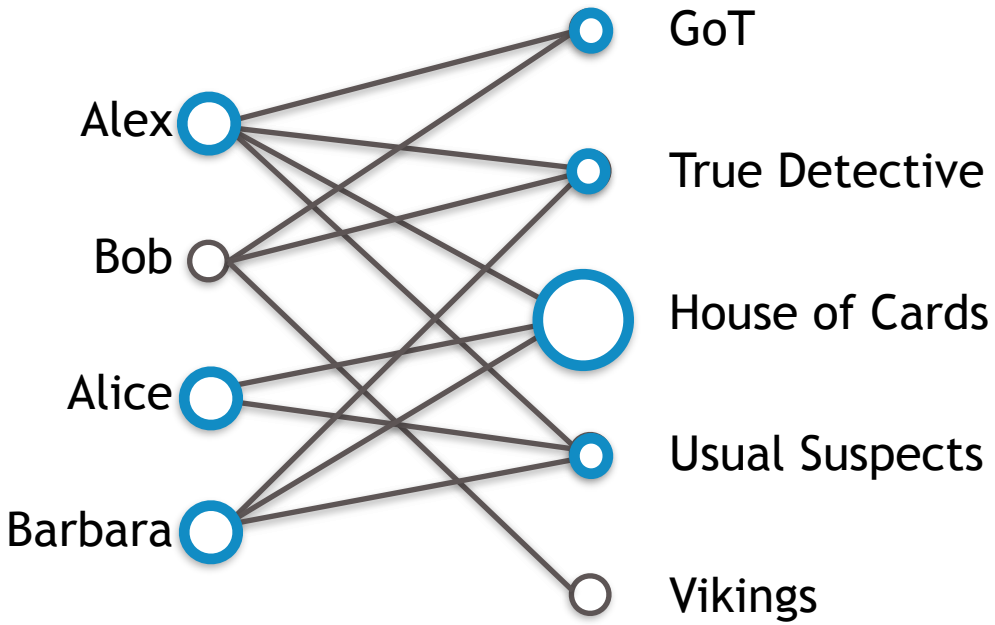
Similarity between
True Detective and Usual Suspects:

$$\frac{\#\text{ who watched both}}{\#\text{ who watched either}} = \frac{2}{4}$$



user_id	item_id
Alex	*Game of Thrones*
Alice	*True Detective*
…	…

For each item:
- Accumulate statistics about the number of users in common
- Rank top 100 nearest items



Creating a recommendation system in GraphLab Create

```
>>> import graphlab
>>> m = graphlab.recommender.create(data)
>>> recs = m.recommend()
```

Getting recommendations for a set of users

```
>>> r = m.recommend(users=my_user)
```

Restricting recommendations to a particular set of items

```
>>> r = m.recommend(items=candidates)
```

Excluding previously seen observations

```
>>> r = m.recommend(exclude=ignore_these)
```

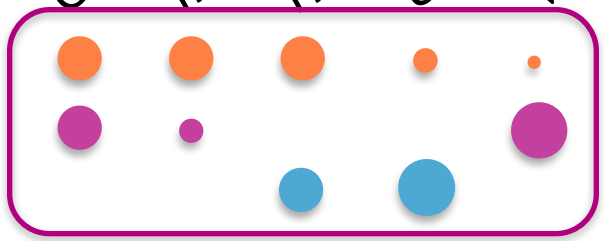
Demo time!

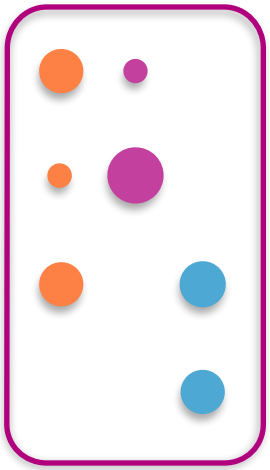
user_id	item_id	rating
Alex	*Game of Thrones*	*5*
Alex	*True Detective*	*5*
Alex	*House of Cards*	*5*
Alex	*Usual Suspects*	*3*
Bob	*Game of Thrones*	*5*
Bob	*True Detective*	*4*
Bob	*Vikings*	*5*
Alice	*Game of Thrones*	*1*
Alice	*True Detective*	*5*
…	…	

	Game of Thrones	True Detective	House of Cards	Usual Suspects	Vikings
Alex	*5*	*5*	*5*	*3*	
Bob	*5*	*4*			*5*
Alice	*1*		*5*	*4*	
Barbara		*3*	*5*	*5*	

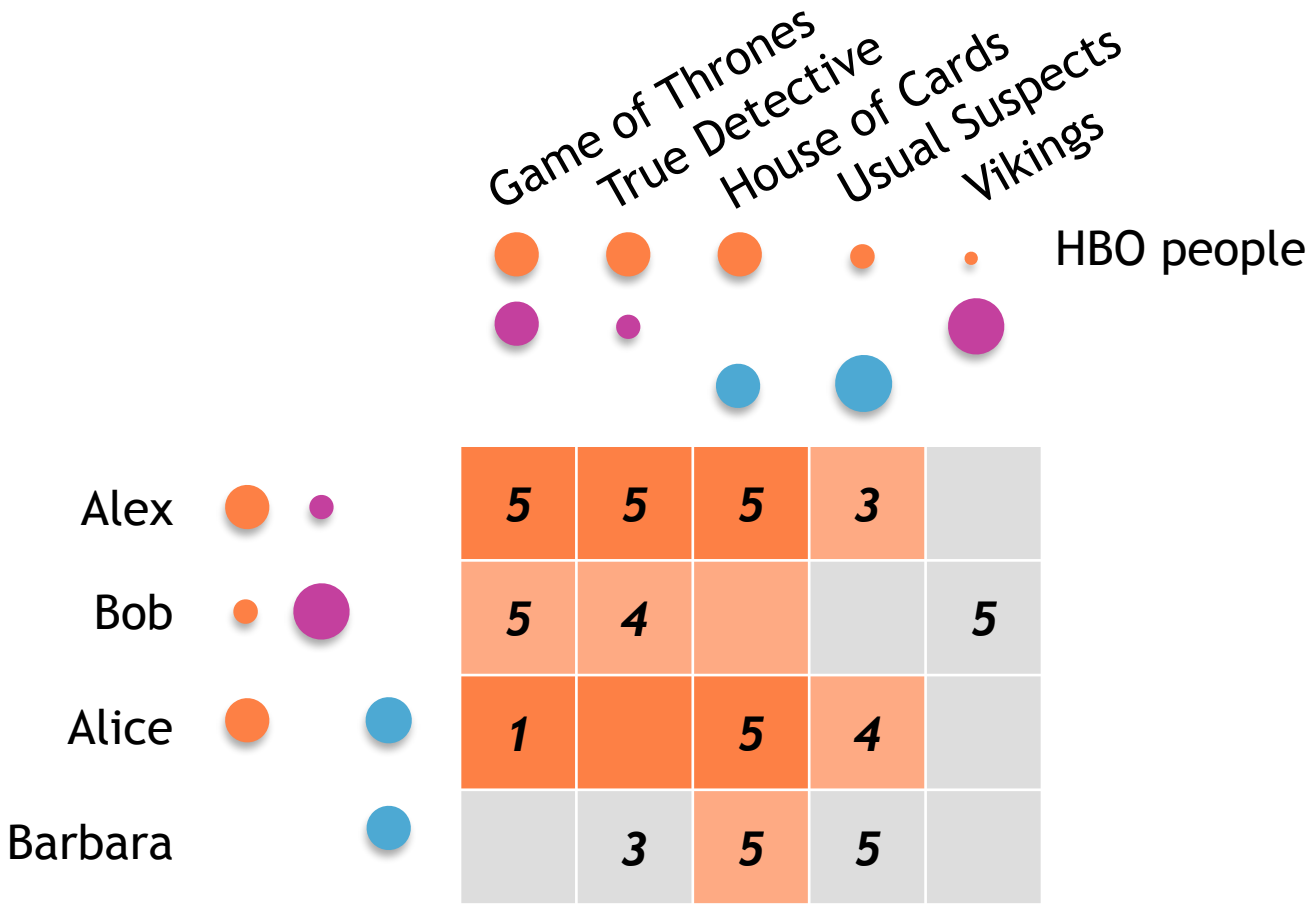
	Game of Thrones	True Detective	House of Cards	Usual Suspects	Vikings
Alex	*5*	*5*	*5*	*3*	
Bob	*5*	*4*			*5*
Alice	*1*		*5*	*4*	
Barbara		*3*	*5*	*5*	

Model parameters





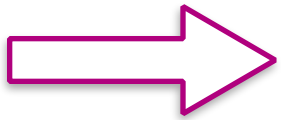
Alex	5	5	5	3	
Bob	5	4			5
Alice	1		5	4	
Barbara		3	5	5	

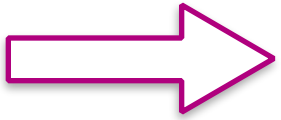

Game of Thrones
True Detective
House of Cards
Usual Suspects
Vikings
HBO people
Alex
Bob
Alice
Barbara
5 5 5 3
5 4 5
1 5 4
3 5 5

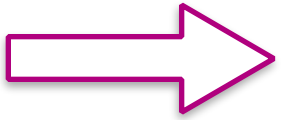

Game of Thrones
True Detective
House of Cards
Usual Suspects
Vikings
HBO people
Violent historical
Alex
Bob
Alice
Barbara
5 5 5 3
5 4 5
1 5 4
3 5 5


Game of Thrones
True Detective
House of Cards
Usual Suspects
Vikings
HBO people
Violent historical
Kevin Spacey fans
Alex
Bob
Alice
Barbara
5 5 5 3
5 4 5
1 5 4
3 5 5

Matrix factorization: Extensible

Side features  factorization_machine

Ranking  unobserved_rating

Overfitting  regularization

```
from graphlab import recommender
recommender.create(data,
                   method='matrix_factorization',
                   n_factors=20)
```

Demo!

Text analytics

Text

- Data often has free-form text
  - Reviews of movies, restaurants, etc.
  - Email, tweets, etc.
- Hard to include in automated analysis
  - Hand-crafted features are not ideal

Tools for common tasks

- SFrames help with typical cleaning tasks
- Method for computing “bag-of-words”
- TF-IDF: discount common words
- Topic modeling
- More to come!

Topic Models

- Statistical model of text that assumes a document collection can be explained by a small set of topics.

The burrito was terrible. I…
Sometimes sushi here …
The waiters never came until…
When you need gyoza, you…
My favorite place ever! You…

Topic Models

- Statistical model of text that assumes a document collection can be explained by a small set of topics.

The burrito was terrible. I…
Sometimes sushi here …
The waiters never came until…
When you need gyoza, you…
My favorite place ever! You…

Demo

Create scalable data products fast in Python

Got questions? Join our community at graphlab.com