



ClusterCatch: Spotting Groups of Suspicious Nodes in an Amazon Customer Graph

Veena Padmanabhan
@veenapad
Applied ML Scientist | TRMS

Yan Zhang
@zhngyn
Principal Scientist | TRMS

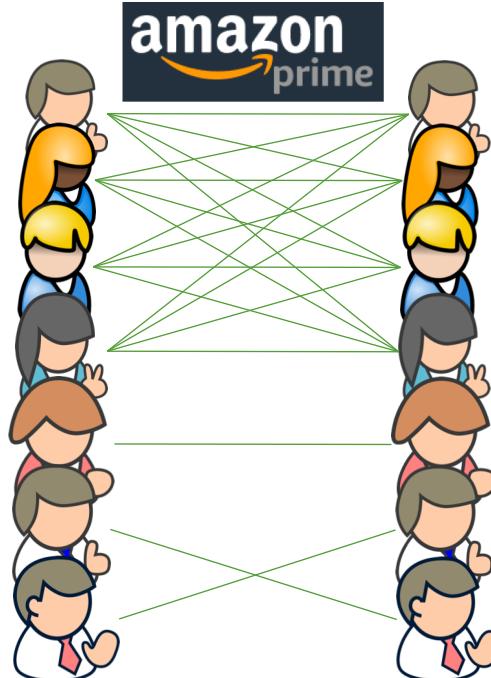
Radu Drossu
@drossu
Director | TRMS

Christos Faloutsos
@faloutso
Professor | CMU, Product Graph

Bad Actors Create Multiple Colluding Accounts

Prime Free-Trial

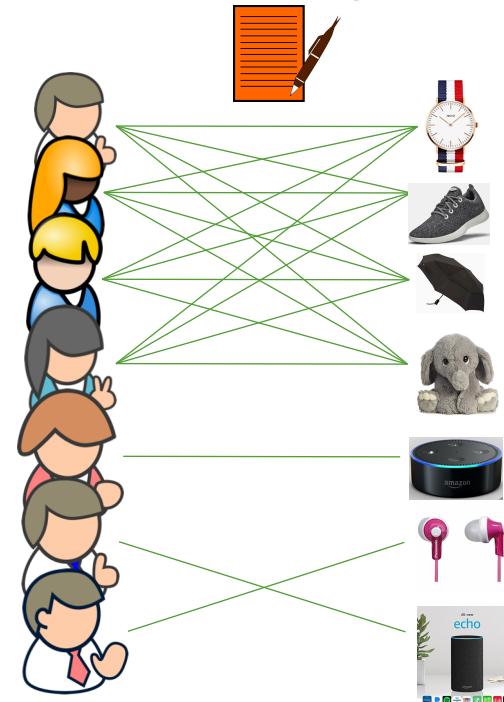
FT customers that are similar to each other



~ Tens of millions USD per year

Review Abuse

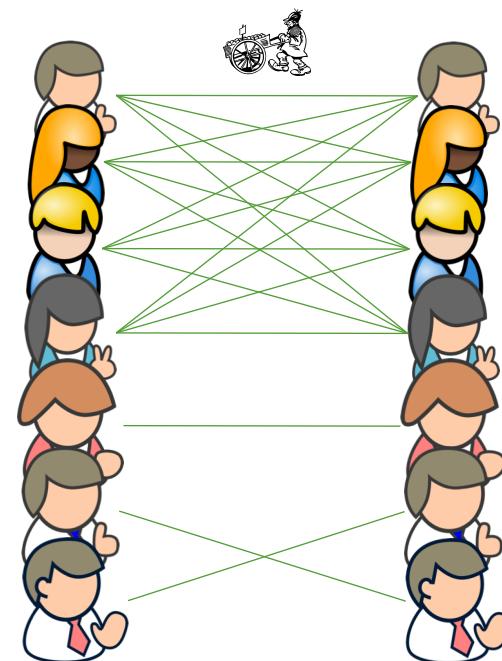
Customers & what products they reviewed



Customer Experience Impact

Reseller

Customers that are similar to each other

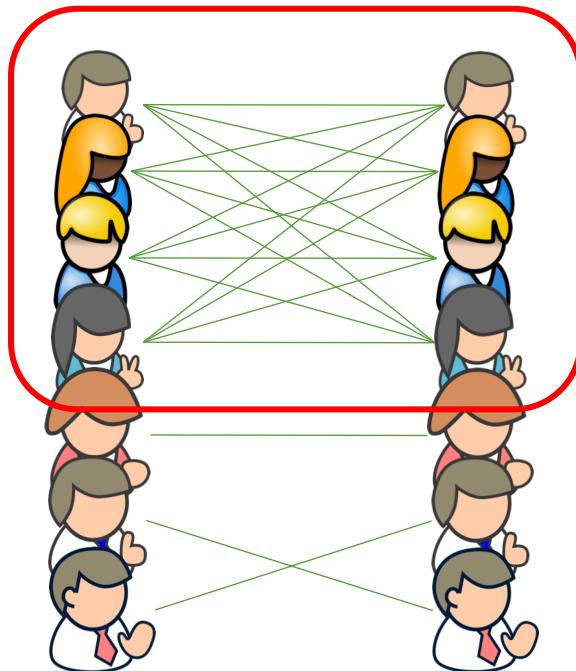


Products under promotion may not reach intended customers

Bad Actors Create Multiple Colluding Accounts

Prime Free-Trial

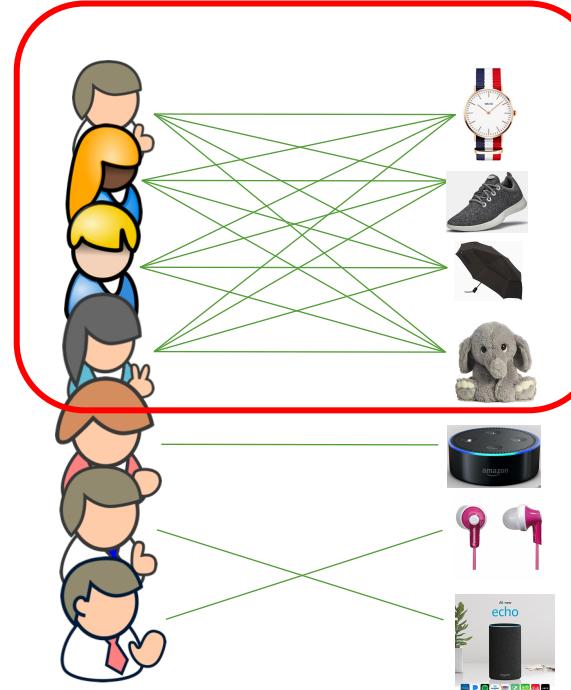
FT customers that are similar to each other



~ Tens of millions USD per year

Review Abuse

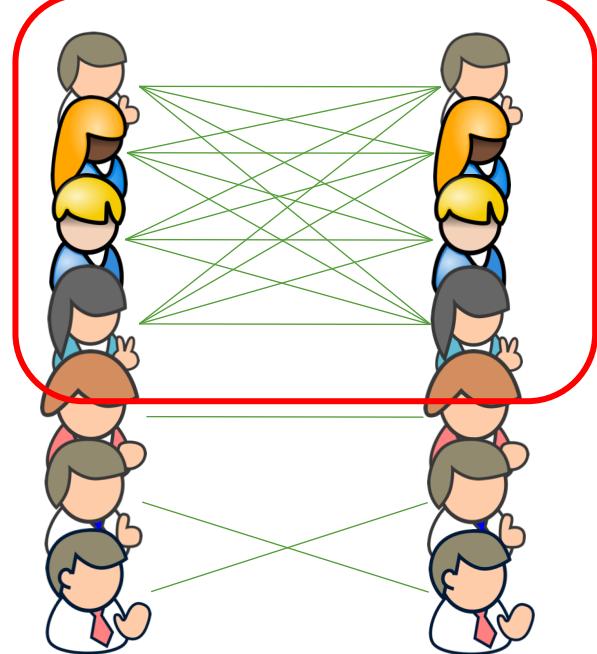
Customers & what products they reviewed



Customer Experience Impact

Reseller

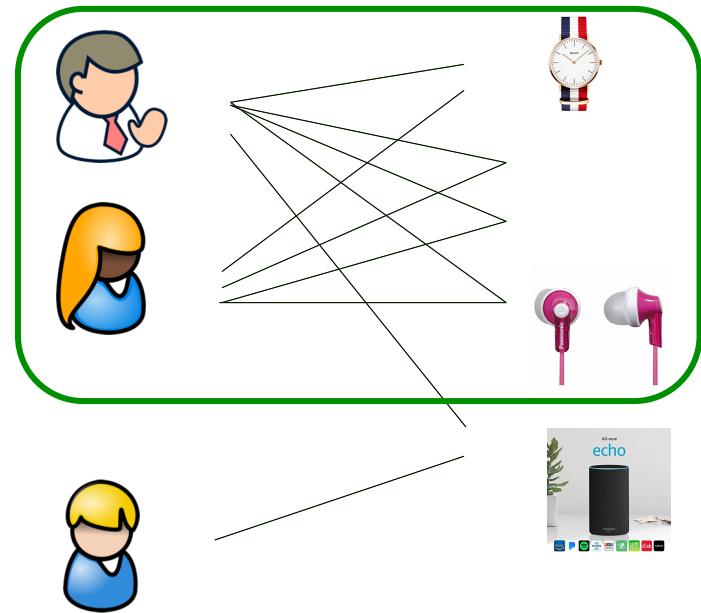
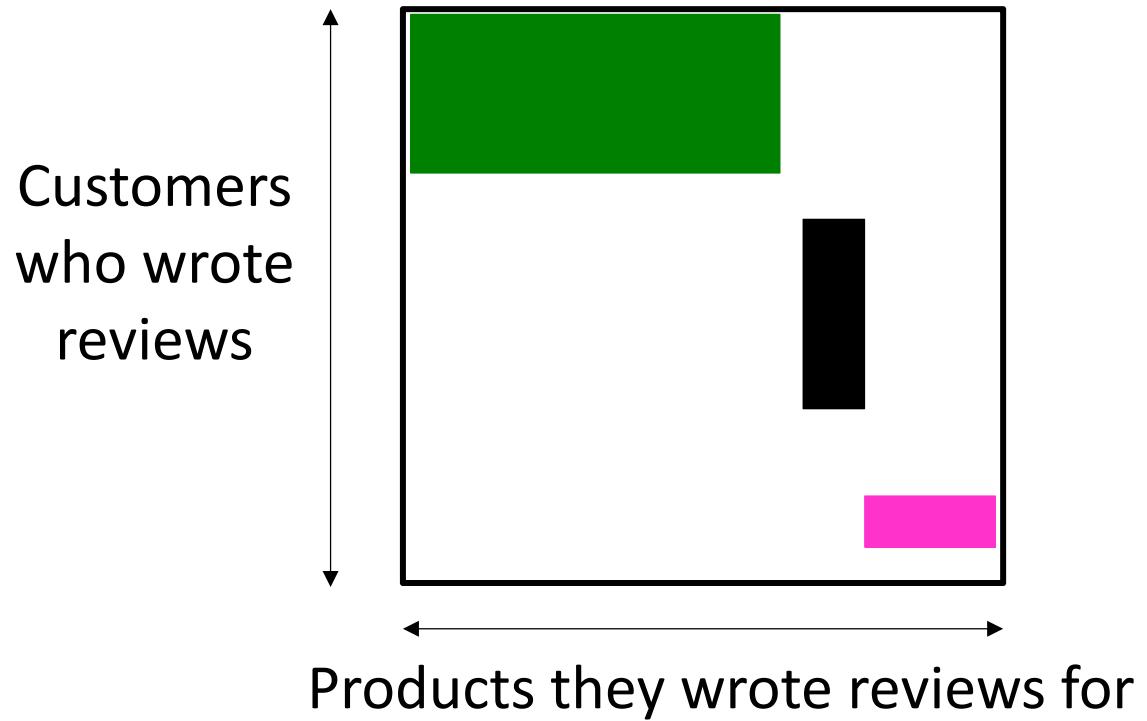
Customers that are similar to each other



Products under promotion may not reach intended customers

Review Abuse

Main Insight
Dense “blocks” are **suspicious**

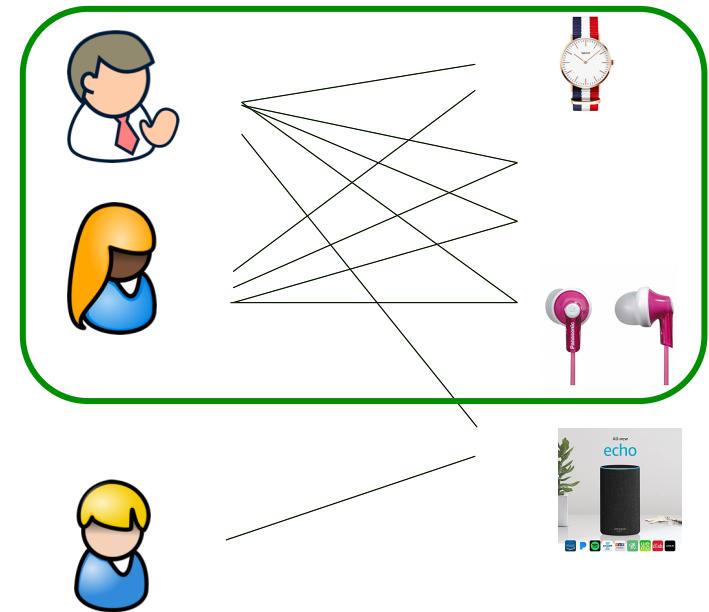
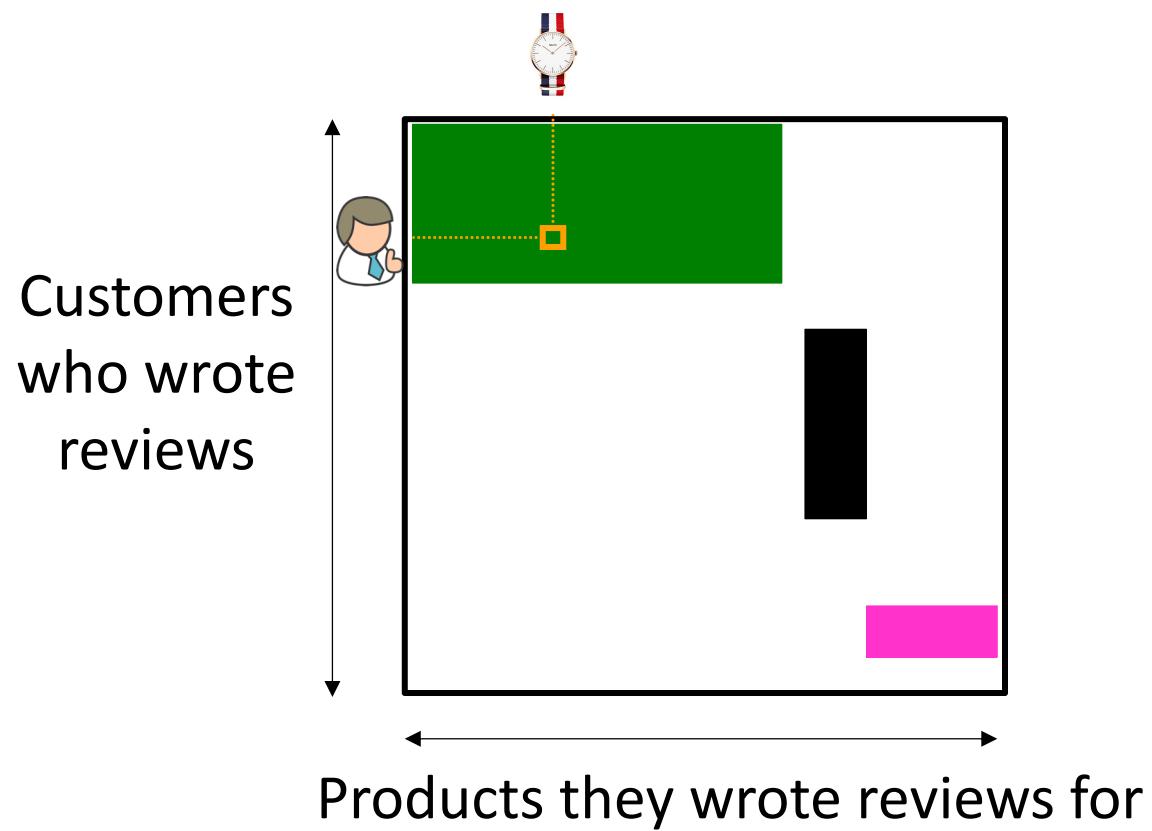


Review Abuse

Main Insight

Dense “blocks” are **suspicious**

Problem Statement: Given a $n \times m$ matrix A, can we spot dense, suspicious blocks in a scalable manner?



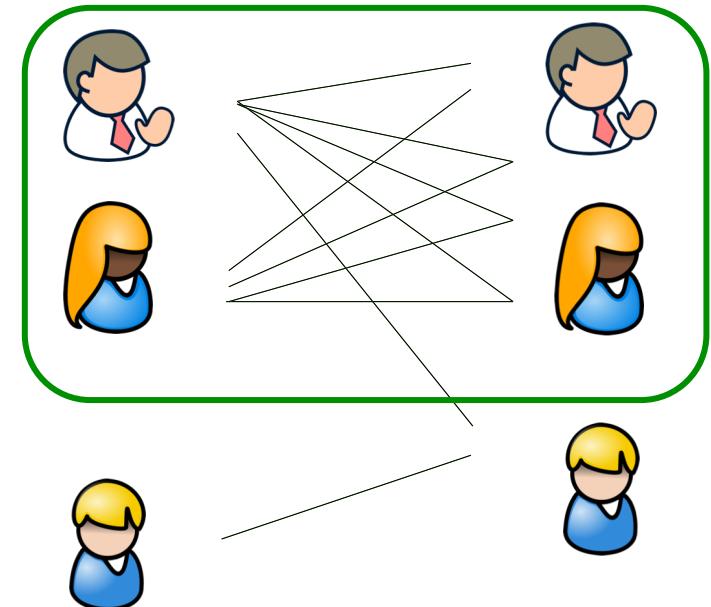
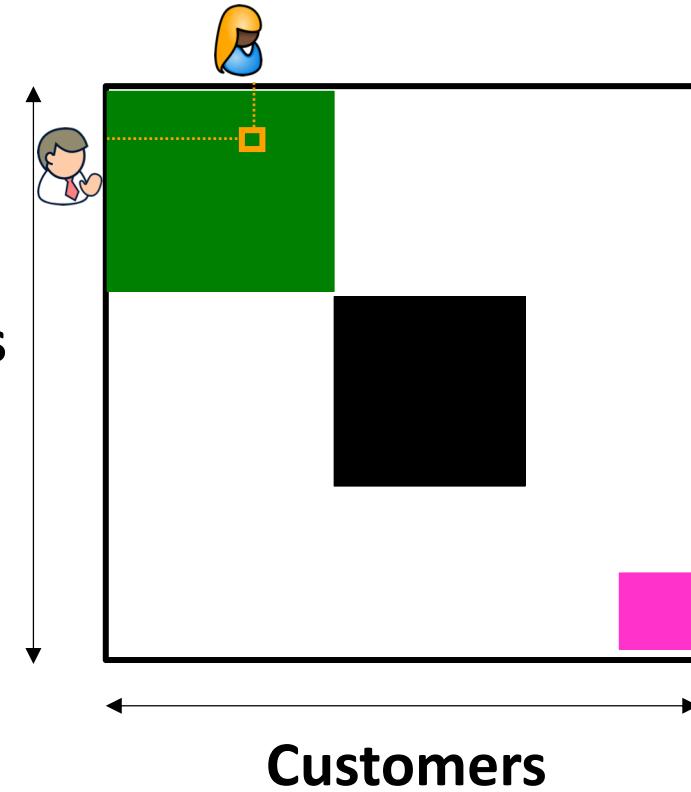
Main Insight

Dense “blocks” are **suspicious**

Problem Statement: Given a $n \times m$ matrix A , can we spot dense, suspicious blocks in a scalable manner?

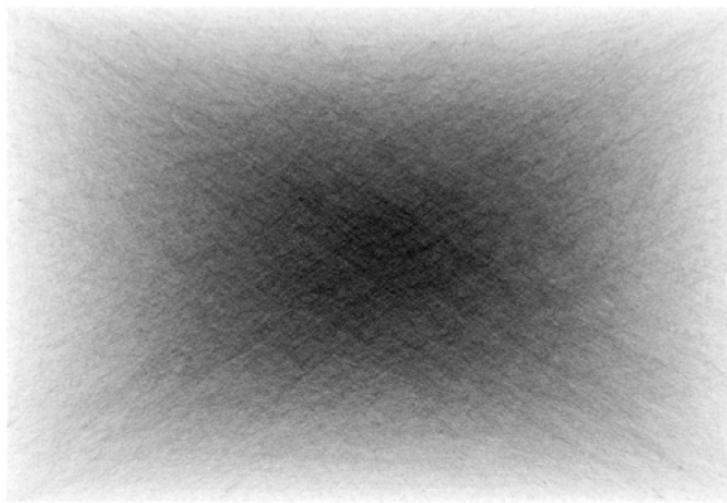
Reseller Abuse

Customers



Why is this challenging?

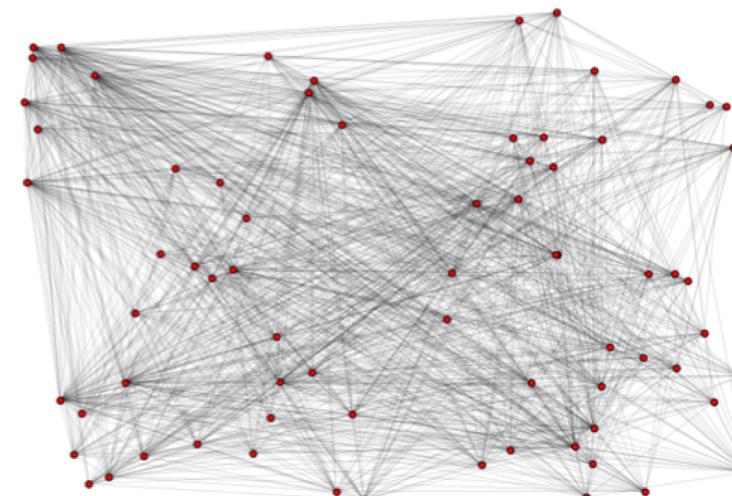
We start with...



~260,000 nodes

And need to find...

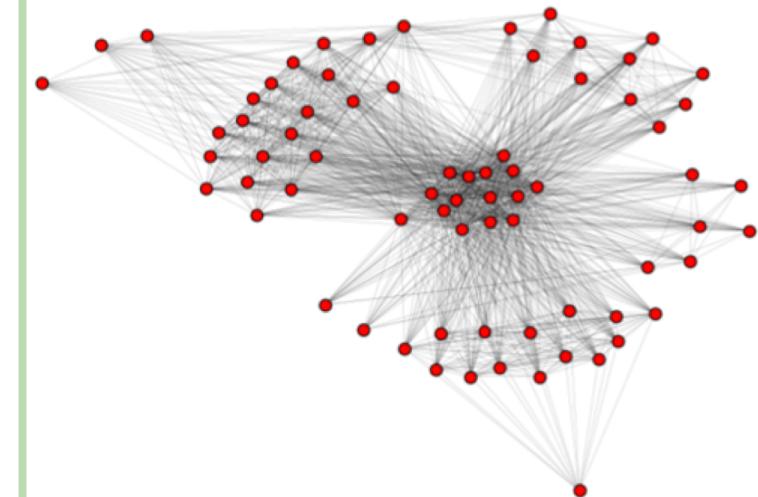
(Extract)



~72 tightly knit
members

Which, rearranged, looks
like...

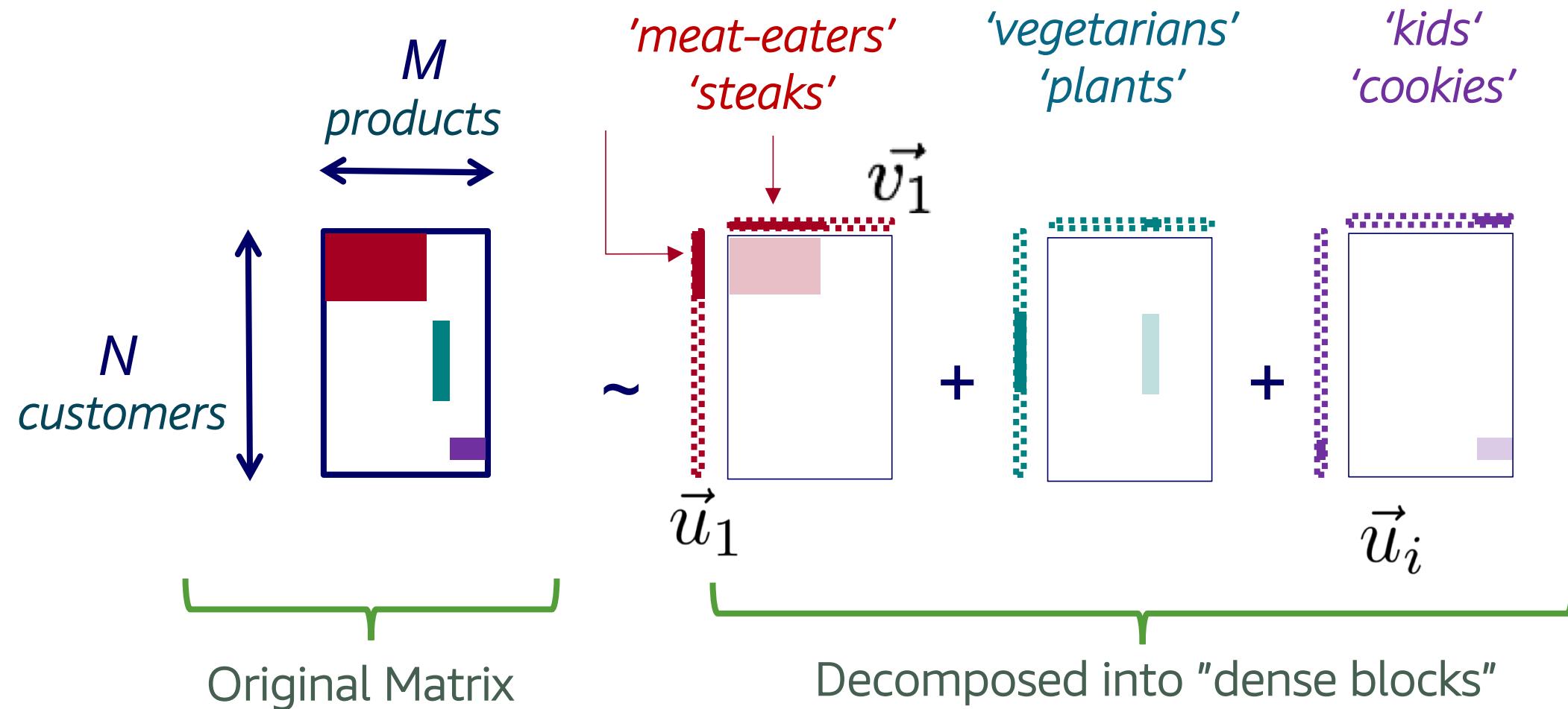
(Visualize)



...using a force-directed
layout

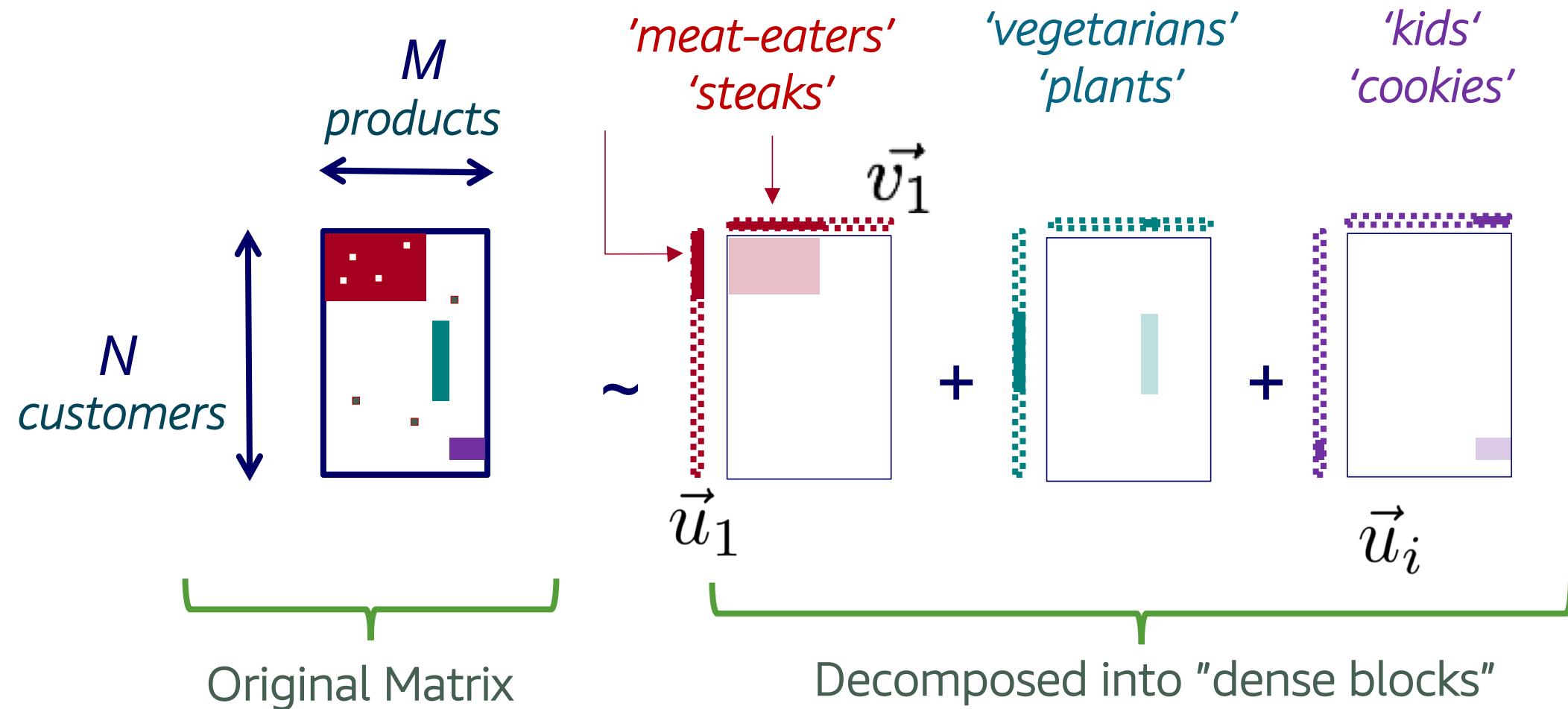
Use Singular Value Decomposition

Recall: (SVD) matrix factorization: finds blocks



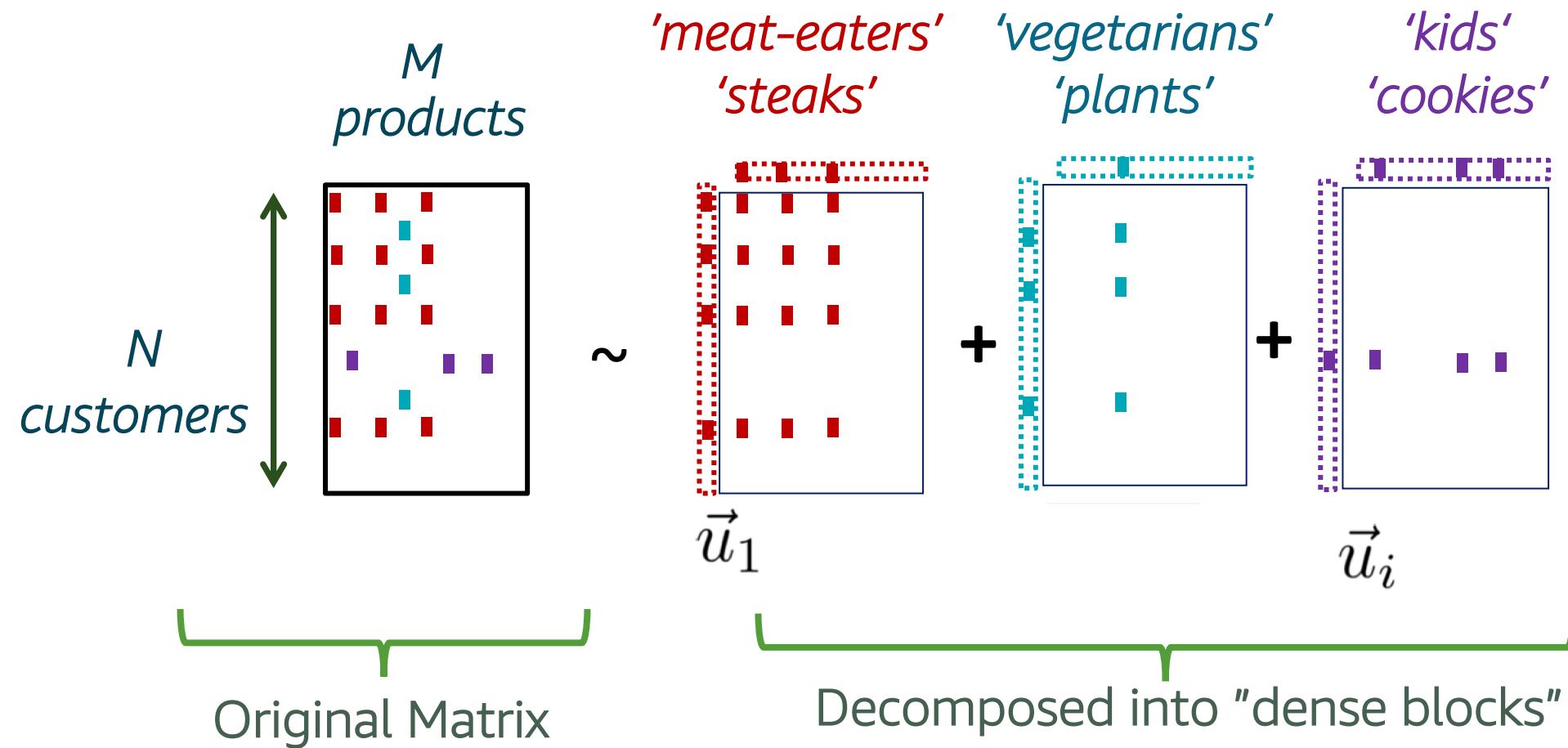
Use Singular Value Decomposition

Recall: (SVD) matrix factorization: finds blocks



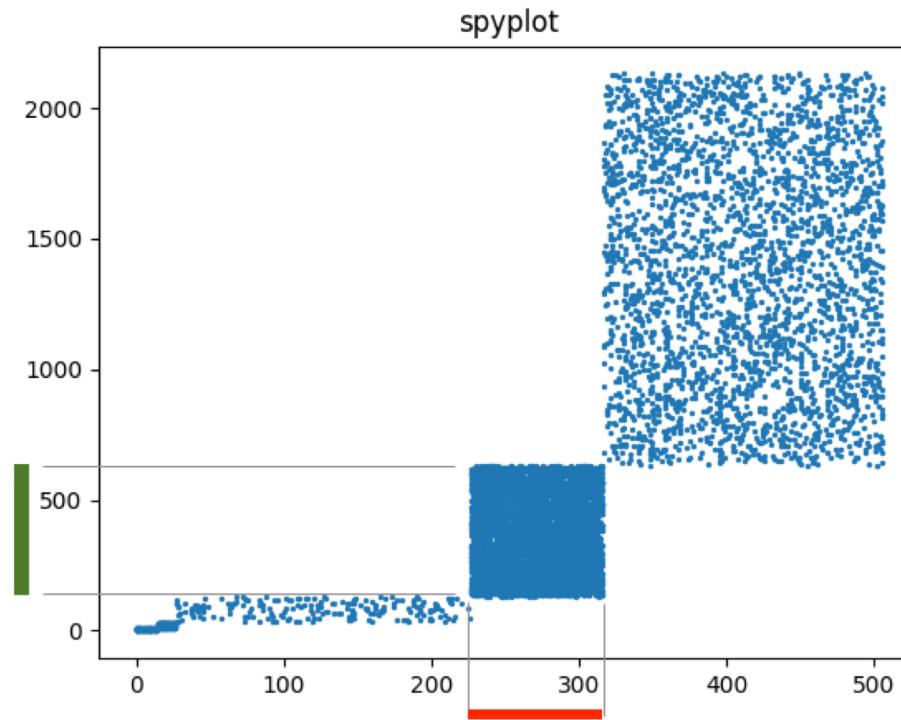
Use Singular Value Decomposition

Recall: (SVD) matrix factorization: finds blocks

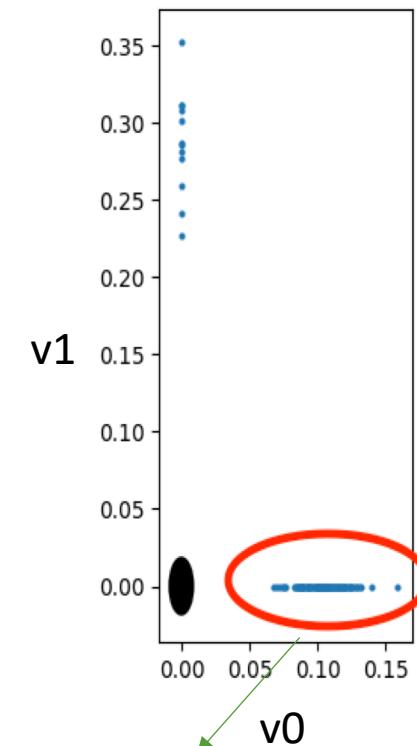
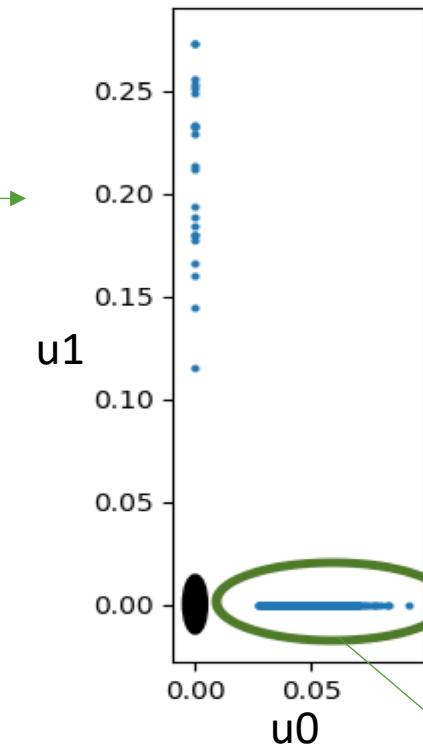


Use Singular Value Decomposition

A Toy Example



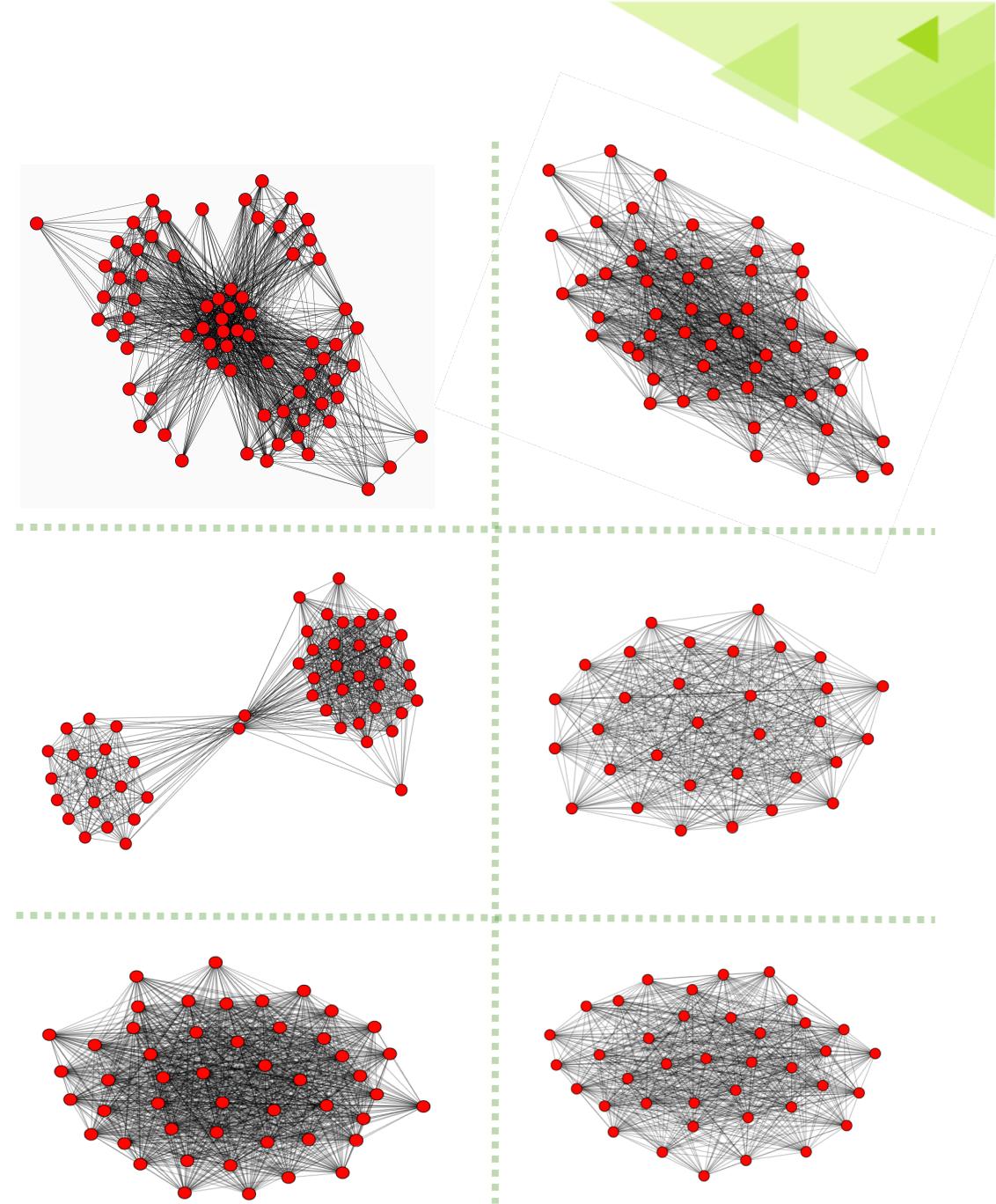
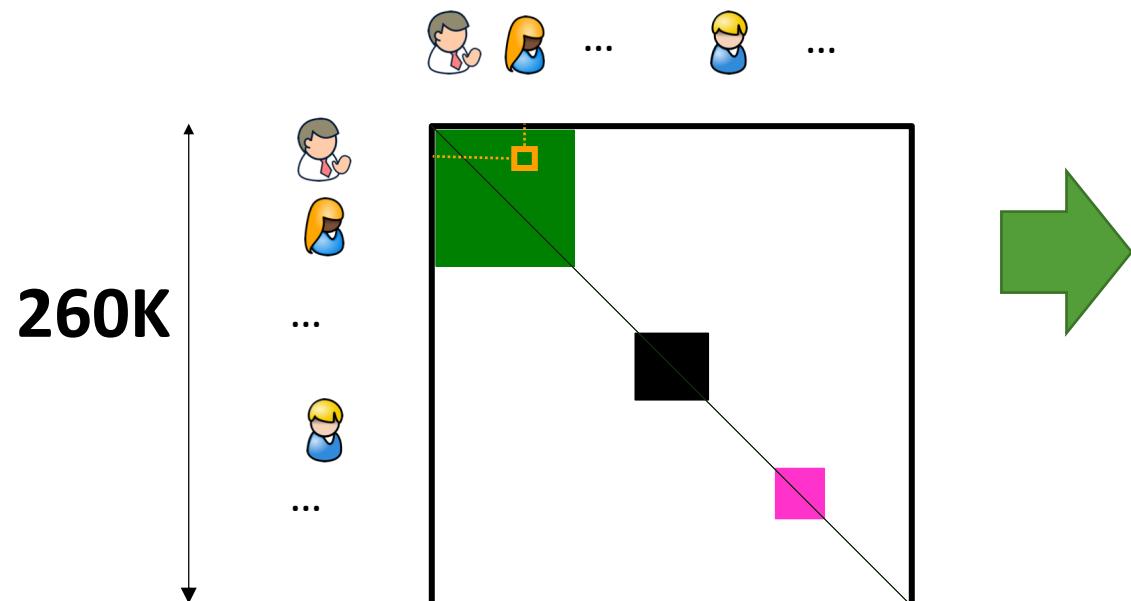
SVD



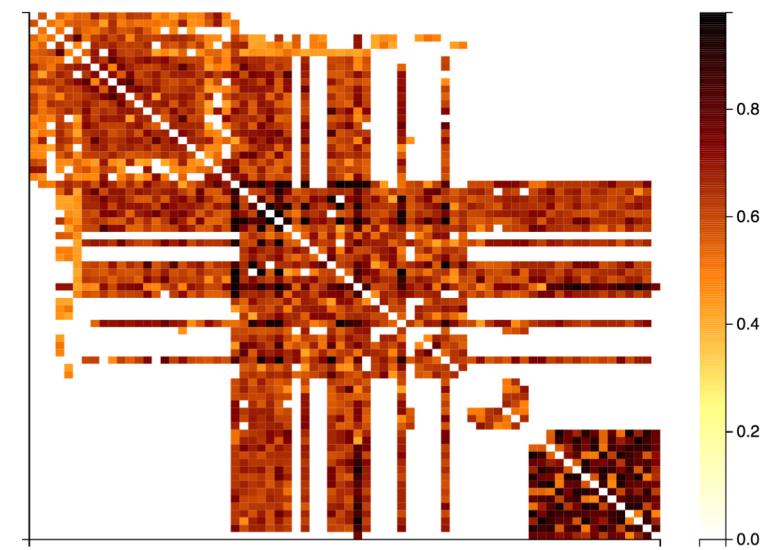
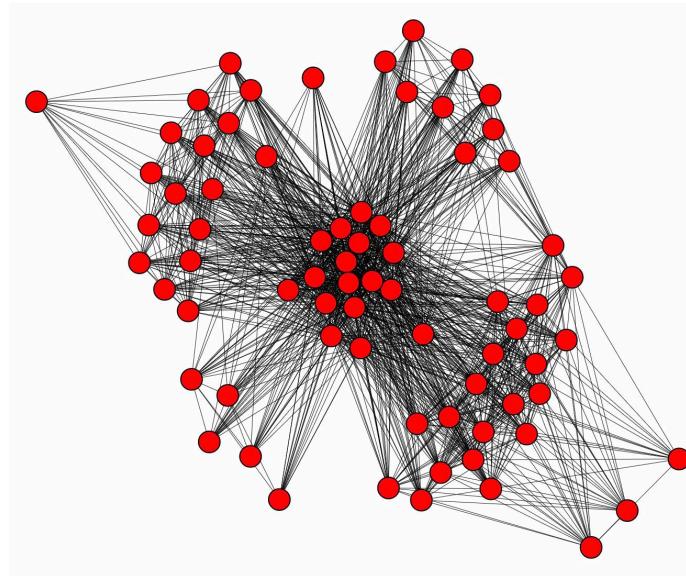
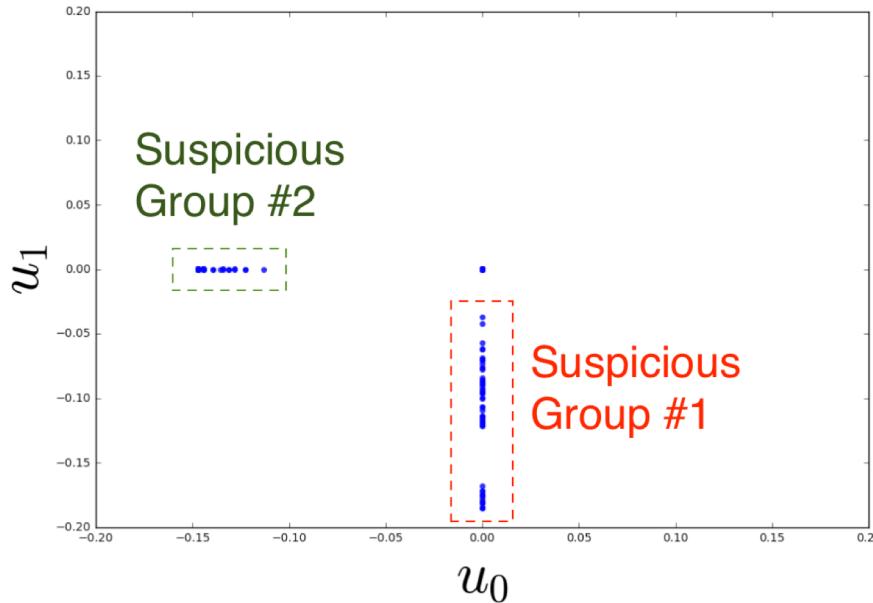
Extract these groups using clustering
(DBSCAN)

ClusterCatch in Action

Data: Customer x Customer Matrix
(~260K customers)



ClusterCatch: The Clover



“Violate Amazon Policy”



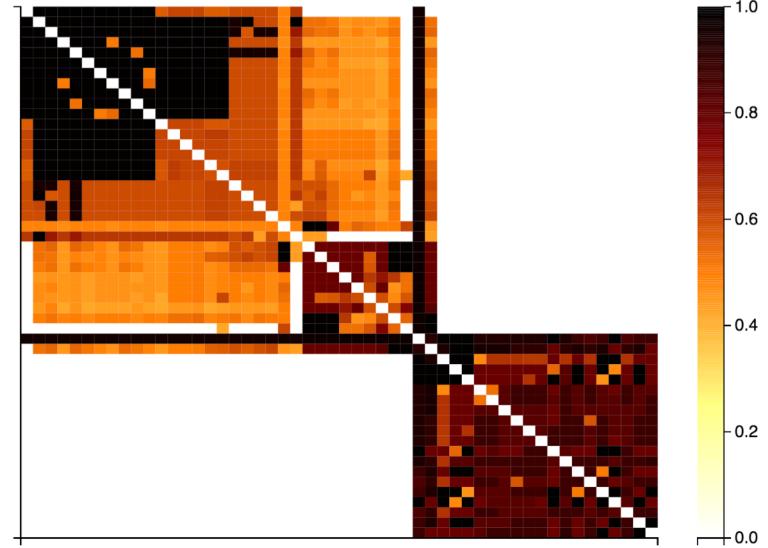
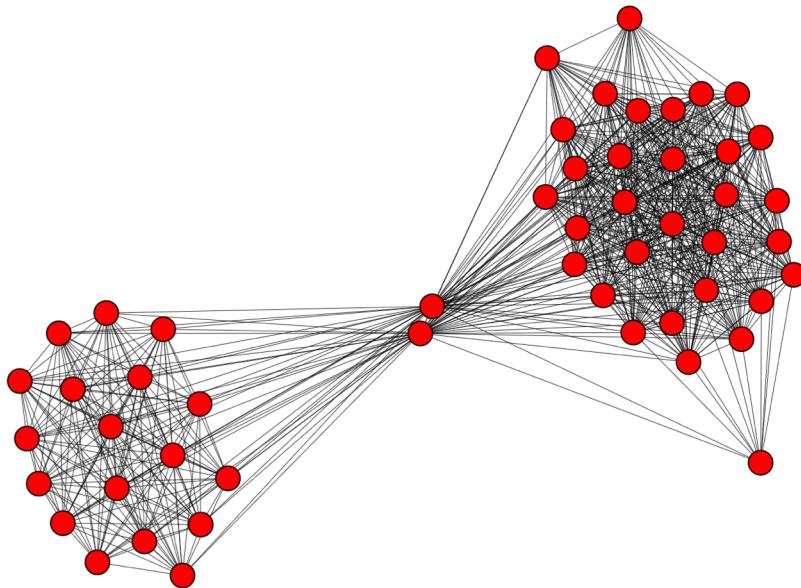
“Same 5 uncommon names”



“Chiefly use Gift Card”



ClusterCatch: The Bowtie



“Single EU country
of origin”



“Very similar
account names”

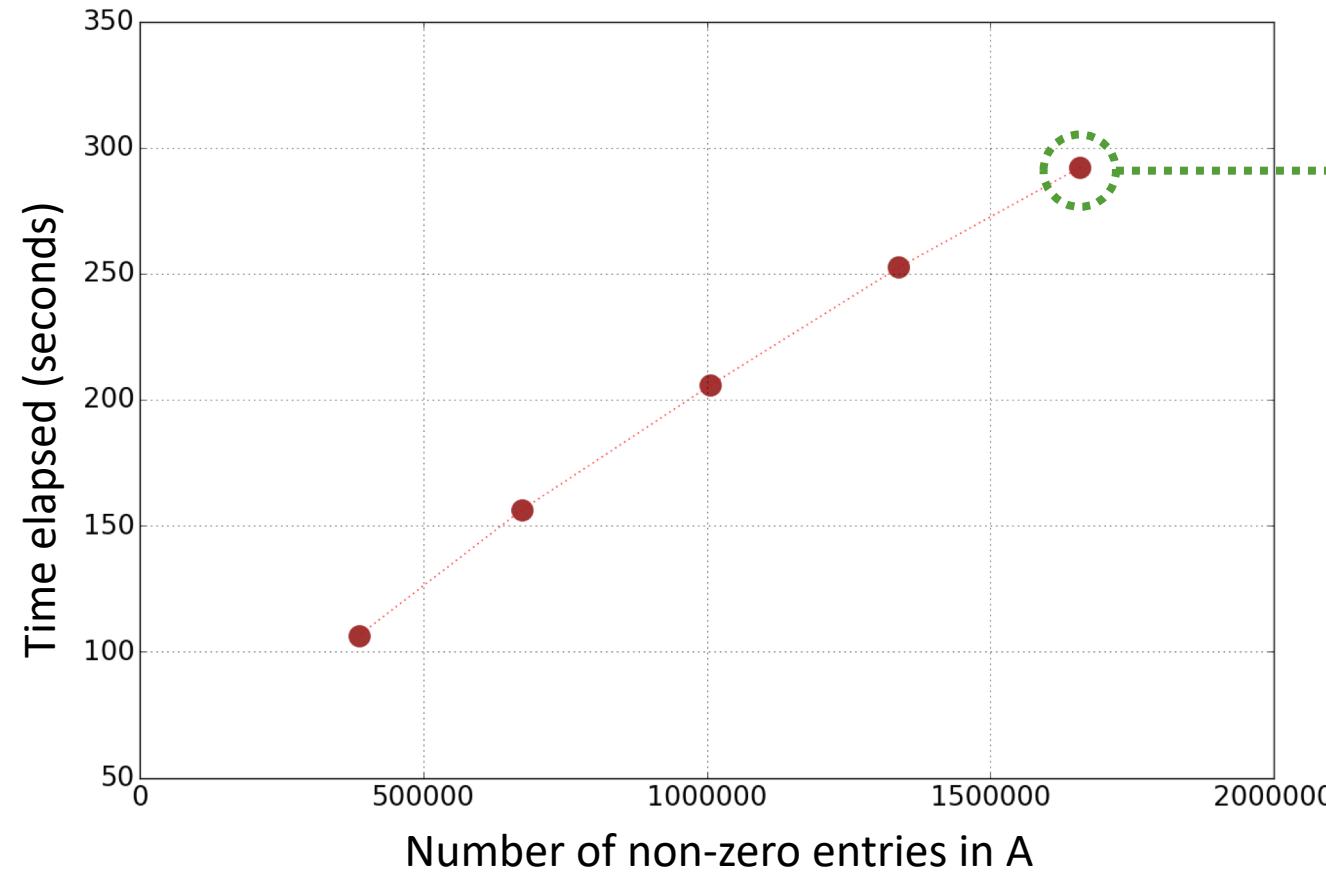


“Near-identical
email addresses”



Scales well: Will it take a week to run?

Experiments performed on standard Amazon EC2 d2.xlarge instance



5 minutes for
matrix A with
~1.7M Edges

ClusterCatch: Tip of the Iceberg

Can be applied to spotting collusion in **many** settings!

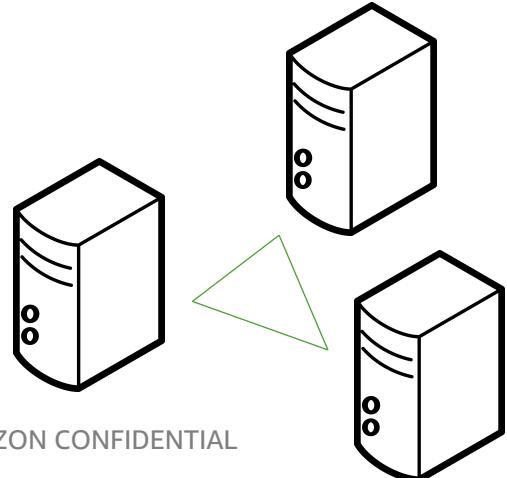
1. Reseller Abuse



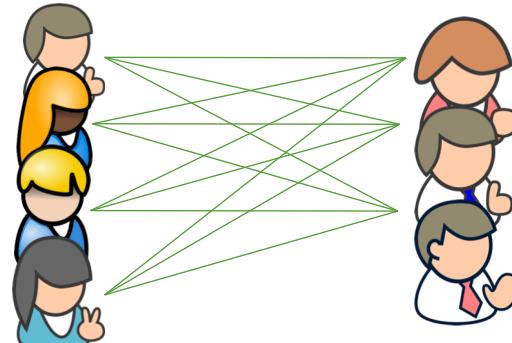
2. Prime Free-Trial



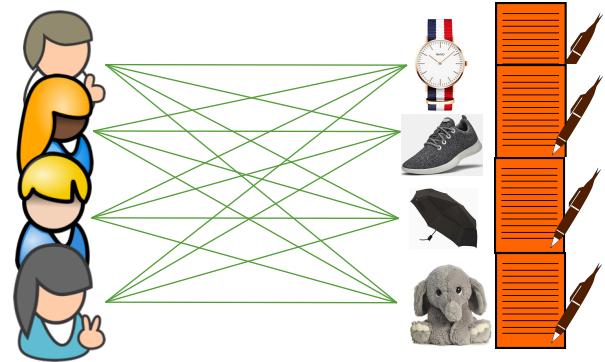
4. Cloud Monitoring



5. Buyer-seller collusion



3. Review Abuse

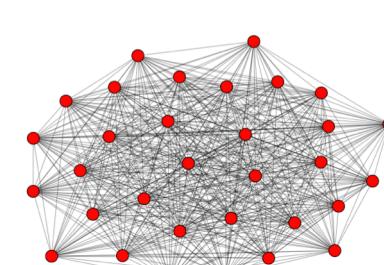
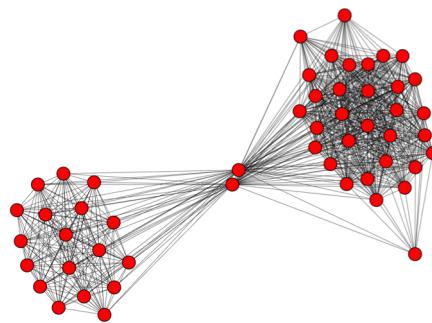
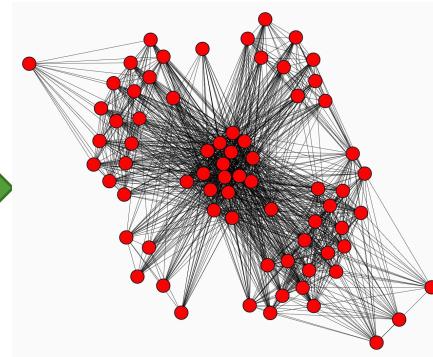
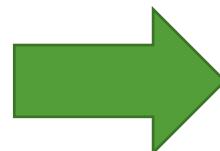
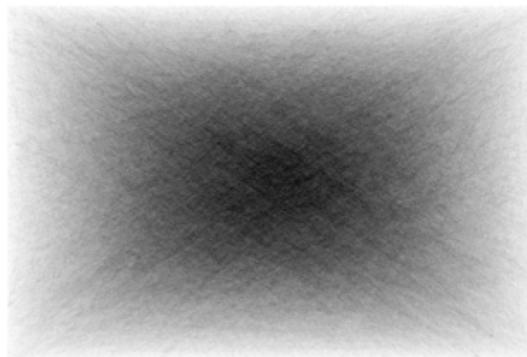


...and probably
many more

ClusterCatch: Conclusions

- **Scalable, effective and explainable** way to find suspicious blocks in a matrix
- Many potential applications

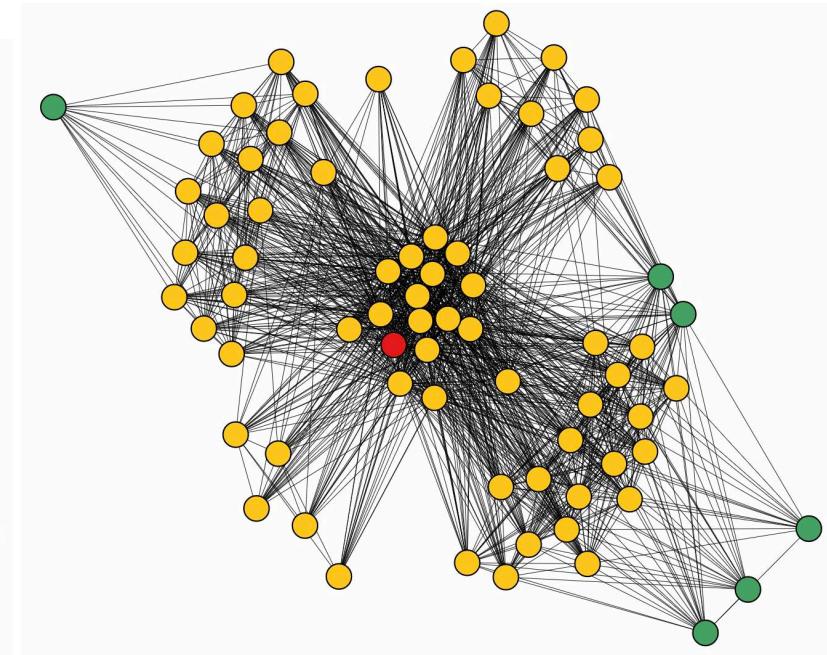
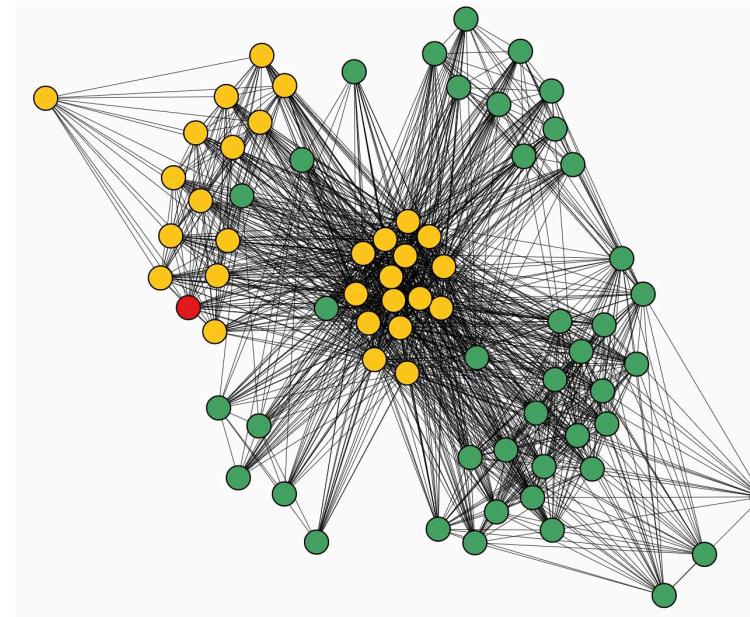
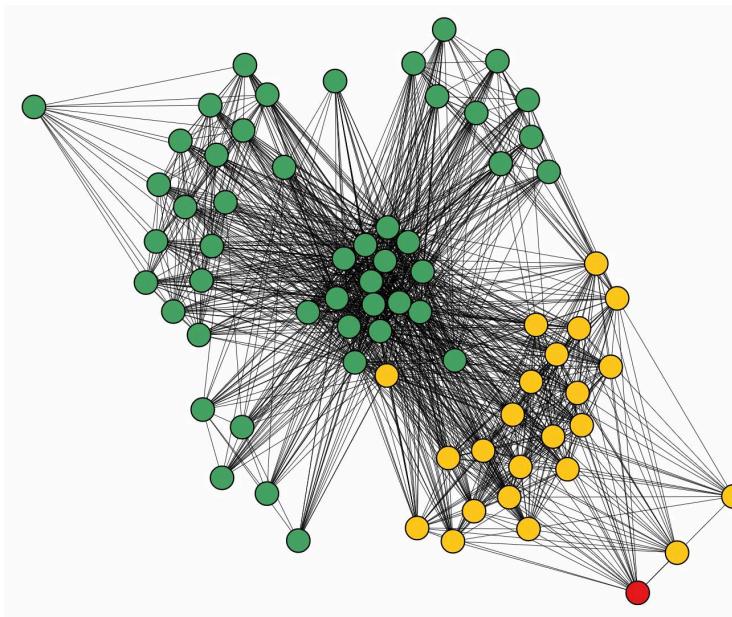
Thank you!





Appendix

Customer Clustering Service vs ClusterCatch



The two are complementary in nature