Recommendation Systems

CS771: Introduction to Machine Learning

Purushottam Kar

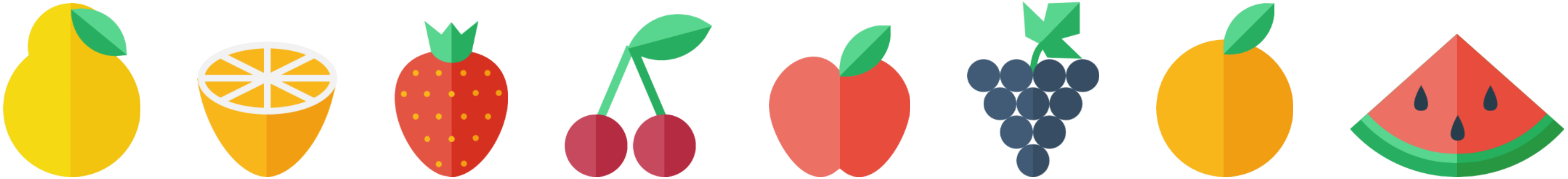
Outline of discussion

- Three major techniques
  - Collaborative Filtering
  - Extreme Classification
  - Multi-armed Bandits
- Barely scratching the surface
- Entire conference RecSys dedicated to recommendation systems
- Look for papers in other conferences too!
  NIPS, ICML, KDD, WWW, WSDM

Problem Statement

- There are users …

- … and there are items …



- … and we need to recommend, to each user, items s/he will “like”
- Users usually come to us one at a time in no particular order

A Powerful Abstraction

User

- Amazon.com user
- Facebook user
- Student
- Patient
- Search query
- …

Items

- Amazon.com products
- Facebook posts
- Study material
- Medicines
- Internet webpages
- …

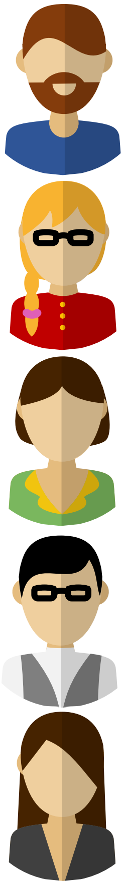
- Users are in millions and thousands come each second
- Items are in millions and each user likes only 5-10 items
- Some items are popular but most items liked by only 5-10 users

Collaborative Filtering

The Matrix Completion Problem

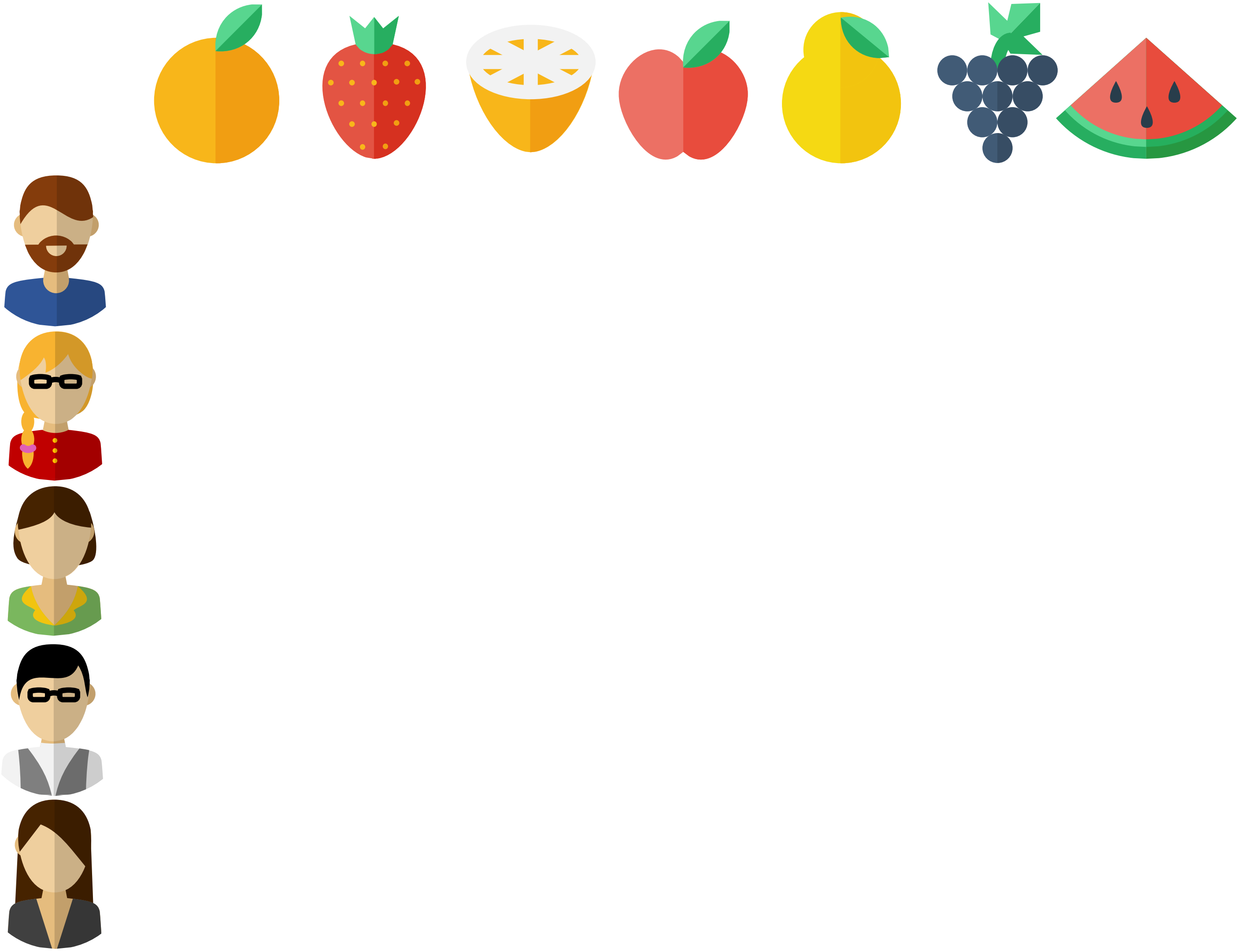
- m users n items
- Rating matrix $X \in \mathbb{R}^{m \times n}$
- X_{ij}: rating indicating how much user i likes item j
- Deeper shade $\Rightarrow$ more like
- Sometimes users rate some items (rarely though)
- So we get to see those entries of X, but only those
- Can we recover the unseen entries i.e. complete X?

The Matrix Completion Problem



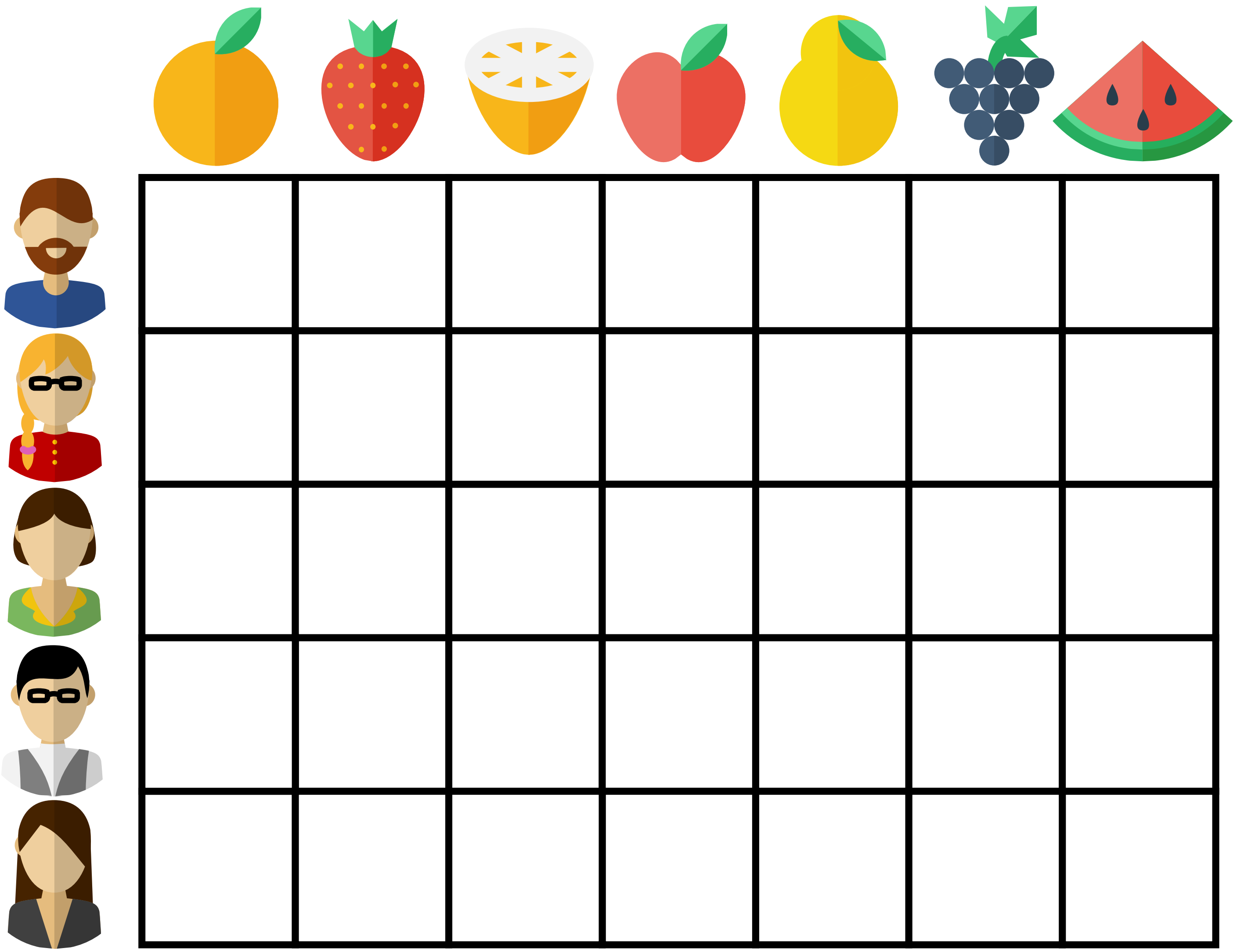
- m users n items
- Rating matrix $X \in \mathbb{R}^{m \times n}$
- X_{ij}: rating indicating how much user i likes item j
- Deeper shade ⇒ more like
- Sometimes users rate some items (rarely though)
- So we get to see those entries of X, but only those
- Can we recover the unseen entries i.e. complete X?

The Matrix Completion Problem



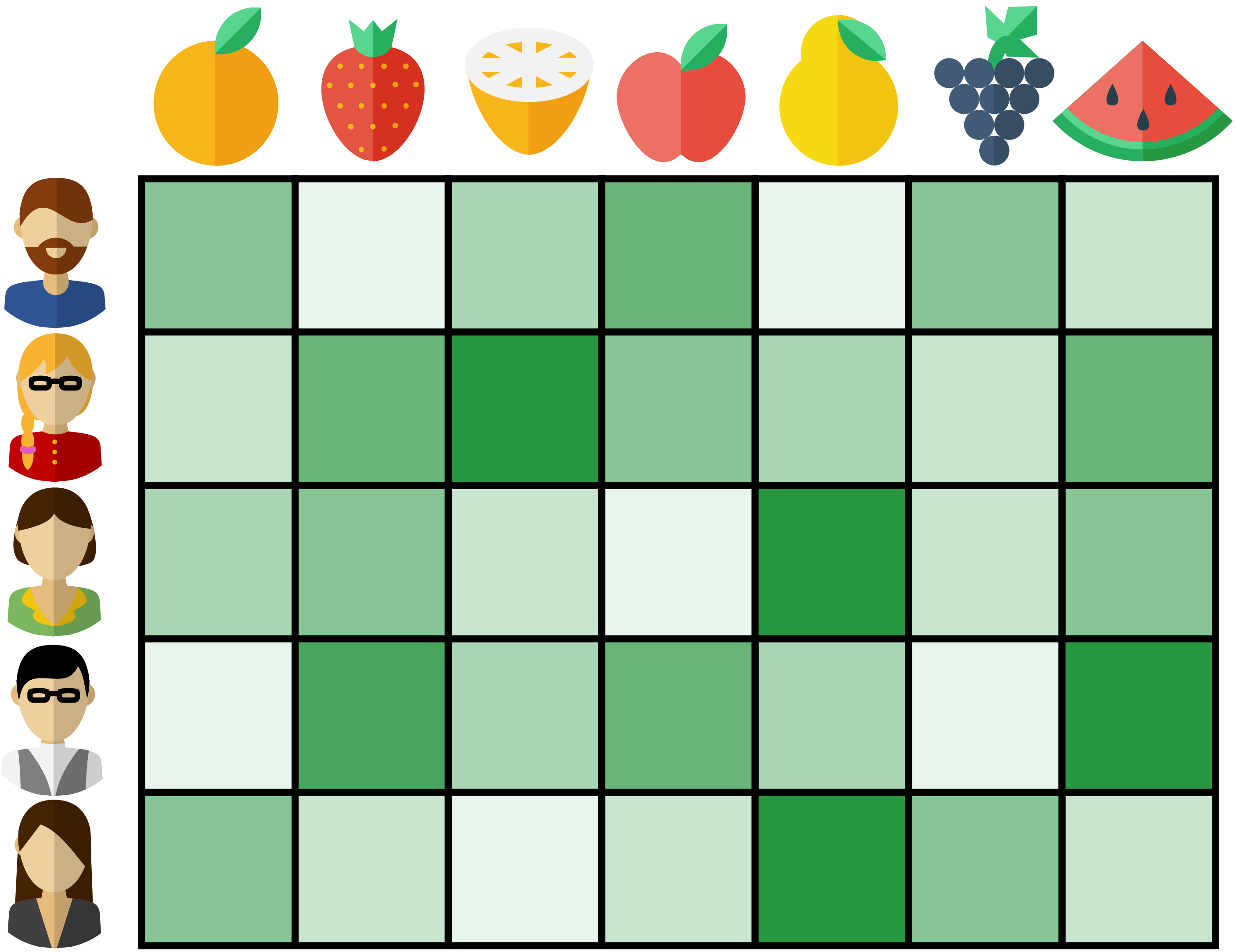
- m users n items
- Rating matrix $X \in \mathbb{R}^{m \times n}$
- X_{ij}: rating indicating how much user i likes item j
- Deeper shade $\Rightarrow$ more like
- Sometimes users rate some items (rarely though)
- So we get to see those entries of X, but only those
- Can we recover the unseen entries i.e. complete X?

The Matrix Completion Problem



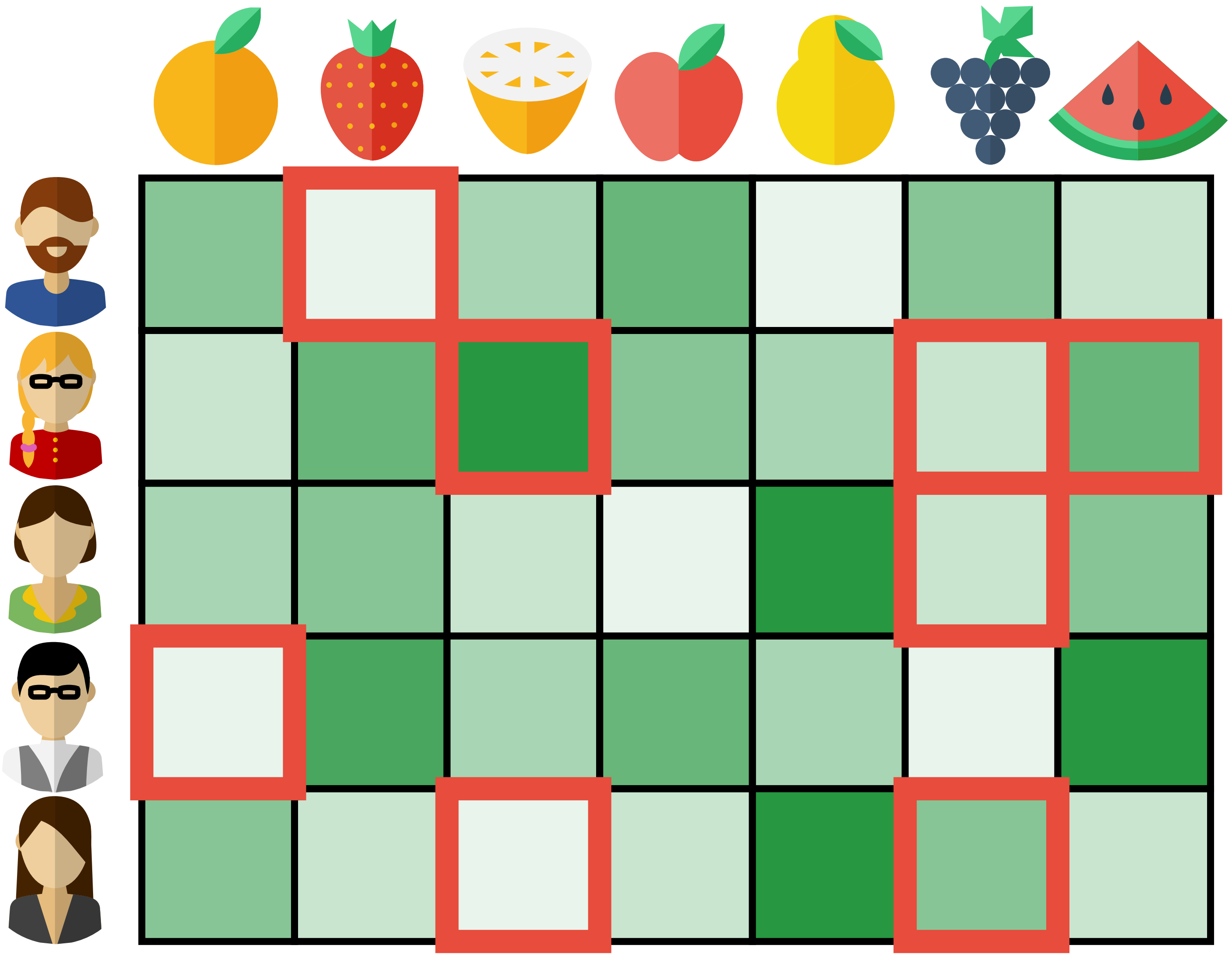
- m users n items
- Rating matrix $X \in \mathbb{R}^{m \times n}$
- X_{ij}: rating indicating how much user i likes item j
- Deeper shade $\Rightarrow$ more like
- Sometimes users rate some items (rarely though)
- So we get to see those entries of X, but only those
- Can we recover the unseen entries i.e. complete X?

The Matrix Completion Problem



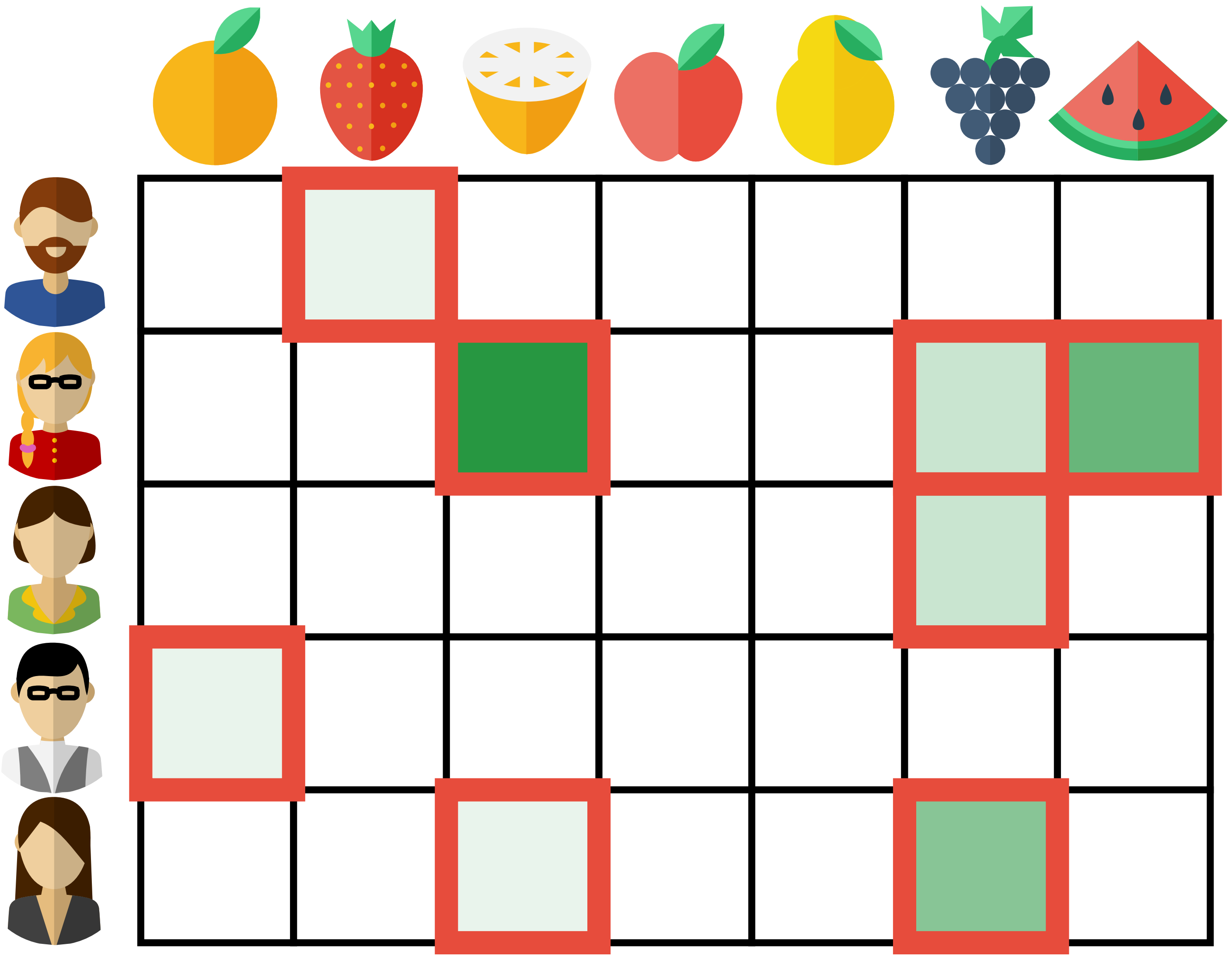
- m users n items
- Rating matrix $X \in \mathbb{R}^{m \times n}$
- X_{ij}: rating indicating how much user i likes item j
- Deeper shade $\Rightarrow$ more like
- Sometimes users rate some items (rarely though)
- So we get to see those entries of X, but only those
- Can we recover the unseen entries i.e. complete X?

The Matrix Completion Problem



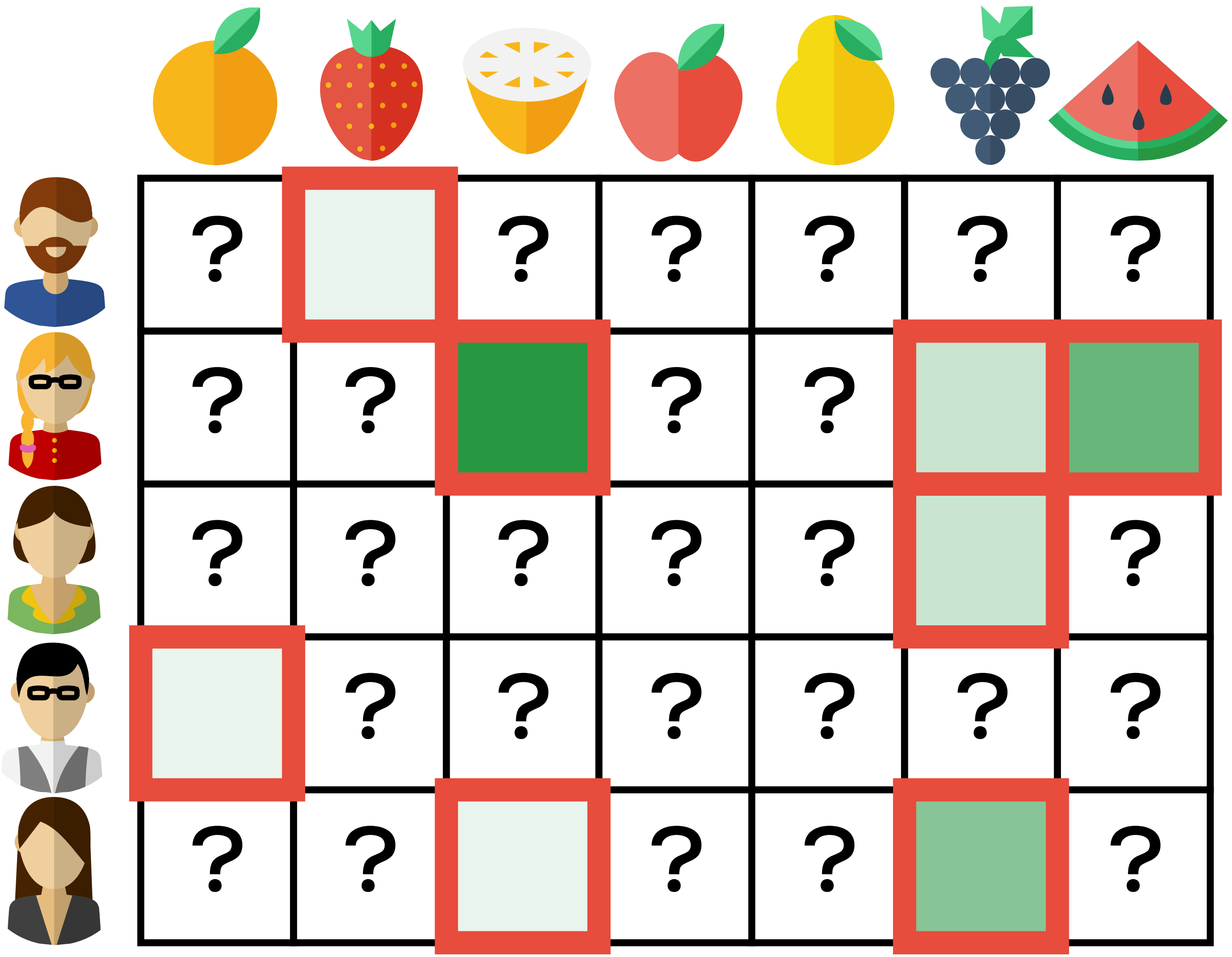
- m users n items
- Rating matrix $X \in \mathbb{R}^{m \times n}$
- X_{ij}: rating indicating how much user i likes item j
- Deeper shade ⇒ more like
- Sometimes users rate some items (rarely though)
- So we get to see those entries of X, but only those
- Can we recover the unseen entries i.e. complete X?

The Matrix Completion Problem



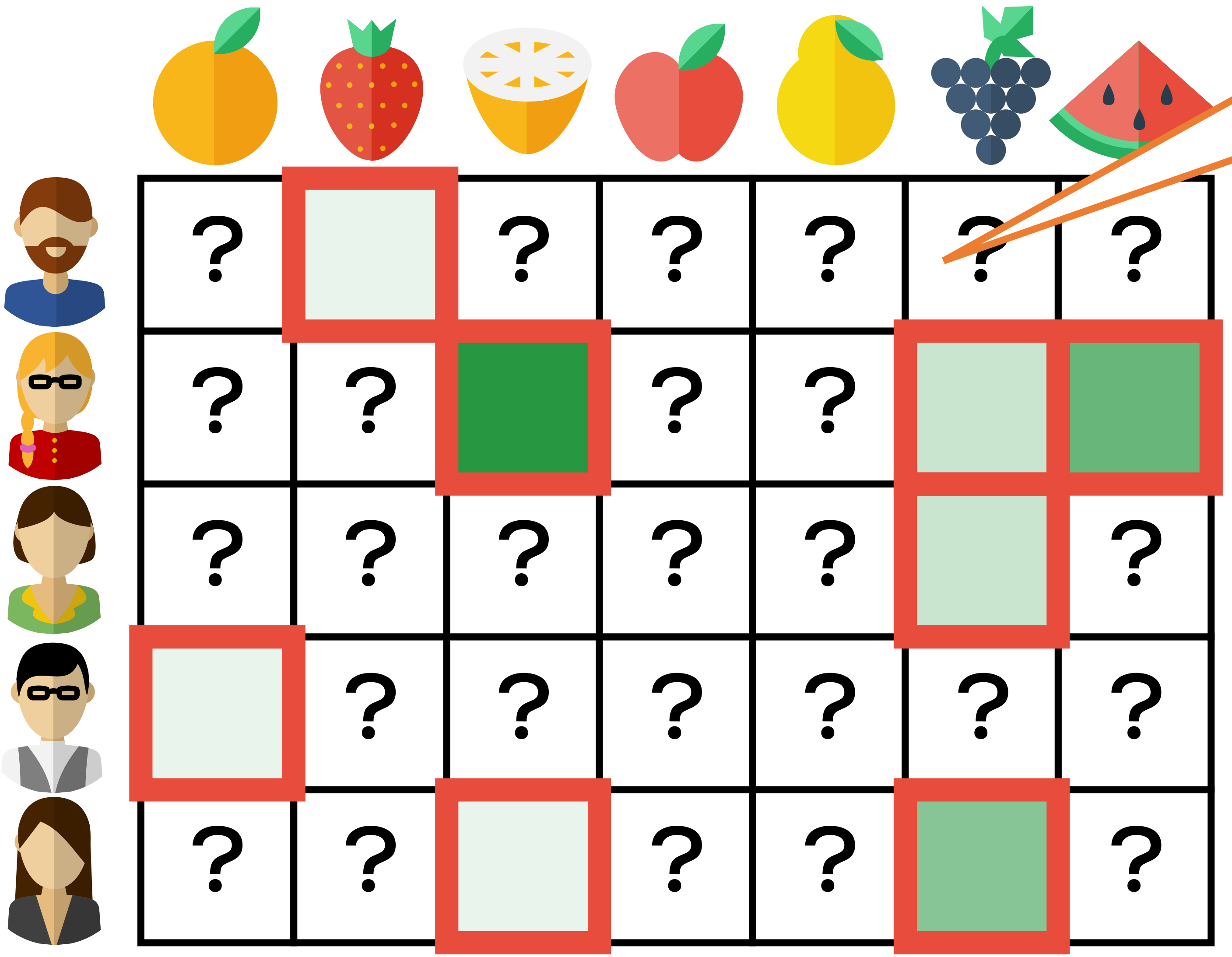
- m users n items
- Rating matrix $X \in \mathbb{R}^{m \times n}$
- X_{ij}: rating indicating how much user i likes item j
- Deeper shade ⇒ more like
- Sometimes users rate some items (rarely though)
- So we get to see those entries of X, but only those
- Can we recover the unseen entries i.e. complete X?

The Matrix Completion Problem



- m users n items
- Rating matrix $X \in \mathbb{R}^{m \times n}$
- X_{ij}: rating indicating how much user i likes item j
- Deeper shade ⇒ more like
- Sometimes users rate some items (rarely though)
- So we get to see those entries of X, but only those
- Can we recover the unseen entries i.e. complete X?

The Matrix Completion Prob



Knowing the entire matrix would reveal the most liked items for every user and allow me to recommend those!

- X_{ij}: rating indicating how much user i likes item j
- Deeper shade ⇒ more like
- Sometimes users rate some items (rarely though)
- So we get to see those entries of X, but only those
- Can we recover the unseen entries i.e. complete X?

The Low-rank Matrix Completion Problem

- Unfortunately the matrix completion problem is ill-posed
- No reason why observed entries of the matrix should tell us anything about the unobserved entries
- Absolutely necessary to impose some structure on the matrix
- Popular assumption: the rating matrix X is low rank (say rank k)
- So need to find a low-rank matrix $\hat{X}$ such that X and $\hat{X}$ agree on the observed entries. Let $\Omega \subset [m] \times [n]$ be the locations observed

$$\arg \min_{\mathrm{rank}(\hat{X}) \leq k} \sum_{(i,j) \in \Omega} \left(X_{ij} - \hat{X}_{ij} \right)^2$$

- However, every rank k matrix $\hat{X}$ can be written as $\hat{X} = UV^\top$ where $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{n \times k}$ (show this using singular value decomp!)

The Low-rank Matrix Completion Problem

- Unf … compl … sed
- No … entries … tell us any … served
- Absolutely necessary to impose some structure on the matrix
- Popular assumption: the rating matrix X is low rank (say rank k)
- So need to find a low-rank matrix $\hat{X}$ such that X and $\hat{X}$ agree on the observed entries. Let $\Omega \subset [m] \times [n]$ be the locations observed

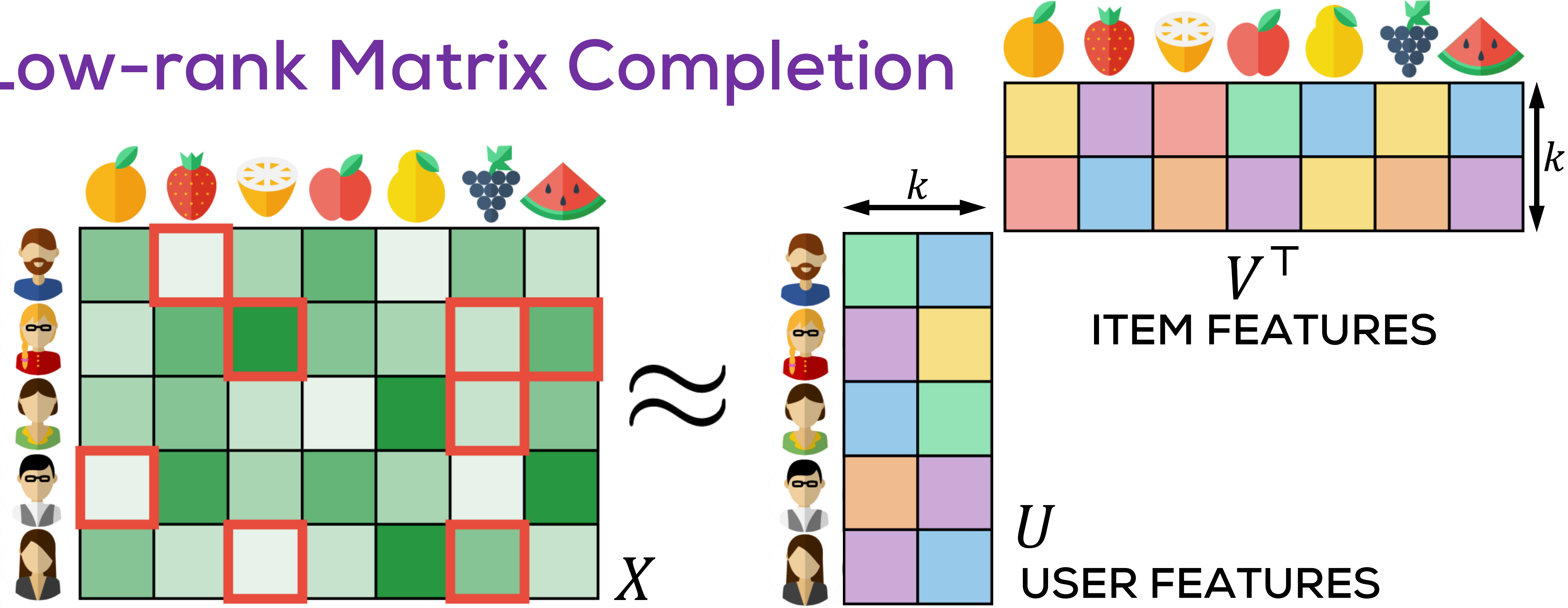
$$\arg \min_{\text{rank}(\hat{X}) \leq k} \sum_{(i,j)\in\Omega} \left(X_{ij} - \hat{X}_{ij}\right)^2$$

- However, every rank k matrix $\hat{X}$ can be written as $\hat{X} = UV^\top$ where $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{n \times k}$ (show this using singular value decomp!)

Can encode intuitions like “similar items rated similarly by the same user/similar users”

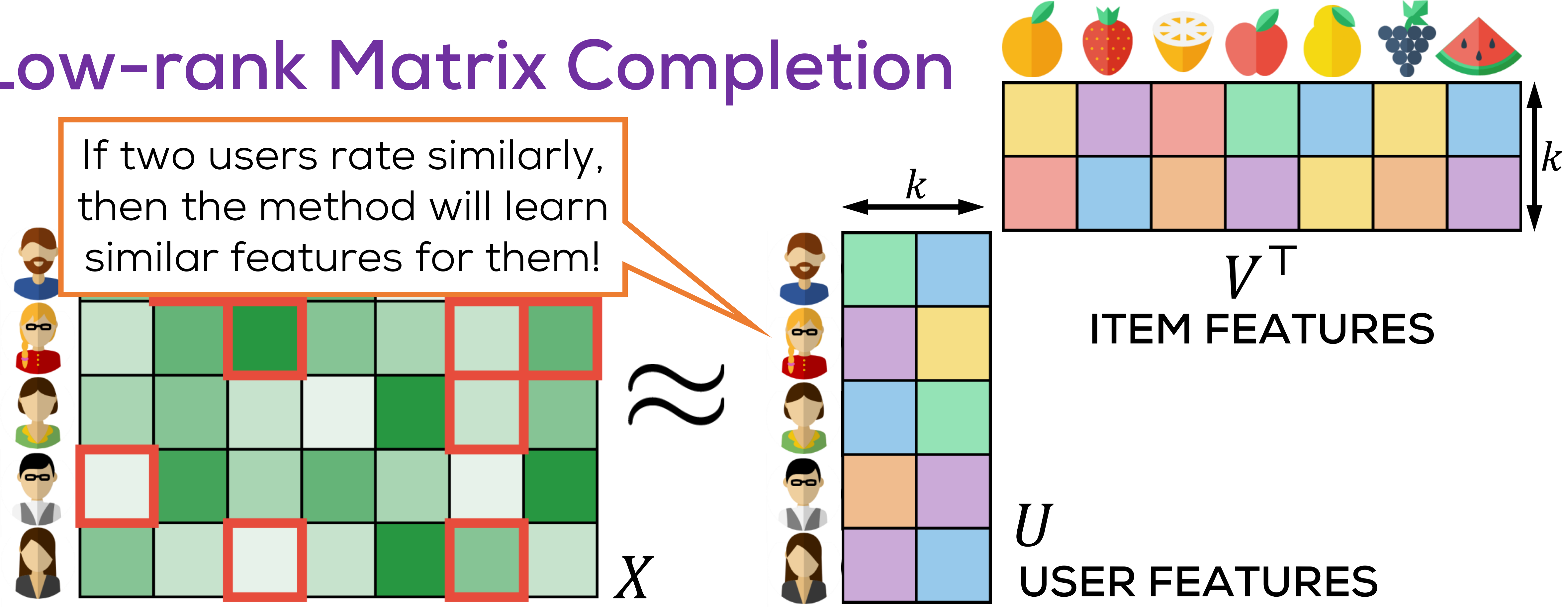
k tuned based on how many entries observed – more entries, larger k

Low-rank Matrix Completion



- So we have $X_{ij} \approx \langle \mathbf{u}^i, \mathbf{v}^j \rangle$ where $U = [\mathbf{u}^1, \dots, \mathbf{u}^m]^\top$ and $V = [\mathbf{v}^1, \dots, \mathbf{v}^n]^\top$
- Nice! As a side effect, vector representations of all users and items
- Such features can be very useful in recommendation (will soon see)
- Can also violate privacy if details users didn't tell are revealed
- E.g. if one of the k features turns out to be correlated with age

Low-rank Matrix Completion



- So we have $X_{ij} \approx \langle \mathbf{u}^i, \mathbf{v}^j \rangle$ where $U = [\mathbf{u}^1, \ldots, \mathbf{u}^m]^\top$ and $V = [\mathbf{v}^1, \ldots, \mathbf{v}^n]^\top$
- Nice! As a side effect, vector representations of all users and items
- Such features can be very useful in recommendation (will soon see)
- Can also violate privacy if details users didn't tell are revealed
- E.g. if one of the k features turns out to be correlated with age

Solving the LRMC problem

Solving the LRMC problem

$$\arg \min_{\text{rank}(\hat{X}) \leq k} \sum_{(i,j) \in \Omega} \left(X_{ij} - \hat{X}_{ij}\right)^2$$

Solving the LRMC problem

$$\arg \min_{\substack{U \in \mathbb{R}^{m \times k} \\ V \in \mathbb{R}^{n \times k}}} \sum_{(i,j) \in \Omega} \left(X_{ij} - \langle \mathbf{u}^i, \mathbf{v}^j \rangle \right)^2$$

Solving the LRMC problem

$$\arg \min_{\substack{U \in \mathbb{R}^{m \times k} \\ V \in \mathbb{R}^{n \times k}}} \sum_{(i,j) \in \Omega} \left(X_{ij} - \langle \mathbf{u}^i, \mathbf{v}^j \rangle \right)^2 + \lambda \cdot \left(\|U\|_F^2 + \|V\|_F^2 \right)$$

Solving the LRMC problem

Nice to regularize

$$\arg \min_{\substack{U \in \mathbb{R}^{m \times k} \\ V \in \mathbb{R}^{n \times k}}} \sum_{(i,j) \in \Omega} \left(X_{ij} - \langle \mathbf{u}^i, \mathbf{v}^j \rangle \right)^2 + \lambda \cdot \left(\|U\|_F^2 + \|V\|_F^2 \right)$$

Solving the LRMC problem

Nice to regularize

$$\arg \min_{\substack{U \in \mathbb{R}^{m \times k} \\ V \in \mathbb{R}^{n \times k}}} \sum_{(i,j) \in \Omega} \left(X_{ij} - \langle \mathbf{u}^i, \mathbf{v}^j \rangle \right)^2 + \lambda \cdot \left(\sum_{i=1}^{m} \left\| \mathbf{u}^i \right\|_2^2 + \sum_{j=1}^{n} \left\| \mathbf{v}^j \right\|_2^2 \right)$$

Solving the LRMC problem

Nice to regularize

$$\arg \min_{\substack{U \in \mathbb{R}^{m \times k} \\ V \in \mathbb{R}^{n \times k}}} \sum_{(i,j) \in \Omega} \left(X_{ij} - \langle \mathbf{u}^i, \mathbf{v}^j \rangle \right)^2 + \lambda \cdot \left(\sum_{i=1}^{m} \| \mathbf{u}^i \|_2^2 + \sum_{j=1}^{n} \| \mathbf{v}^j \|_2^2 \right)$$

- Note that loss calculated only over observed entries i.e. $(i,j) \in \Omega$
- How to solve this ? Non convex problem! ☹
- Can use projected gradient descent on $\arg \min_{\mathrm{rank}(\hat{X}) \leq k} \sum_{(i,j) \in \Omega} \left(X_{ij} - \hat{X}_{ij} \right)^2$
- A faster way exists using alternating minimization
- AltMin is very versatile, gives us EM, Lloyd's algo, and now this
- Current state-of-the-art for matrix completion is AltMin

AltMin for LRMC

$$\arg\min_{\substack{U \in \mathbb{R}^{m\times k}\\ V \in \mathbb{R}^{n\times k}}} \sum_{(i,j)\in\Omega} \left(X_{ij} - \langle \mathbf{u}^i, \mathbf{v}^j \rangle\right)^2 + \lambda \cdot \left(\sum_{i=1}^{m} \left\| \mathbf{u}^i \right\|_2^2 + \sum_{j=1}^{n} \left\| \mathbf{v}^j \right\|_2^2 \right)$$

- For now, fix all $\mathbf{v}^j$ and all $\mathbf{u}^i$ except $\mathbf{u}^1$. The problem now becomes

$$\mathbf{u}^1 = \arg\min_{\mathbf{u}\in\mathbb{R}^k} \sum_{(1,j)\in\Omega} \left(X_{1j} - \langle \mathbf{u}, \mathbf{v}^j \rangle\right)^2 + \lambda \cdot \|\mathbf{u}\|_2^2$$

- Wait … this is just a ridge regression problem!
- What happens if I fix all $\mathbf{u}^i$ and all $\mathbf{v}^j$ except $\mathbf{v}^1$

$$\mathbf{v}^1 = \arg\min_{\mathbf{v}\in\mathbb{R}^k} \sum_{(i,1)\in\Omega} \left(X_{i1} - \langle \mathbf{u}^i, \mathbf{v} \rangle\right)^2 + \lambda \cdot \|\mathbf{v}\|_2^2$$

- Nice … again just a ridge regression problem!

AltMin for LRMC

Notice that summation is only over items that user 1 has rated – usually very small

$$\ldots - \langle \mathbf{u}^i, \mathbf{v}^j \rangle\big)^2 + \lambda \cdot \Bigg(\ldots$$

$V \in \mathbb{R}^{n \times k}$

$\mathbf{v}^j$ act as the "feature" vectors and X_{1j} as "responses" in this ridge regression problem

- For now, fix all $\mathbf{v}^j$ and all $\mathbf{u}^i$ except $\mathbf{u}^1$. The problem now becomes

$$\mathbf{u}^1 = \arg\min_{\mathbf{u} \in \mathbb{R}^k} \sum_{(1,j) \in \Omega} \left(X_{1j} - \langle \mathbf{u}, \mathbf{v}^j \rangle\right)^2 \ldots$$

Notice that summation is only over users that rated item 1 – usually small for most items

- Wait … this is just a ridge regression pro
- What happens if I fix all $\mathbf{u}^i$ and all $\mathbf{v}^j$ except $\mathbf{v}^1$

$$\mathbf{v}^1 = \arg\min_{\mathbf{v} \in \mathbb{R}^k} \sum_{(i,1) \in \Omega} \left(X_{i1} - \langle \mathbf{u}^i, \mathbf{v} \rangle\right)^2 + \lambda \cdot \|\mathbf{v}\|_2^2$$

This means that these problems can be solved cheaply

- Nice … again just a ridge regression problem!

Alternating Optimization Revisited

ALTERNATING OPTIMIZATION

1. Observed locations Ω and entries $\{X_{ij}: (i,j) \in \Omega\}$
2. Initialize $\mathbf{v}^{1,0}, \mathbf{v}^{2,0}, \ldots, \mathbf{v}^{n,0}$
3. For $t = 1,2, \ldots$
   1. Update the user vectors for $i = 1, \ldots, m$

      $$\mathbf{u}^{i,t} \leftarrow \left(\sum_{(i,j)\in\Omega} \mathbf{v}^{j,t-1} \left(\mathbf{v}^{j,t-1}\right)^\top + \lambda \cdot I \right)^{-1} \left(\sum_{(i,j)\in\Omega} X_{ij} \cdot \mathbf{v}^{j,t-1} \right)$$

   2. Update the item vectors for $j = 1, \ldots, n$

      $$\mathbf{v}^{j,t} \leftarrow \left(\sum_{(i,j)\in\Omega} \mathbf{u}^{j,t} \left(\mathbf{u}^{j,t}\right)^\top + \lambda \cdot I \right)^{-1} \left(\sum_{(i,j)\in\Omega} X_{ij} \cdot \mathbf{u}^{j,t} \right)$$

4. Repeat until convergence

Alternating Optimization Revisited

ALTERNATING OPTIMIZATION

1. Observed locations Ω and entries $\{X_{ij} : (i,j) \in \Omega\}$
2. Initialize $\mathbf{v}^{1,0}, \mathbf{v}^{2,0}, \ldots, \mathbf{v}^{n,0}$
3. For $t = 1,2, \ldots$
   1. Update the user vectors for $i = 1$

      $$\sum_{\Omega} \mathbf{v}^{j,t-1}\left(\mathbf{v}^{j,t-1}\right)^{\top} + \lambda \cdot I\Bigg)^{-1}\Bigg(\sum_{(i,j)\in\Omega} X_{ij} \cdot \mathbf{v}^{j,t-1}\Bigg)$$

   2. Update the item vectors for $j = 1, \ldots, n$

      $$\mathbf{v}^{j,t} \leftarrow \Bigg(\sum_{(i,j)\in\Omega} \mathbf{u}^{j,t}\left(\mathbf{u}^{j,t}\right)^{\top} + \lambda \cdot I\Bigg)^{-1}\Bigg(\sum_{(i,j)\in\Omega} X_{ij} \cdot \mathbf{u}^{j,t}\Bigg)$$

4. Repeat until convergence

Inexpensive for most items and users

$\mathcal{O}\left(k^3 + k^2(m+n)\right)$ time per iteration

Don't solve these RR problems optimally, just take an SGD step!

Can speed up even more

Alternating Optimization Revisited

SGD ALTERNATING OPTIMIZATION

1. Observed locations Ω and entries $\{X_{ij} : (i,j) \in \Omega\}$
2. Initialize $\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^m$ and $\mathbf{v}^1, \mathbf{v}^2, \dots, \mathbf{v}^n$
3. For $t = 1,2, \dots$
   1. Choose a random observed entry $(i^t, j^t) \in \Omega$
   2. Update the user i^t, and item j^t

      $$\mathbf{u}^{i^t} \leftarrow \mathbf{u}^{i^t} - 2\eta_t \cdot \left(\lambda \cdot \mathbf{u}^{i^t} - \left(X_{i^t j^t} - \left\langle \mathbf{u}^{i^t}, \mathbf{v}^{j^t} \right\rangle\right) \cdot \mathbf{v}^{j^t}\right)$$

      $$\mathbf{v}^{j^t} \leftarrow \mathbf{v}^{j^t} - 2\eta_t \cdot \left(\lambda \cdot \mathbf{v}^{j^t} - \left(X_{i^t j^t} - \left\langle \mathbf{u}^{i^t}, \mathbf{v}^{j^t} \right\rangle\right) \cdot \mathbf{u}^{i^t}\right)$$

   3. All other user and item vectors stay the same
4. Repeat until convergence

Alternating Optimization Revisited

Using simpler notation to avoid clutter

SGD ALTERNATING OPTIMIZ

1. Observed locations Ω and entries $\{X_{ij} : (i,j) \in \Omega\}$
2. Initialize $\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^m$ and $\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^n$
3. For $t = 1,2, \ldots$
   1. Choose a random observed entr
   2. Update the user i^t, and item j^t

      $$\mathbf{u}^{i^t} \leftarrow \mathbf{u}^{i^t} - 2\eta_t \cdot \left(\lambda \cdot \mathbf{u}^{i^t} - \left(X_{i^t j^t} - \left\langle \mathbf{u}^{i^t}, \mathbf{v}^{j^t} \right\rangle\right) \cdot \mathbf{v}^{j^t}\right)$$

      $$\mathbf{v}^{j^t} \leftarrow \mathbf{v}^{j^t} - 2\eta_t \cdot \left(\lambda \cdot \mathbf{v}^{j^t} - \left(X_{i^t j^t} - \left\langle \mathbf{u}^{i^t}, \mathbf{v}^{j^t} \right\rangle\right) \cdot \mathbf{u}^{i^t}\right)$$

   3. All other user and item vectors stay the same
4. Repeat until convergence

$\mathcal{O}(k)$ time per iteration

Pros and Cons

Pros

- Relies purely on behavioural data
- Does not require users to submit personal information
- Does not require sellers to submit item information
- Collaborative method: lazy users benefit from active users

Cons

- Relies purely on behavioural data
- Cannot utilize user and item information even if present
- Expensive to recompute matrix factorization again and again
- Adding new users and items not simple

Please give your Feedback

http://tinyurl.com/ml17-18afb