# Large-scale Robust Online Matching

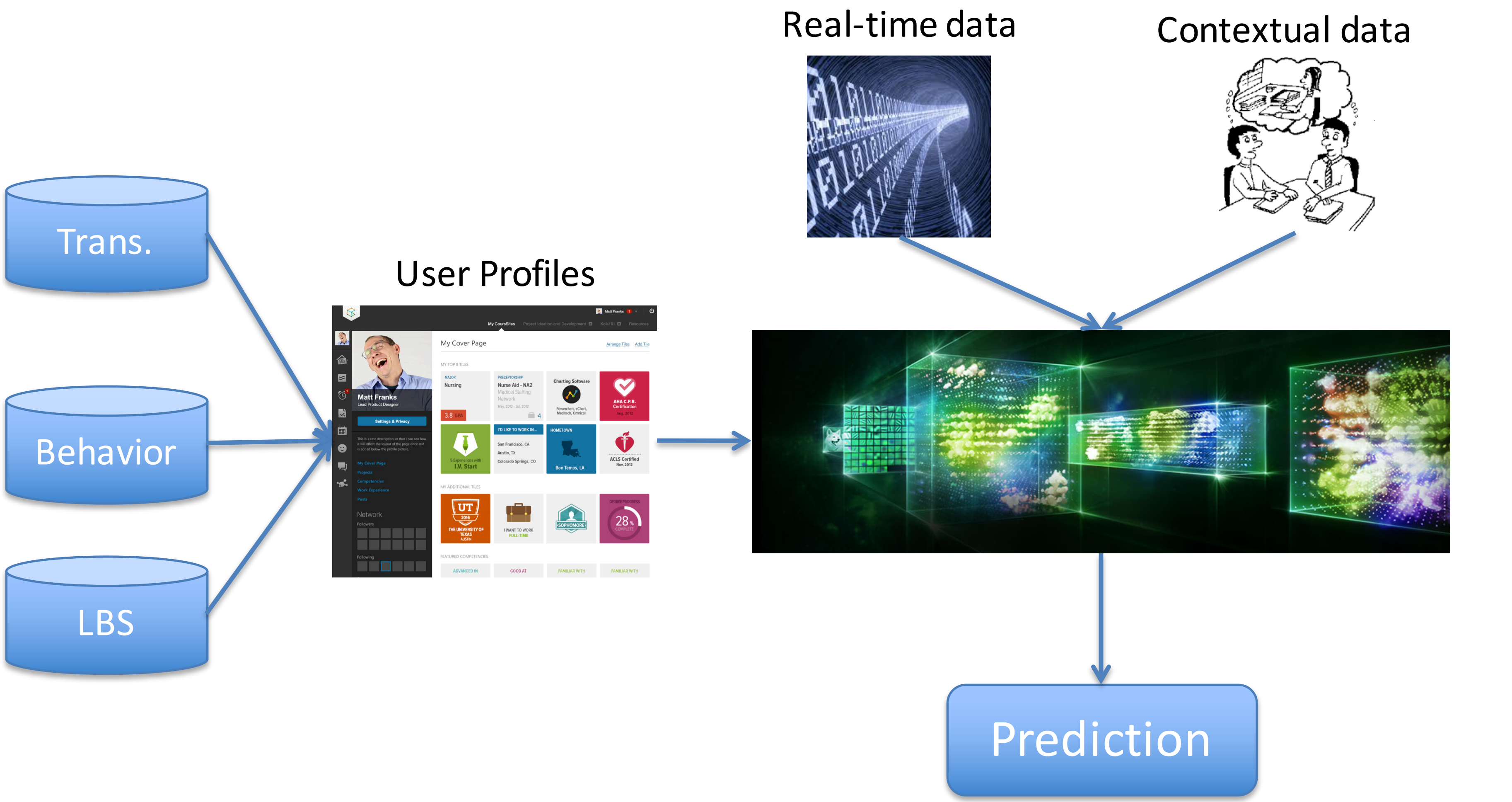
Rong Jin
Alibaba Group

# E-commerce

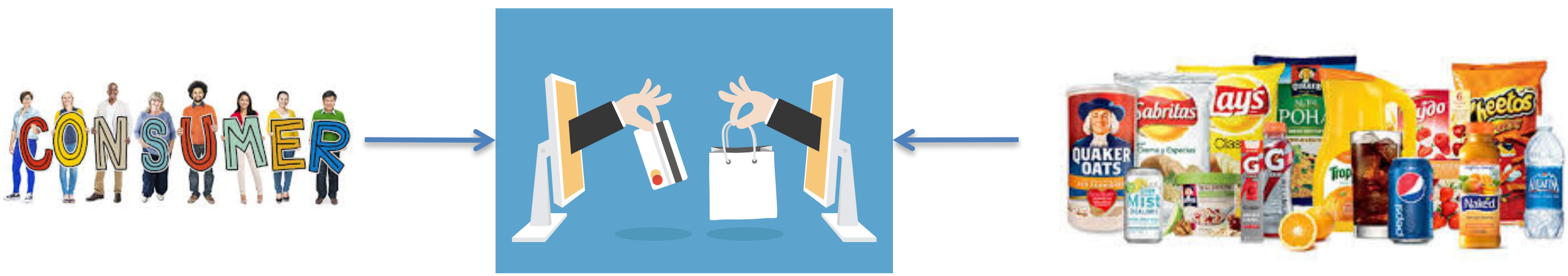
- Connect consumers with sellers/products
  - Efficiency is the key of any e-commerce platform

- Two key problems
  - Prediction: accurately predict users' needs
  - Matching: effectively match users' needs with products

# Accurately Predict User Needs



# Effectively Matching User Needs




Ease case: supplies ≥ demands

- A simple greedy matching works best
- Match each user with the best prediction

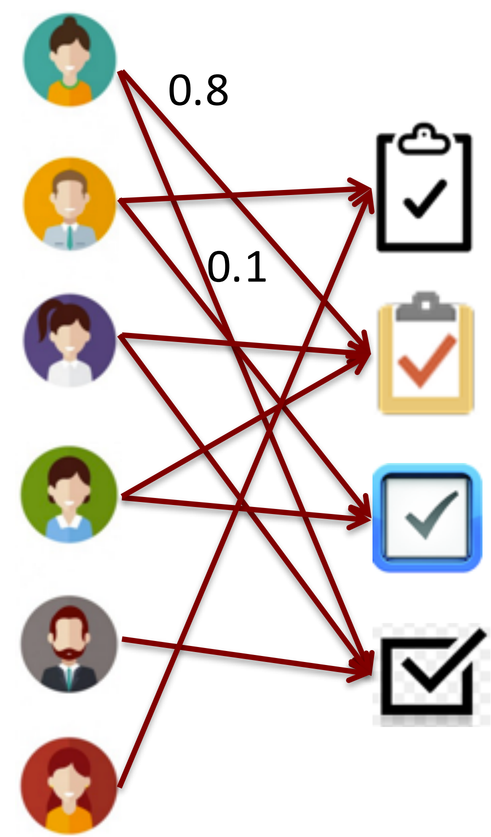
# Effectively Matching User Needs


Difficult case: supplies < demands

- How to decide which users will get what ?
- A greedy algorithm does not work, leading to the matching problem

# Assignment Problem [1]

- We are given a bipartite graph
  - m agents & n tasks
  - A different award is given when one agent is assigned to perform a task
- Optimal task assignment
  - Goal: maximize the overall awards
  - Constraints: each task can be assigned to a limited number of agents (or/and each agent is allowed to perform a limited number of tasks)



# Applications in Alibaba

- Online advertisement
  - Match users with different ads
  - Awards: the number of clicks
  - Constraints: budgeted number of impressions for individual ads

# Ask-All (问大家)

## Basic

- Allow customers to raise questions about products before making purchases
- The system distributes questions to users who are likely to provide useful answers within a short period of time

## Goal

- Match each question with a number of appropriate users

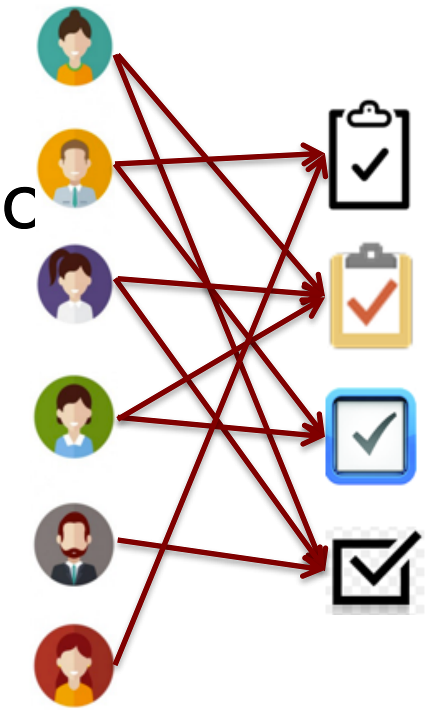
## Constraints

- Only a limited number of questions answered by each user

# Example of Ask-All

# Other Applications in Alibaba

- Mobile message push
- Online traffic allocation
- Online distribution of coupons
- Stock aware online recommendatic
- Online assignment of delivery requests



# Assignment as a LP Problem [1]

- $n$: number of agents
- $m$: number of tasks
- $c_{i,j}$: estimated award when assigning agent $i$ to task $j$
- $K_i$: maximum number of tasks assigned to agent $i$

$$\max_{x \in \{0,1\}^{n \times m}} \sum_{i=1}^{n} \sum_{j=1}^{m} x_{i,j} c_{i,j}$$
$$\text{s. t.} \quad \sum_{j=1}^{m} x_{i,j} \leq K_i, \; i \in [n]$$
$$\sum_{i=1}^{n} x_{i,j} = 1, \; j \in [m]$$

**Linear Relaxation** →

$$\max_{x \in [0,1]^{n \times m}} \sum_{i=1}^{n} \sum_{j=1}^{m} x_{i,j} c_{i,j}$$
$$\text{s. t.} \quad \sum_{j=1}^{m} x_{i,j} \leq K_i, \; i \in [n]$$
$$\sum_{i=1}^{n} x_{i,j} = 1, \; j \in [n]$$

- Apply rounding procedure to turn fractionals into integers

# Assignment as a LP Problem [1]

$$\max_{x \in \{\mathbf{0},\mathbf{1}\}^{n\times m}} \sum_{i=1}^{n}\sum_{j=1}^{m} x_{i,j}\mathbf{c_{i,j}}$$

$$\text{s. t.} \quad \sum_{j=1}^{m} x_{i,j} \le K_i, \;\; i \in [n]$$

$$\sum_{i=1}^{n} x_{i,j} = 1, \;\; j \in [m]$$

Linear Relaxation

$$\max_{x \in [\mathbf{0},\mathbf{1}]^{n\times m}} \sum_{i=1}^{n}\sum_{j=1}^{m} x_{i,j}\mathbf{c_{i,j}}$$

$$\text{s. t.} \quad \sum_{j=1}^{m} x_{i,j} \le K_j, \;\; i \in [n]$$

$$\sum_{i=1}^{n} x_{i,j} = 1, \;\; j \in [n]$$

- Problem 1
  - Solutions are too sensitive to estimated quantities $\{c_{i,j}\}$
- Problem 2
  - Unable to handle online requests (i.e. emerging tasks)

# Large-scale Online Robust Matching

- Robust matching
  - **Robust optimization** with application to online display ads
  - Stable matching with application to mobile message push
- Online matching
  - Online algorithm with application to Ask-All
  - Online constrained optimization with application to online traffic allocation

# Robust Optimization [2]

- Take into account the uncertainty in $\{c_{i,j}\}$ when solving the optimization problem

$$
g(c) = \begin{array}{lll} \max\limits_{x \in [\mathbf{0,1}]^{n \times m}} & \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{m} x_{i,j} c_{i,j} \\ \text{s. t.} & \sum\limits_{j=1}^{m} x_{i,j} = K_i, \; i \in [n] \\ & \sum\limits_{i=1}^{n} x_{i,j} = 1, \; j \in [m] \end{array}
\quad + \quad
\begin{array}{c} c_{1,1} \in [a_{1,1}, b_{1,1}] \\ c_{1,2} \in [a_{1,2}, b_{1,2}] \\ \ldots \\ c_{n,n} \in [a_{n,n}, b_{n,n}] \end{array}
$$

$$
\begin{array}{ll} \min\limits_{c} & g(c) \\ \text{s. t.} & c_{i,j} \in [a_{i,j}, b_{i,j}], \; \forall (i,j) \end{array}
$$

**Minimax optimization**

# Online Display Ads [4]

- Advertiser
  - Market its products
- Platform
  - Attract enough traffic
- User
  - Find products/service

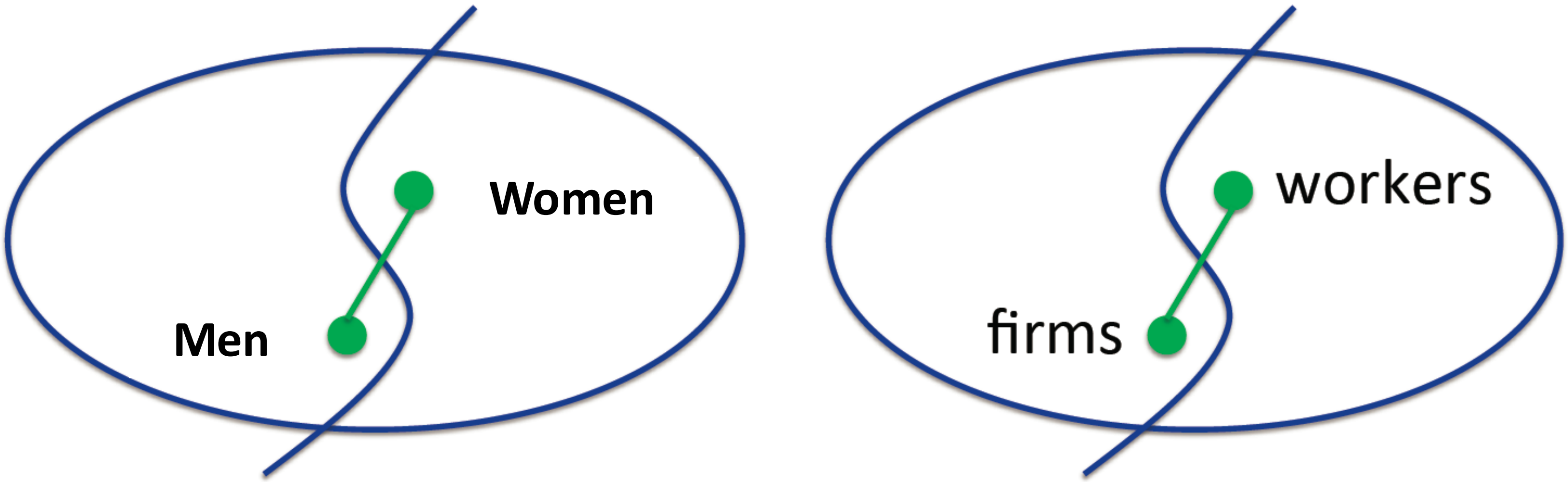
- Experiment
  - 10K+ advertisers
  - 100 million users
  - Improve revenue by 20+% with robust optimization

# Large-scale Online Robust Matching

- Robust matching
  - Robust optimization with application to online display ads
  - **Stable matching** with application to mobile message push

- Online matching
  - Primal and dual based online algorithm with application to 问大家
  - Online constrained optimization with application to online traffic allocation

# Two-sided Matching [5]




**One-to-one matching**
**(stable marriage)**

**One-to-many matching**
**(college admission)**

- Fundamental to many economy problems

# Stable Marriage Model [5]

- Men $M=\{m_1,\ldots,m_n\}$
- Women $W=\{w_1,\ldots,w_p\}$
- Each man and woman specify his/her preference list

$P(m_1) = w_2, w_1, w_3$ $P(w_1) = m_1, m_3, m_2$

$P(m_2) = w_1, w_3, w_2$ $P(w_2) = m_3, m_1, m_2$

$P(m_3) = w_1, w_2, w_3$ $P(w_3) = m_1, m_3, m_2$

# Stable Marriage Model [5]

$P(m_1) = \boxed{w_2, w_1,}\ w_3$ $\quad$ $P(w_1) = (m_1, m_3, m_2)$

$P(m_2) = w_1, w_3, w_2$ $\quad$ $P(w_2) = (m_3, \boxed{m_1, m_2})$

$P(m_3) = w_1, w_2, w_3$ $\quad$ $P(w_3) = (m_1, m_3, m_2)$

- Matching $\mu$: $(m_1, w_1), (m_2, w_2), (m_3, w_3)$
- Blocking pair: $(m_1, w_2)$
- A matching is **stable** if it is not blocked by any pair

- Deferred-Acceptance (DA) algorithm finds one of the best matchings within all stable matchings

# Advantages of Two-sided Matching

- It is robust
  - only depends on preference lists
- It is efficient ($O(n^2)$ vs. $O(n^3)$)
- It is optimal among all stable matchings

- Let $\mu^*$ be the optimal solution found by the two-sided matching algorithm. We have

$$f(\mu^*) \geq \left(1 - \frac{1}{e}\right) f^*$$

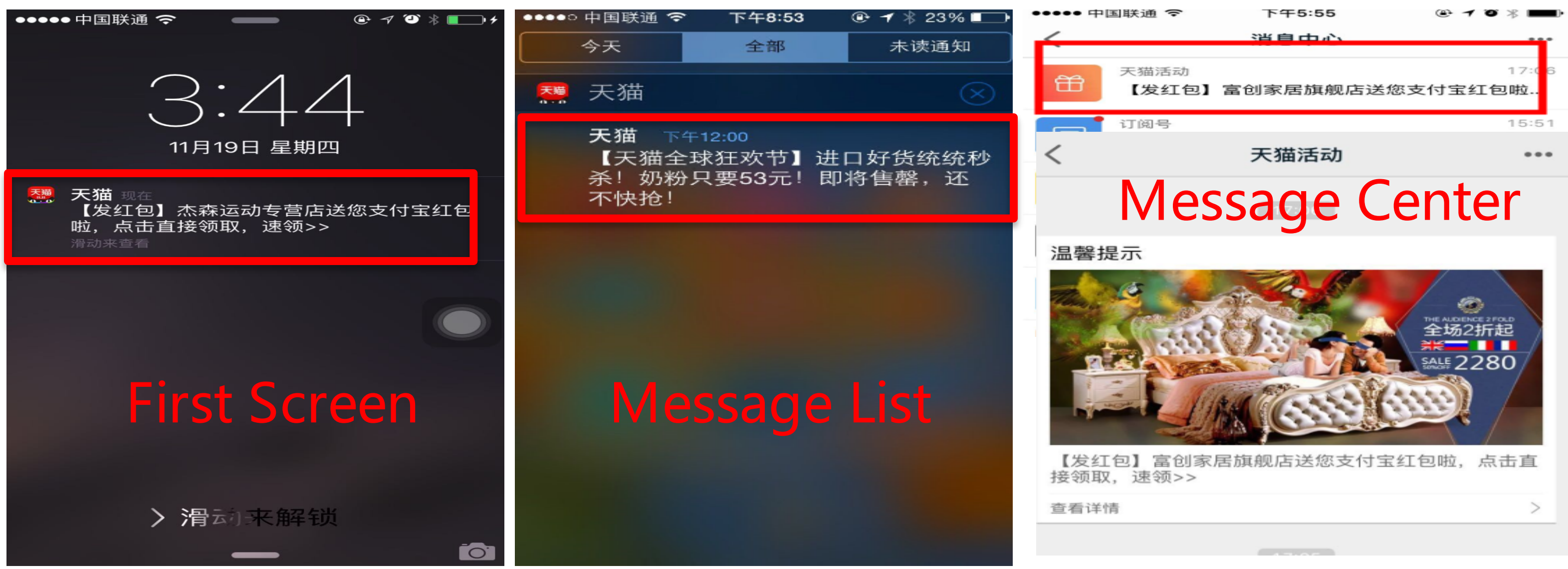
$$\boldsymbol{f^*} = \max_{x \in \{\mathbf{0},\mathbf{1}\}^{n \times n}} \quad f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{i,j} c_{i,j}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{i,j} = 1, \; i \in [n]$$

$$\sum_{i=1}^{n} x_{i,j} = 1, \; j \in [n]$$

# Mobile Message Push

- Activate app users by pushing them appropriate mobile messages (items, 红包, coupons)
- Experiment
  - 400 messages, 10 million users
  - improves opening rate by 20% compared to robust optimization



# Large-scale Online Robust Matching

- Robust matching
  - Robust optimization with application to online display ads
  - Stable matching with application to mobile message push

- **Online matching**
  - Primal and dual based online algorithm with application to Ask-All (问大家)
  - Online constrained optimization with application to online traffic allocation

# Online Matching

- Tasks are created in an online fashion
- Each assignment decision has to be made online

$$\max_{x\in\{\mathbf{0},\mathbf{1}\}^{n\times T}} \sum_{i=1}^{n}\sum_{t=1}^{T} x_{i,t}c_{i,t}$$

$$\text{s. t.} \quad \sum_{t=1}^{T} x_{i,t} \le K_i,\ i\in[n]$$

$$\sum_{i=1}^{n} x_{i,t} = m,\ t\in[T]$$

# Online Matching

- Tasks are created in an online fashion
- Each assignment decision has to be made online

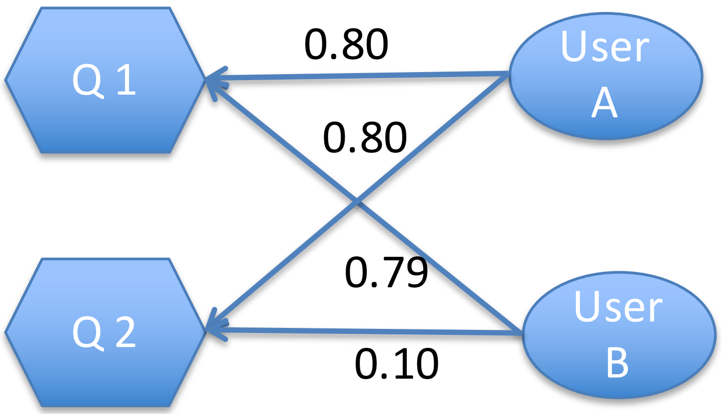
$$\max_{x \in \{0,1\}^{n \times T}} \sum_{i=1}^{n} \sum_{t=1}^{T} x_{i,t} c_{i,t}$$

$$\text{s. t.} \quad \sum_{t=1}^{T} x_{i,t} \leq K_i, \; i \in [n]$$

$$\sum_{i=1}^{n} x_{i,t} = M, \; t \in [T]$$

## Greedy approach

- For each emerging task, select the first *M* available agents with the largest awards



# Online Matching

- Tasks are created in an online fashion
- Each assignment decision has to be made online

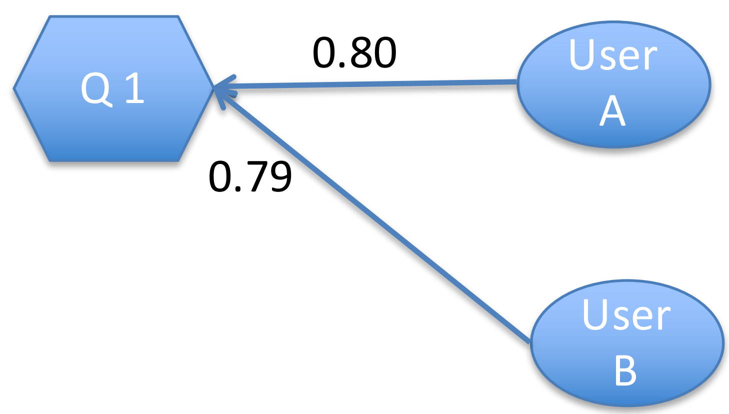
$$\max_{x\in\{0,1\}^{n\times T}} \sum_{i=1}^{n}\sum_{t=1}^{T} x_{i,t}c_{i,t}$$

$$\text{s. t.} \quad \sum_{t=1}^{T} x_{i,t} \le K_i,\ i \in [n]$$

$$\sum_{i=1}^{n} x_{i,t} = M,\ t \in [T]$$

## Greedy approach

- For each emerging task, select the first *M* available agents with the largest awards



# Online Matching

- Tasks are created in an online fashion
- Each assignment decision has to be made online

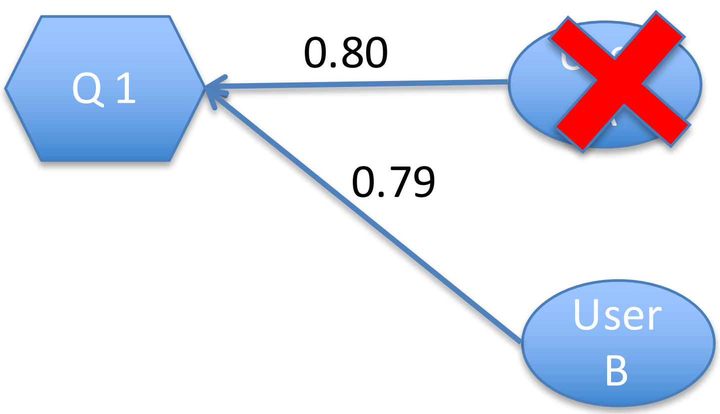
$$\max_{x \in \{0,1\}^{n \times T}} \sum_{i=1}^{n} \sum_{t=1}^{T} x_{i,t} c_{i,t}$$

$$\text{s. t.} \quad \sum_{t=1}^{T} x_{i,t} \le K_i, \ i \in [n]$$

$$\sum_{i=1}^{n} x_{i,t} = M, \ t \in [T]$$

Greedy approach

- For each emerging task, select the first *M* available agents with the largest awards



# Online Matching

- Tasks are created in an online fashion
- Each assignment decision has to be made online

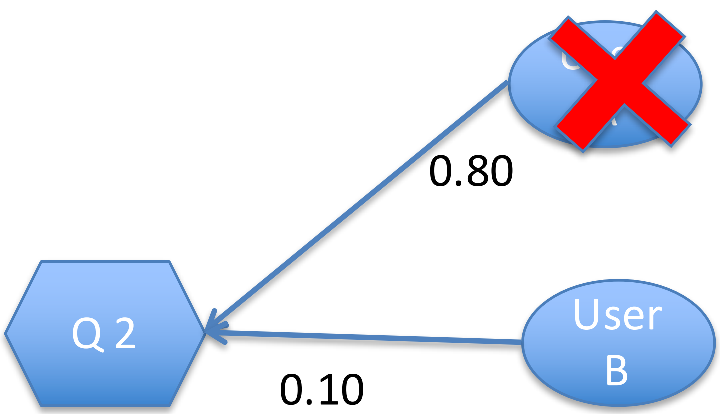
$$\max_{x \in \{0,1\}^{n \times T}} \sum_{i=1}^{n} \sum_{t=1}^{T} x_{i,t} c_{i,t}$$

$$\text{s. t.} \quad \sum_{t=1}^{T} x_{i,t} \leq K_i, \; i \in [n]$$

$$\sum_{i=1}^{n} x_{i,t} = M, \; t \in [T]$$

## Greedy approach

- For each emerging task, select the first *M* available agents with the largest awards



# Online Matching

- Tasks are created in an online fashion
- Each assignment decision has to be made online

$$\max_{x \in \{\mathbf{0},\mathbf{1}\}^{n \times T}} \quad \sum_{i=1}^{n} \sum_{t=1}^{T} x_{i,t} c_{i,t}$$

$$\text{s. t.} \quad \sum_{t=1}^{T} x_{i,t} \leq K_i, \ i \in [n]$$

$$\sum_{i=1}^{n} x_{i,t} = M, \ t \in [T]$$

Online algorithm [8]

- Based on primal dual theory
- **Discount awards** of agent *i* after it is assigned to perform a task
- Choose users based on discounted awards

# Online Matching Algorithm

- Tasks are created in an online fashion
- Each assignment decision has to be made online

$$\max_{x \in \{\mathbf{0},\mathbf{1}\}^{n \times T}} \sum_{i=1}^{n} \sum_{t=1}^{T} x_{i,t} c_{i,t}$$

$$\text{s. t.} \quad \sum_{t=1}^{T} x_{i,t} \leq K_i, \; i \in [n]$$

$$\sum_{i=1}^{n} x_{i,t} = M, \; t \in [T]$$

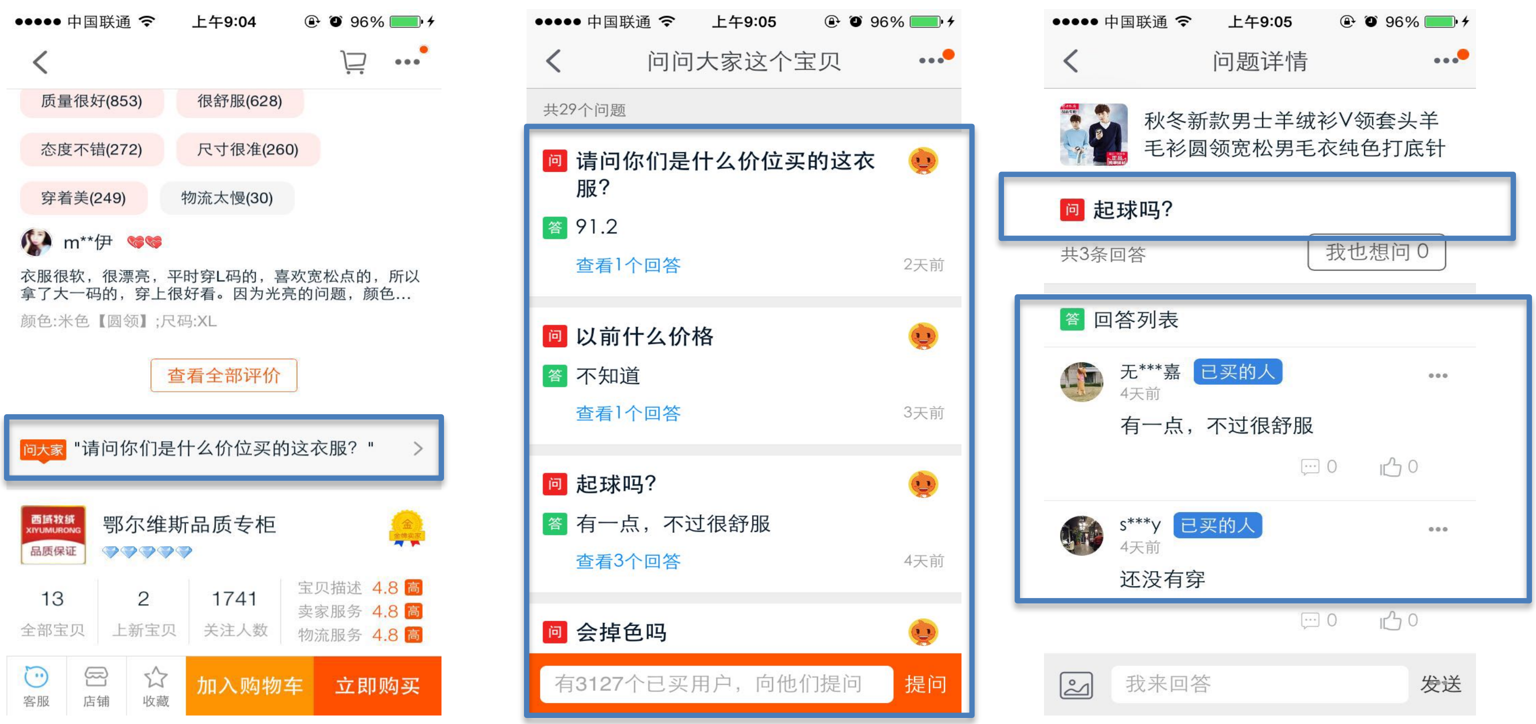
**Theorem**

Let f and f* be the awards received by the online matching algorithm and the optimal matching result, respectively. We have

$$f \geq \left(1 - \frac{1}{Z}\right) f^*$$

where Z is close to e when n is large

# Ask-All (问大家)




- Experiment
  - 20 million users, 500K questions per day
  - Online algorithm improves the answering rate of the greedy algorithm by 40%

# Online Constrained Optimization

- Limitation of online algorithm
  - f – f* can be very large
  - Limited to linear award functions

- Online optimization with *long term constraints*

$$\max_{x \in \{\mathbf{0},\mathbf{1}\}^{n \times T}} \sum_{i=1}^{n} \sum_{t=1}^{T} x_{i,t} c_{i,t}$$

$$\text{s. t.} \quad \sum_{t=1}^{T} x_{i,t} \leq K_i, \ i \in [n]$$

$$\sum_{i=1}^{n} x_{i,t} = M, \ t \in [T]$$

Generalize $\Rightarrow$

$$\min_{\mathbf{x}_t \in \mathbb{R}^d, t \in [T]} \sum_{t=1}^{T} f_t(\mathbf{x}_t)$$

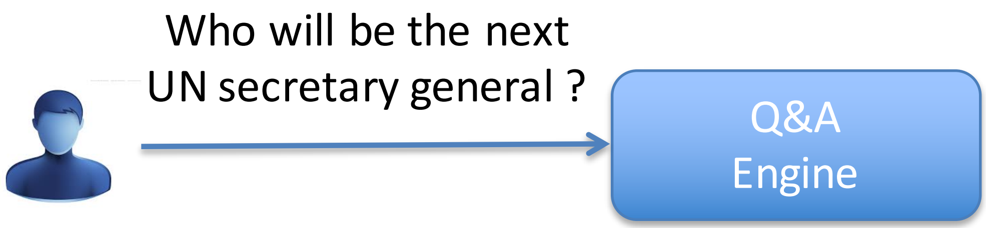
$$\text{s. t.} \quad \sum_{t=1}^{T} A\mathbf{x}_t \leq \mathbf{b}$$

# Online Constrained Optimization

- Our approach
  - Based on online convex-concave optimization
  - Performance guarantee $\max(R_T, C_T) \leq O\left(\Sigma_T + \sqrt{T}\right)$
- Application to Constrained Traffic Optimization
  - Optimize traffic efficiency under the constraints of traffic guarantee for merchants
  - Less than 1% constraint violation
  - Improve CTR by 100% compared to a greedy approach

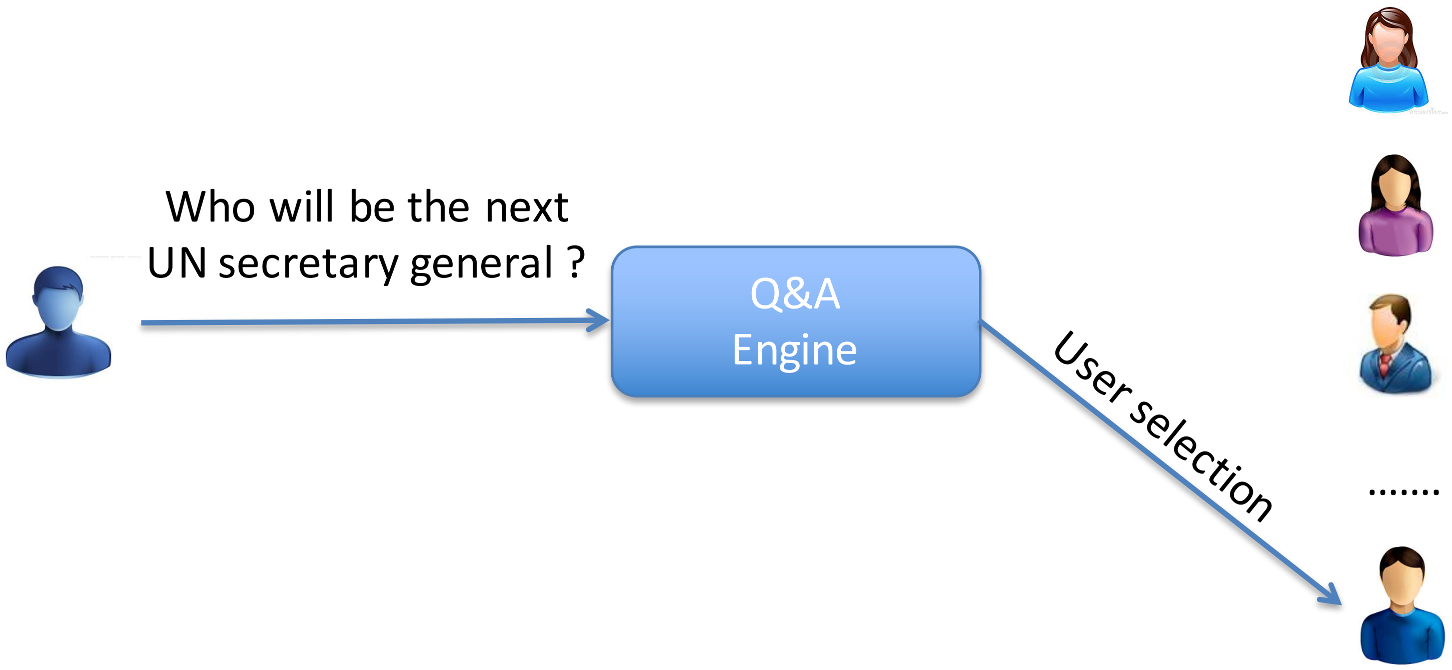
# Collaborative Intelligence

- Beyond machine intelligence



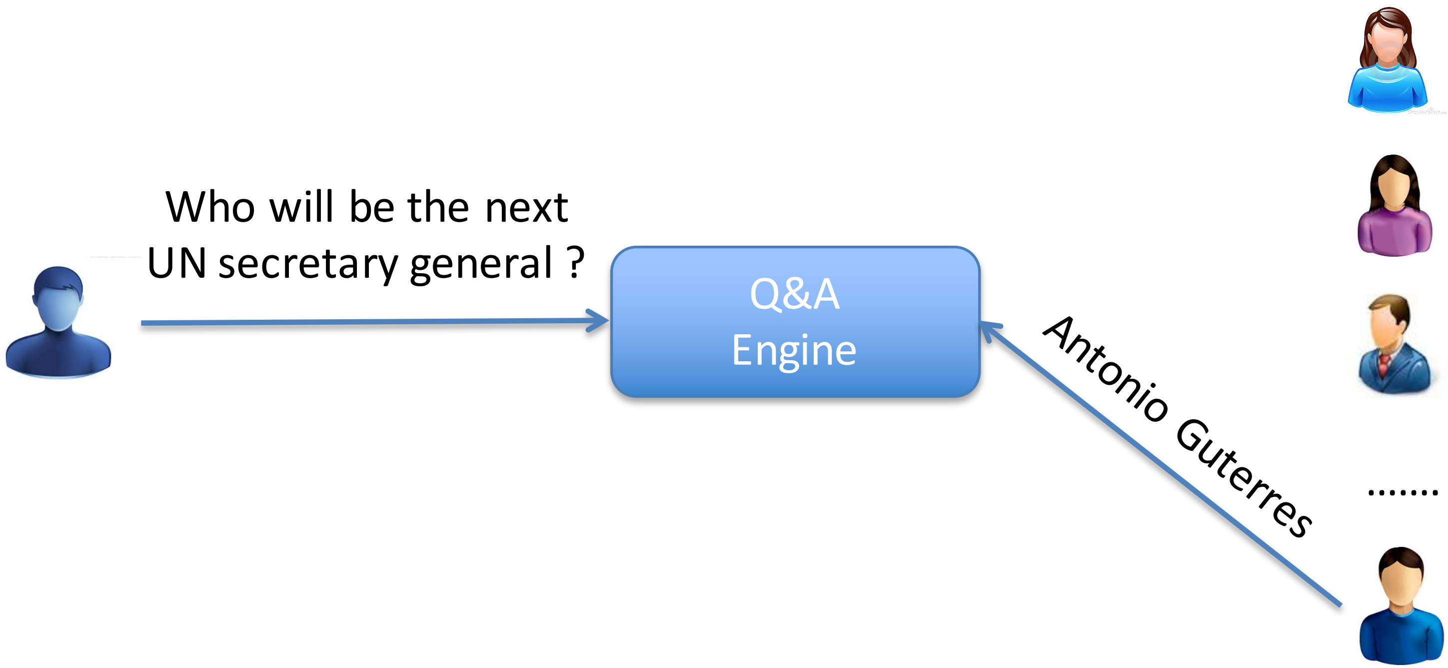
# Collaborative Intelligence

- Beyond machine intelligence
- We have a few **billion** users, and every user is an **Intelligent Agent**



# Collaborative Intelligence

- Beyond machine intelligence
- We have a few **billion** users, and every user is an **Intelligent Agent**



# Reference


[1] R. A. Beezer, Assignment Problems, SIAM Review 53(1): 180-181, 2011

[2] A. Ben-Tal, L. El Ghaoui, and A. S. Nemirovski, Robust optimization, 2009

[3] Y. Nesterov. Introductory lectures on convex optimization: A basic course, 2004

[4] B. Croft, D. Metzler, and T. Strohman, Search Engines: Information Retrieval in Practice, 2009

[5] A. E. Roth and M. Sotomayor. Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis, 1990

[6] B. Kalyanasundaram and K. R. Pruhs, An optimal deterministic algorithm for online b-matching, Theoretical Computer Science 233(1): 319–325, 2000

[7] M. Baïou and M. Balinski, The stable admissions polytope, Math. Program 87: 427-439, 2000

[8] N. Buchbinder and J. Naor, The Design of Competitive Online Algorithms via a Primal: Dual Approach, Foundations and Trends in Theoretical Computer Science 3(2): 93-263, 2009

# Thanks