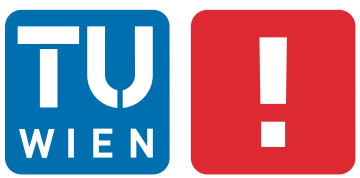
Cooperative Optimization Approaches for Distributing Service Points: Extension to a Large Number of Users

Thomas Jatschka, Günther Raidl, Tobias Rodemann

PhD Seminar, April 3, 2019



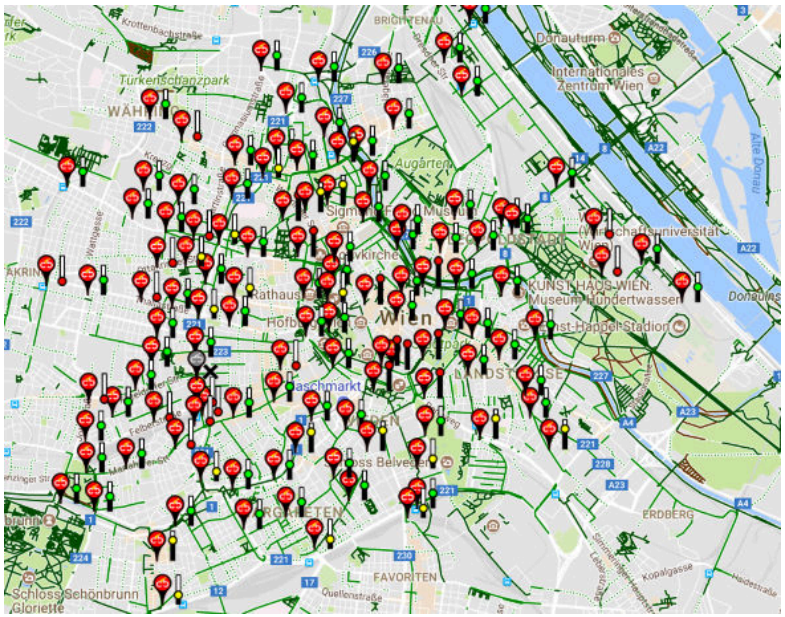
Motivation

Goal: find an optimal set of locations within a certain geographical area for placing service points

Motivation

Goal: find an optimal set of locations within a certain geographical area for placing service points

- for mobility purposes:
  - bike sharing stations
  - rental stations for car sharing
  - charging stations for electric vehicles
  - ...



Where to place service points?

Where to place service points?

estimate demands upfront using

- demographic data,
- geographic data,
- information on public transport and the street network,
- and knowledge on diverse special locations,
- and surveys of potential customers
- ...

Where to place service points?

estimate demands upfront using

- demographic data,
- geographic data,
- information on public transport and the street network,
- and knowledge on diverse special locations,
- and surveys of potential customers
- ...

$\Rightarrow$ challenging and error-prone

Motivation

the actual usage of a service system by the user depends on

- service points on a few specific locations
- non-trivial relationships of the user's necessities and preferences in conjunction with larger parts of the whole service system

Motivation

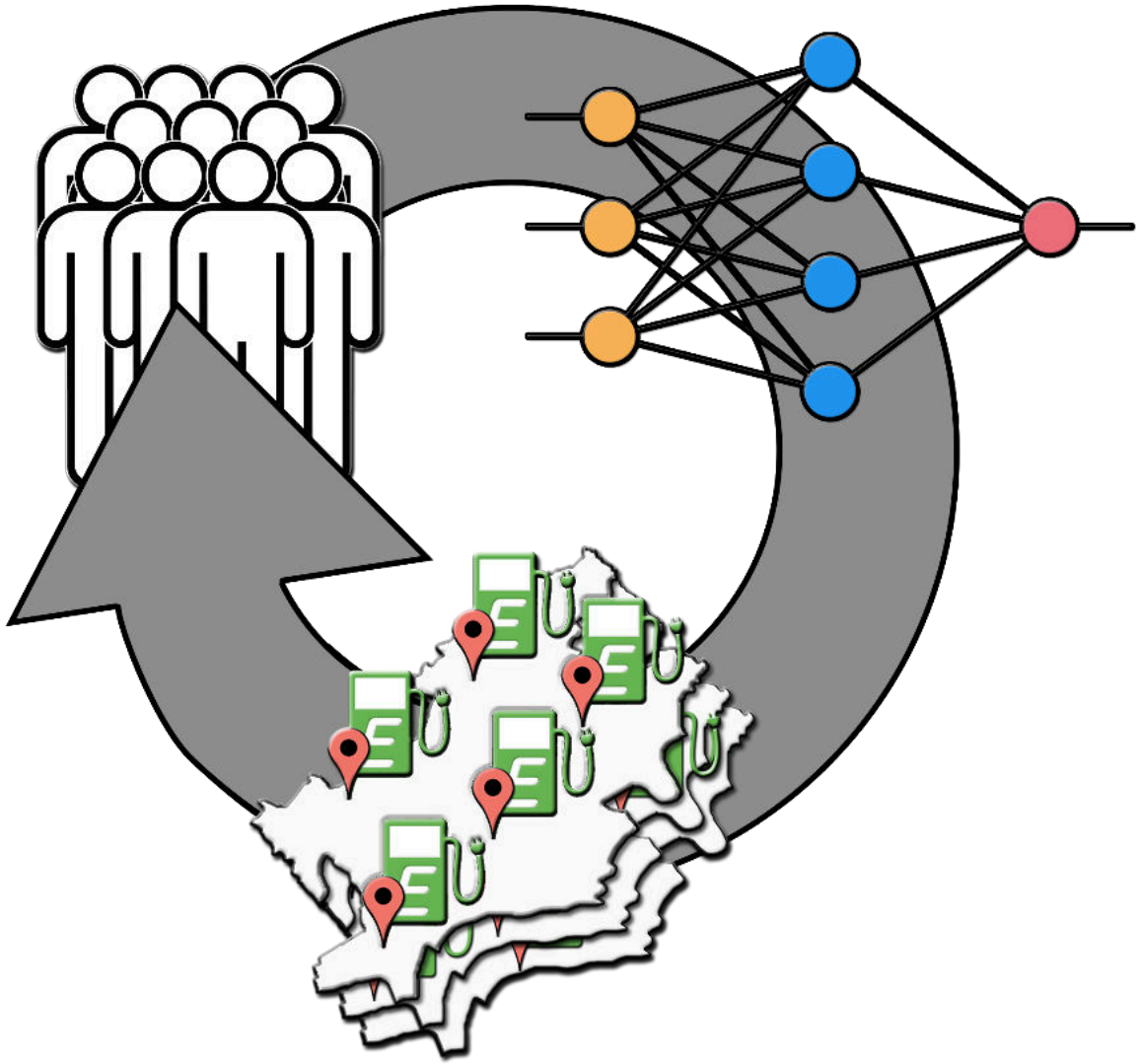
the actual usage of a service system by the user depends on

- service points on a few specific locations
- non-trivial relationships of the user's necessities and preferences in conjunction with larger parts of the whole service system

Assumption

We are in general not able to obtain complete information from a possible user about the conditions on which how much of his potential demand will be fulfilled!

Cooperative Optimization Approach (COA)



We are given

- a set of locations $V = \{1, \ldots, n\}$ at which service points may be built,
- a set of potential users $U = \{1, \ldots, m\}$,
- building costs c_v and maintenance costs z_v for each location $v \in V$
- a maximum budget B for building service points,
- and a prize q that is earned for each unit of satisfied customer demand

Objective Function

$$\max f(x) = \max \left(q \cdot \sum_{u \in U} \sum_{v \in V} d(u, v, x) - \sum_{v \in V} z_v x_v \right)$$

$d(u, v, x)$ = demand of user $u \in U$ fulfilled at service point $v \in V$ in solution x

Objective Function

$$\max f(x) = \max \left(q \cdot \sum_{u \in U} \sum_{v \in V} d(u, v, x) - \sum_{v \in V} z_v x_v \right)$$

$d(u, v, x)$ = demand of user $u \in U$ fulfilled at service point $v \in V$ in solution x

$\Rightarrow$ can only be determined by directly asking the user

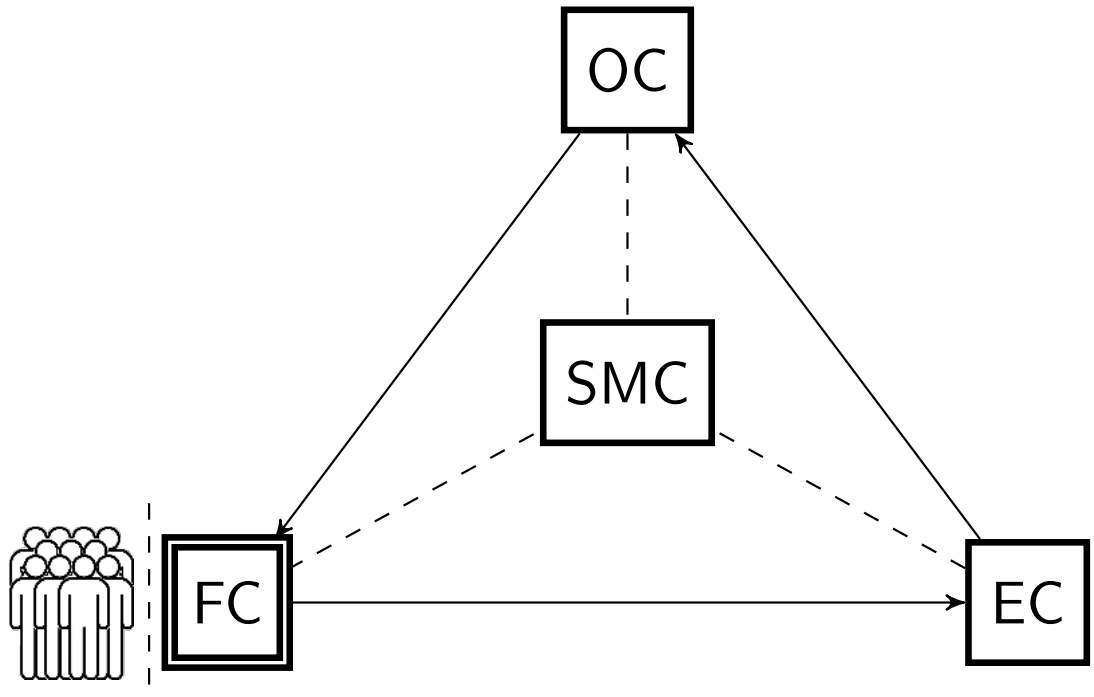
- it is not reasonable for a user to evaluate every generated solution
- it is in practice not possible to explicitly express $d(u, v, x)$

- it is not reasonable for a user to evaluate every generated solution
- it is in practice not possible to explicitly express $d(u, v, x)$

$\Rightarrow$ surrogate function $\tilde{d}(u, v, x)$

Cooperative Optimization Approach (COA)

Framework



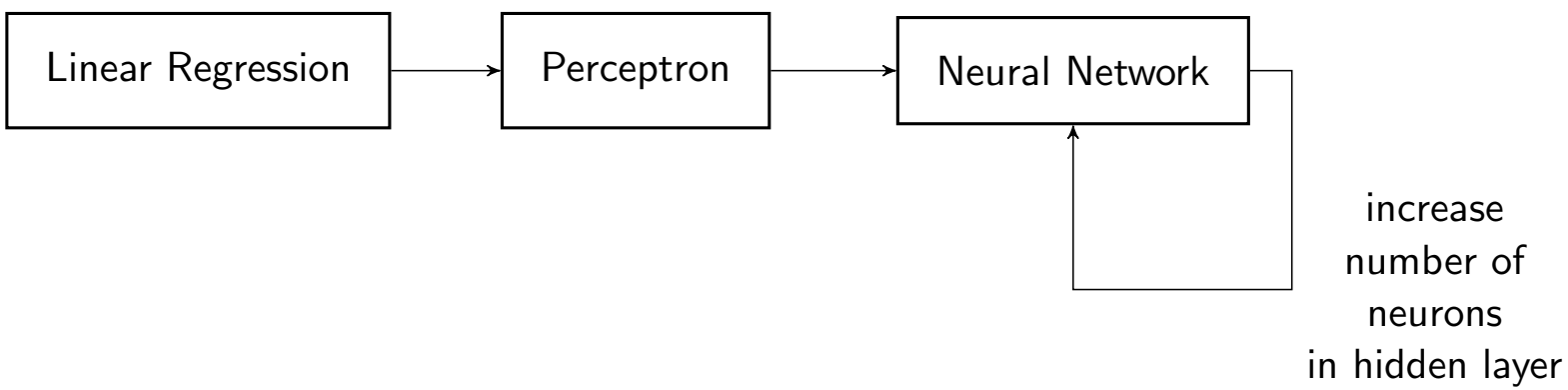
Evaluation Component

Solution:

$$x = (x_v)_{v \in V}$$

Surrogate function:

$$\tilde{d}(u, v, x) = \begin{cases} 0 & \text{if } x_v = 0 \vee v \notin V_u \\ \max(0, g_{u,v}(x)) & \text{else} \end{cases}$$



Previous Results

n	m	$\overline{\%\text{-gap}}$	$\sigma_{\%\text{-gap}}$	$\overline{\%\text{-}\Delta\tilde{f}}$	$\sigma_{\%\text{-}\Delta\tilde{f}}$	t[s]
50	50	0.20	0.45	0.17	0.17	2662.28
50	60	0.06	0.18	0.21	0.17	2643.06
50	70	0.09	0.44	0.27	0.22	3763.76
50	80	0.04	0.15	0.27	0.22	3919.22
50	90	0.19	0.71	0.39	0.25	4516.27
50	100	0.02	0.08	0.37	0.30	4994.73
60	50	0.05	0.11	0.26	0.22	2944.35
70	50	0.13	0.43	0.25	0.16	3657.66
80	50	0.11	0.28	0.26	0.35	4809.90
90	50	0.26	0.65	0.24	0.31	5435.47
100	50	0.07	0.15	0.19	0.19	7196.73

Bottlenecks

- Surrogate function: a large number of individual machine learning models $g_{u,v}$
- Optimization:
  - black-box optimization with neighborhood in $\mathcal{O}(n^2)$
  - no incremental evaluation
- Users have to evaluate a lot of solutions

Observations

- users have certain use cases, e.g.,going to work, to a recreational facility, shopping, etc.
- each use case associated with
  - some demand/usage frequency,
  - and one or more points of interests (POIs), i.e., the location of the work, recreational facility, shopping center, etc.
- users prefer some service point locations over others w.r.t. a POI $\Rightarrow$ ranking of service point locations
- it is unlikely that two users have the same needs in all respects
- however, we assume that many users have common POIs

$\Rightarrow$ users may rank locations similarly w.r.t. a POI

The Service Point Distribution Problem (SPDP)

Problem Formalization

We are given

- a set of locations $V = \{1, \ldots, n\}$ at which service points may be built,
- a set of potential users $U = \{1, \ldots, m\}$,
- building costs c_v and maintenance costs z_v for each location $v \in V$,
- a maximum budget B for building service points,
- and a prize q that is earned for each unit of satisfied customer demand

Ask from users:

- set of use cases E_u for each user u, where each use case $e \in E_u$ corresponds to a set of POIs,
- and use case demands $D_{u,e}$.

Ask from users:

- set of use cases E_u for each user u, where each use case $e \in E_u$ corresponds to a set of POIs,
- and use case demands $D_{u,e}$.

$\Rightarrow$ ranking of service point locations?

The Service Point Distribution Problem (SPDP)

Retrieving ranking of service point locations

- rating instead of ranking

The Service Point Distribution Problem (SPDP)

Retrieving ranking of service point locations

- rating instead of ranking
- Ask users to provide rating for all of their relevant locations?

The Service Point Distribution Problem (SPDP)

Retrieving ranking of service point locations

- rating instead of ranking
- Ask users to provide rating for all of their relevant locations?
  - tedious for users
  - users may forget locations
  - users may not consider locations initially
  - might not be necessary if we can accurately predict them

- present a set of locations s to user
- ask user to specify/rate best service point for each of his POIs w.r.t. $s \Rightarrow w(u, v, p)$
- rating $w(u, v, p)$: integer from zero to four

The Service Point Distribution Problem (SPDP)

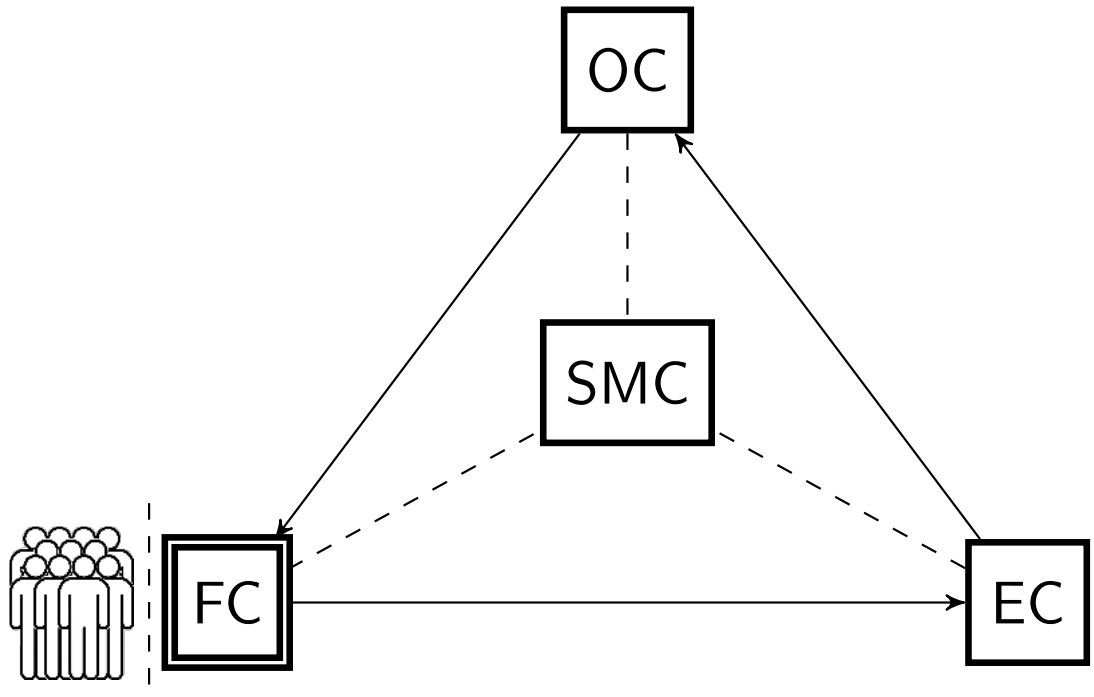
Objective Function

$$f(x) = q \cdot \sum_{u \in U} \sum_{e \in E_u} D_{u,e} \cdot \frac{\min_{p \in e} \left(\max_{v \in V} w(u, v, p, x) \right)}{4} - \sum_{v \in V} z_v^{\text{var}} x_v,$$

$$w(u, v, p, x) = w(u, v, p) \cdot x_v$$

Cooperative Optimization Approach (COA)

Framework



Theorem

Given an $m \times n$ matrix W, then there exists a factorization of the form

$$W = A\Sigma Q^T$$

where

- *A is an $m \times k$ matrix,*
- *Σ is a $k \times k$ diagonal matrix,*
- *and Q is a $k \times n$ matrix.*

"merge" Σ into one of the other matrices: $W = AQ^T$

Evaluation Component (EC)

Singular Value Decomposition (SVD) - Movie Ratings

$$W = \begin{array}{c} \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \end{array} \begin{pmatrix} 4 & 2 & 5 & 2.5 \\ 4 & 0 & 4 & 2 \\ 1.5 & 3 & 3 & 1.5 \\ 4.5 & 1 & 5 & 2.5 \\ 1.75 & 2.5 & 2.5 & 1.75 \end{pmatrix}$$

M_1 M_2 M_3 M_4 M_5

Evaluation Component (EC)

Singular Value Decomposition (SVD) - Example

$$
\begin{array}{c}
\\
M_1 \\
M_2 \\
M_3 \\
M_4 \\
M_5
\end{array}
\begin{pmatrix}
4 & 2 & 5 & 2.5 \\
4 & 0 & 4 & 2 \\
1.5 & 3 & 3 & 1.5 \\
4.5 & 1 & 5 & 2.5 \\
1.75 & 2.5 & 2.5 & 1.75
\end{pmatrix}
$$

Evaluation Component (EC)

Singular Value Decomposition (SVD) - Example

$$\begin{array}{c} \\ F_1 \\ F_2 \end{array} \begin{pmatrix} 1 & 0 & 1 & 0.5 \\ 0.5 & 1 & 1 & 0.5 \end{pmatrix}$$

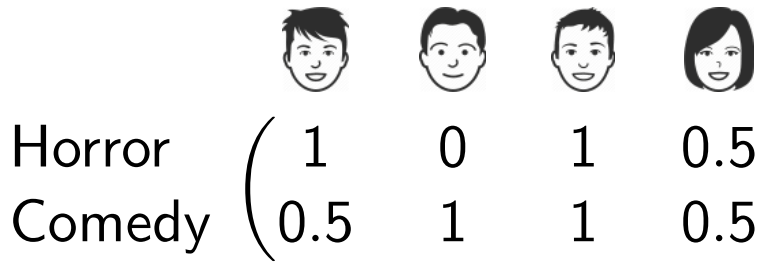
$$\begin{array}{c} \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \end{array} \begin{pmatrix} F_1 & F_2 \\ 3 & 2 \\ 4 & 0 \\ 0 & 3 \\ 4 & 1 \\ 0 & 2.5 \end{pmatrix}$$

$$\begin{array}{c} \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \end{array} \begin{pmatrix} 4 & 2 & 5 & 2.5 \\ 4 & 0 & 4 & 2 \\ 1.5 & 3 & 3 & 1.5 \\ 4.5 & 1 & 5 & 2.5 \\ 1.75 & 2.5 & 2.5 & 1.75 \end{pmatrix}$$

$F_1, F_2 \ldots$ hidden features

Evaluation Component (EC)

Singular Value Decomposition (SVD) - Example



$$\begin{array}{l} \text{Horror} \\ \text{Comedy} \end{array} \begin{pmatrix} 1 & 0 & 1 & 0.5 \\ 0.5 & 1 & 1 & 0.5 \end{pmatrix}$$

$$\begin{array}{c} \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \end{array} \begin{array}{c} \begin{array}{cc} \text{Horror} & \text{Comedy} \end{array} \\ \begin{pmatrix} 3 & 2 \\ 4 & 0 \\ 0 & 3 \\ 4 & 1 \\ 0 & 2.5 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \end{array} \begin{pmatrix} 4 & 2 & 5 & 2.5 \\ 4 & 0 & 4 & 2 \\ 1.5 & 3 & 3 & 1.5 \\ 4.5 & 1 & 5 & 2.5 \\ 1.75 & 2.5 & 2.5 & 1.75 \end{pmatrix}$$

$F_1, F_2 \ldots$ hidden features

What about missing values?

$$
\begin{array}{c}
\\
M_1 \\
M_2 \\
M_3 \\
M_4 \\
M_5
\end{array}
\begin{pmatrix}
4 & 2 & ? & 2.5 \\
4 & 0 & ? & 2 \\
1.5 & 3 & 3 & 1.5 \\
? & 1 & 5 & 2.5 \\
1.75 & 2.5 & 2.5 & 1.75
\end{pmatrix}
$$

What about missing values?

$$\begin{array}{c} \\ M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \end{array} \begin{pmatrix} 4 & 2 & ? & 2.5 \\ 4 & 0 & ? & 2 \\ 1.5 & 3 & 3 & 1.5 \\ ? & 1 & 5 & 2.5 \\ 1.75 & 2.5 & 2.5 & 1.75 \end{pmatrix}$$

- average rating
- normalize data and set missing values to zero
- ignore them $\Rightarrow$ matrix factorization

- no exact approach, $W \approx AQ^T$
- only based on non missing values
- missing values $w_{i,j}$ should be predicted by $a_i q_j^T$
- minimizes a loss function using e.g., gradient descent

Loss function:

$$\min \sum_{(i,j) \in W} \left(w_{i,j} - a_i q_j^T \right)^2$$

- W usually very sparse, i.e., prone to overfitting

$\Rightarrow$ regularization parameter

- W usually very sparse, i.e., prone to overfitting

$\Rightarrow$ regularization parameter

- some users rate stricter than other
- some movies receive better ratings than others

$\Rightarrow$ bias

- W usually very sparse, i.e., prone to overfitting

$\Rightarrow$ regularization parameter

- some users rate stricter than other
- some movies receive better ratings than others

$\Rightarrow$ bias

Loss function:

$$\min \sum_{(i,j)\in W} (w_{i,j} - \tilde{w}_{i,j})^2 + \lambda(\|a_i\|^2 + \|q_j^T\|^2 + b_i^2 + b_j^2)$$

$$\tilde{w}_{i,j} = \mu + a_i q_j^T + b_i + b_j$$

- ratings $w(u, v, p)$
- saved in matrix W with $|U| \times |P|$ columns and $|V|$ rows
- W decomposed into two matrices A and Q, such that

$$\tilde{w}(u, v, p) = \begin{cases} w(u, v, p) & w_{(u,p),v} \in V \\ a_{(u,p)} q_v^T, & \text{otherwise} \end{cases}$$

Optimization Component (OC)

- generates solution(s) based on ratings $\tilde{w}(u, v, p)$ provided by EC
- previously: black-box optimization based on PBIG or VNS
- now: MILP

Optimization Component (OC)

- $x_v \in \{0,1\}$ indicating whether a service point is built at location $v \in V$,
- $y_{u,e} \in [0,1]$ the expected degree to which a use case $e \in E_u$ is satisfied
- $z_{u,v,p} \in \{0,1\}$ indicating whether a users uses a service point at location v to satisfy the demand of a POI p

$$\max\ q \cdot \sum_{u \in U} \sum_{e \in E_u} D_{u,e} \cdot y_{u,e} - \sum_{v \in V} z_v^{\text{var}} \cdot x_v$$

$$y_{u,e} \leq \sum_{v \in \tilde{V}(u,p)} \frac{z_{u,v,p} \cdot \tilde{w}(u,v,p)}{4} \qquad \forall u \in U, \forall e \in E_u, \forall p \in e$$

$$z_{u,v,p} \leq x_v \qquad \forall u \in U, v \in V, \forall p \in P(u)$$

$$\sum_{v \in \tilde{V}(u,p)} z_{u,v,p} \leq 1 \qquad \forall u \in U, p \in P(u)$$

$$\sum_{v \in V} c_{v,1}^{\text{fix}} \cdot x_v \leq B$$

$$x_v, z_{u,v,p} \in \{0,1\} \qquad \forall v \in V, \forall u \in U, \forall p \in P(u)$$

$$y_{u,e} \in [0,1] \qquad \forall u \in U, e \in E_u$$

Feedback Component (FC)

- generates sets of locations for users to evaluate
- ask user to specify/rate best service point for each of his POIs w.r.t. $s \Rightarrow w(u, v, p)$
- rating $w(u, v, p)$: integer from zero to four

Goals:

- the number of solutions evaluated by a user should be kept small
- check approximated ratings of currently best solution
- identify potential promising service point locations

Feedback Component (FC)

- exploits similarities between users and items:
  - does not present same location to similar users
  - does not present similar locations to same user

Two evaluation rounds:

- first round: generate scenarios based on:
  - locations with promising approximated ratings $\tilde{w}(u, v, p)$
  - locations in the currently best found solution
  - locations in proximity to relevant POIs
- second round: generate scenarios based on:
  - unrated locations

Thank you for your attention!
Questions?