# Primal-Dual Algorithms for Clustering and Feature Allocation

Nathan Cordner

Boston University

1 October 2018

# Introduction

## Clustering Problem

# Introduction

## Clustering Problem

# Introduction

## Clustering Problem

# Introduction

## Clustering Problem

# Introduction

## Feature Allocation Problem

# Introduction

## Feature Allocation Problem

# Introduction

## Feature Allocation Problem

# Introduction

## Feature Allocation Problem

# (Metric) Uncapacitated Facility Location

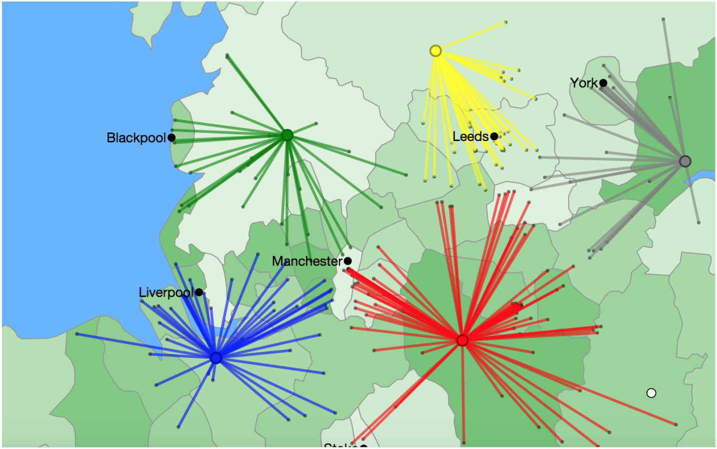


Figure: http://examples.gurobi.com/facility-location

# (Metric) Uncapacitated Facility Location

Given a set of facilities $F$, and a set of clients $C$

# (Metric) Uncapacitated Facility Location

Given a set of facilities $F$, and a set of clients $C$

- $c_{ij}$ = distance between facility $i$ and client $j$

# (Metric) Uncapacitated Facility Location

Given a set of facilities $F$, and a set of clients $C$

- $c_{ij}$ = distance between facility $i$ and client $j$
- $f_i$ = cost of opening facility $i$

# (Metric) Uncapacitated Facility Location

Given a set of facilities $F$, and a set of clients $C$

- $c_{ij}$ = distance between facility $i$ and client $j$
- $f_i$ = cost of opening facility $i$
- $x_{ij}$ = indicator for whether client $j$ connects to facility $i$

# (Metric) Uncapacitated Facility Location

Given a set of facilities $F$, and a set of clients $C$

- $c_{ij}$ = distance between facility $i$ and client $j$
- $f_i$ = cost of opening facility $i$
- $x_{ij}$ = indicator for whether client $j$ connects to facility $i$
- $y_i$ = indicator for whether facility $i$ is open

# (Metric) Uncapacitated Facility Location

Primal IP:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in F, j \in C} c_{ij} x_{ij} + \sum_{i \in F} f_i y_i \\
\text{subject to} \quad & \forall j \in C : \sum_{i \in F} x_{ij} \ge 1, \\
& \forall i \in F, j \in C : y_i - x_{ij} \ge 0, \\
& \forall i \in F, j \in C : x_{ij} \in \{0, 1\}, \\
& \forall i \in F : y_i \in \{0, 1\}.
\end{aligned}
$$

$c_{ij}$ = distance, $f_i$ = facility cost,
$x_{ij}$ = client connection, $y_i$ = facility open

# (Metric) Uncapacitated Facility Location

Primal LP Relaxation:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in F, j \in C} c_{ij} x_{ij} + \sum_{i \in F} f_i y_i \\
\text{subject to} \quad & \forall j \in C : \sum_{i \in F} x_{ij} \geq 1, \\
& \forall i \in F, j \in C : y_i - x_{ij} \geq 0, \\
& \forall i \in F, j \in C : x_{ij} \geq 0, \\
& \forall i \in F : y_i \geq 0.
\end{aligned}
$$

$c_{ij}$ = distance, $f_i$ = facility cost,
$x_{ij}$ = client connection, $y_i$ = facility open

# (Metric) Uncapacitated Facility Location

Dual LP:

maximize $\sum_{j 2 C} \quad {}_{j}$

subject to $8i\ 2\ F; j\ 2\ C: \quad {}_{j} \quad {}_{ij} \quad c_{ij};$

$8i\ 2\ F: \sum_{j 2 C} \quad {}_{ij} \quad f_i;$

$8j\ 2\ C: \quad {}_{j} \quad 0;$

$8i\ 2\ F; j\ 2\ C: \quad {}_{ij} \quad 0:$

# (Metric) Uncapacitated Facility Locatio

Primal-Dual Approach [JV]: increase dual variables unt
constraints become tight

# (Metric) Uncapacitated Facility Locatio

Primal-Dual Approach [JV]: increase dual variables unt constraints become tight

- $c_{ij}$ = cost of client $j$ to use facility $i$

# (Metric) Uncapacitated Facility Locatio

Primal-Dual Approach [JV]: increase dual variables unt
constraints become tight

- $c_{ij}$ = cost of client $j$ to use facility $i$
- $_{ij}$ = contribution of client $j$ to opening facility $i$

# (Metric) Uncapacitated Facility Locatio

Primal-Dual Approach [JV]: increase dual variables unt constraints become tight

- $c_{ij}$ = cost of client $j$ to use facility $i$
- $_{ij}$ = contribution of client $j$ to opening facility $i$

$(j) = i$ denotes that client $j$ is connected to facility $i$

# (Metric) Uncapacitated Facility Locatio

Primal-Dual Approach [JV]: increase dual variables unt
constraints become tight

- $c_{ij}$ = cost of client $j$ to use facility $i$
- $_{ij}$ = contribution of client $j$ to opening facility $i$

$(j) = i$ denotes that client $j$ is connected to facility $i$

Facility $i$ is paid for when $\sum_{j:\ (j)=i} {}_{ij} = f_i$

# (Metric) Uncapacitated Facility Locatio

Primal-Dual Approach [JV]: increase dual variables unt constraints become tight

- $c_{ij}$ = cost of client $j$ to use facility $i$
- $_{ij}$ = contribution of client $j$ to opening facility $i$

$(j) = i$ denotes that client $j$ is connected to facility $i$

Facility $i$ is paid for when $\sum_{j:\ (j)=i} \ _{ij} = f_i$

Total price paid by client $j$: $_j = \ _{(j)j} + c_{(j)j}$

# (Metric) Uncapacitated Facility Locatio

Algorithm Outline: sort list of edges in increasing order

# (Metric) Uncapacitated Facility Locatio

Algorithm Outline: sort list of edges in increasing order

If an edge $(i; j)$ goes tight:

# (Metric) Uncapacitated Facility Locatio

Algorithm Outline: sort list of edges in increasing order

If an edge $(i, j)$ goes tight:

- If $i$ is not paid for, then it gets one more contributo

# (Metric) Uncapacitated Facility Locatio

Algorithm Outline: sort list of edges in increasing order

If an edge $(i; j)$ goes tight:

- If $i$ is not paid for, then it gets one more contributo
- Else, $j$ connects to $i$ and $j$ is removed as a contributor to all other facilities

# (Metric) Uncapacitated Facility Locatio

Algorithm Outline: sort list of edges in increasing order

If an edge $(i; j)$ goes tight:

- If $i$ is not paid for, then it gets one more contributo
- Else, $j$ connects to $i$ and $j$ is removed as a contributor to all other facilities

If a facility is paid for:

# (Metric) Uncapacitated Facility Locatio

Algorithm Outline: sort list of edges in increasing order

If an edge $(i;j)$ goes tight:

- If $i$ is not paid for, then it gets one more contributo
- Else, $j$ connects to $i$ and $j$ is removed as a contributor to all other facilities

If a facility is paid for:

- Each contributing client is now declared connecte and removed as contributor to all other facilities

# (Metric) Uncapacitated Facility Locatio

Algorithm Outline: sort list of edges in increasing order

If an edge $(i, j)$ goes tight:

- If $i$ is not paid for, then it gets one more contributo
- Else, $j$ connects to $i$ and $j$ is removed as a contributor to all other facilities

If a facility is paid for:

- Each contributing client is now declared connecte and removed as contributor to all other facilities

Finish when all clients are connected

# (Metric) Uncapacitated Facility Location

Running Time: $O(m \log m)$, where $m = |F|\ |C|$

# (Metric) Uncapacitated Facility Locatio

Running Time: $O(m \log m)$, where $m = |F|\,|C|$

Approximation Bound:

$$\sum_{i \in F, j \in C} c_{ij} x_{ij} + \sum_{i \in F} f_i y_i \le 3\,\mathrm{OPT}$$

# Primal-Dual Clustering

Now let $C = F$, and each $f_i =$

# Primal-Dual Clustering

Now let $C = F$, and each $f_i =$

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j \in C} c_{ij} x_{ij} + \sum_{i \in C} y_i \\
\text{subject to} \quad & \forall j \in C : \sum_{i \in C} x_{ij} \geq 1; \\
& \forall i,j \in C : y_i - x_{ij} \geq 0; \\
& \forall i,j \in C : x_{ij} \in \{0,1\}; \\
& \forall i \in C : y_i \in \{0,1\}.
\end{aligned}
$$

# Primal-Dual Clustering

Now let $C = F$, and each $f_i =$

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j \in C} c_{ij} x_{ij} + \sum_{i \in C} y_i \\
\text{subject to} \quad & \forall j \in C : \sum_{i \in C} x_{ij} \geq 1; \\
& \forall i,j \in C : y_i - x_{ij} \geq 0; \\
& \forall i,j \in C : x_{ij} \in \{0, 1\}; \\
& \forall i \in C : y_i \in \{0, 1\}.
\end{aligned}
$$

LP relaxation and dual programs are similar

# Primal-Dual Clustering

Algorithm is just Facility Location in the special case

# Primal-Dual Clustering

Algorithm is just Facility Location in the special case

- At worst a 3-approximation algorithm

# Primal-Dual Clustering

Algorithm is just Facility Location in the special case

- At worst a 3-approximation algorithm
- As gets large, results converge to OPT

# Primal-Dual Clustering

Algorithm is just Facility Location in the special case

- At worst a 3-approximation algorithm
- As gets large, results converge to OPT

Running time is $O(n^2 \log n)$ where $n = |C|$

# Primal-Dual Clustering

Algorithm is just Facility Location in the special case

- At worst a 3-approximation algorithm
- As gets large, results converge to OPT

Running time is $O(n^2 \log n)$ where $n = |C|$

- Can run quicker for smaller values of

# Primal-Dual Clustering

Algorithm is just Facility Location in the special case

- At worst a 3-approximation algorithm
- As  gets large, results converge to OPT

Running time is $O(n^2 \log n)$ where $n = |C|$

- Can run quicker for smaller values of

Compare with K-means

# Primal-Dual Clustering

Algorithm is just Facility Location in the special case

- At worst a 3-approximation algorithm
- As gets large, results converge to OPT

Running time is $O(n^2 \log n)$ where $n = |C|$

- Can run quicker for smaller values of

Compare with K-means

- PD approach takes a little longer, but can give better results

# Primal-Dual Clustering

Examples: 1200 points

# Primal-Dual Clustering

Examples: 1200 points

K-means: 0.06 s, $k = 4$ / PD: 0.77 s, $= 7$

# Primal-Dual Clustering

Examples: 1200 points

K-means: 0.04 s, $k = 2$ / PD: 0.59 s, $= 3$

# Primal-Dual Clustering

Examples: 1200 points

K-means: 0.04 s, $k = 2$ / PD: 0.54 s, $= 2$

# Primal-Dual Clustering

Examples: 1200 points

K-means: 0.04 s, $k = 3$ / PD: 0.83 s, $= 5$

# Primal-Dual Clustering

Examples: 1200 points

K-means: 0.12 s, $k = 2$ / PD: 1.15 s, $= 10$

# Primal-Dual Feature Allocation

Similar to clustering, with relaxed constraint for $x_{ij}$

# Primal-Dual Feature Allocation

Similar to clustering, with relaxed constraint for $x_{ij}$

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j \in C} c_{ij} x_{ij} + \sum_{i \in C} y_i \\
\text{subject to} \quad & \forall j \in C : \sum_{i \in C} x_{ij} \geq 1; \\
& \forall i,j \in C : y_i - x_{ij} \geq 0; \\
& \forall i,j \in C : x_{ij} \geq 0; \\
& \forall i \in C : y_i \in \{0, 1\}.
\end{aligned}
$$

# Primal-Dual Feature Allocation

Similar to clustering, with relaxed constraint for $x_{ij}$

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j \in C} c_{ij} x_{ij} + \sum_{i \in C} y_i \\
\text{subject to} \quad & \forall j \in C : \sum_{i \in C} x_{ij} \geq 1; \\
& \forall i,j \in C : y_i - x_{ij} \geq 0; \\
& \forall i,j \in C : \color{red}{x_{ij} \geq 0}; \\
& \forall i \in C : y_i \in \{0, 1\}.
\end{aligned}
$$

LP relaxation and dual programs are the same

# Primal-Dual Feature Allocation

Algorithm Outline:

# Primal-Dual Feature Allocation

Algorithm Outline:

- same as clustering, except we do not remove any contributions from clients when they connect to facilities

# Primal-Dual Feature Allocation

Algorithm Outline:

- same as clustering, except we do not remove any contributions from clients when they connect to facilities

A client may contribute to more than one facility, and thus have multiple features

# Primal-Dual Feature Allocation

Example:

# Primal-Dual Feature Allocation

Example: = 1

# Primal-Dual Feature Allocation

Example: = 2

# Primal-Dual Feature Allocation

Example: = 7

# Primal-Dual Feature Allocation

Example: = 11

# Primal-Dual Feature Allocation

Example: = 15

# Primal-Dual Feature Allocation

Compare with BP Means [BKJ]:

# Primal-Dual Feature Allocation

Compare with BP Means [BKJ]:

# Primal-Dual Feature Allocation

Analysis:

# Primal-Dual Feature Allocation

Analysis:

- Theoretical run time is still $O(n^2 \log n)$, though can run quicker for smaller values of

# Primal-Dual Feature Allocation

Analysis:

- Theoretical run time is still $O(n^2 \log n)$, though can run quicker for smaller values of
- As gets large, the algorithm's result equals OPT

# Primal-Dual Feature Allocation

Analysis:

- Theoretical run time is still $O(n^2 \log n)$, though can run quicker for smaller values of
- As gets large, the algorithm's result equals OPT
- Worst case approximation bound: ongoing work

# Leaving Out

We require 0

# Leaving Out

We require 0

Theorem: The smallest value of that allows all edges to go tight is $= \max_i \; n \; \max_j(c_{ij}) \; \sum_j c_{ij}$

# Leaving Out

We require 0

Theorem: The smallest value of that allows all edges to go tight is $= \max_i \; n \; \max_j(c_{ij}) \; \sum_j c_{ij}$

Theorem: For $= \min_i \; n \; \max_j(c_{ij}) \; \sum_j c_{ij}$ ,

$\mathrm{OPT} = \min_i \; \sum_j c_{ij} \; +$

# Leaving Out

Running PD algorithms for multiple values of

# Leaving Out

Running PD algorithms for multiple values of

- Before the  rst facility gets paid for, the algorithm's computations are the same for each value of

# Leaving Out

Running PD algorithms for multiple values of

- Before the rst facility gets paid for, the algorithm's computations are the same for each value of
- Branch computations for a particular when it reaches its rst paid facility event

# Leaving Out

Running PD algorithms for multiple values of

- Before the  rst facility gets paid for, the algorithm's computations are the same for each value of
- Branch computations for a particular when it reaches its  rst paid facility event

Running PD algorithms without  rst choosing

# Leaving Out

Running PD algorithms for multiple values of

- Before the  rst facility gets paid for, the algorithm's computations are the same for each value of
- Branch computations for a particular when it reaches its  rst paid facility event

Running PD algorithms without  rst choosing

- Algorithm chooses values of 2 [0; ] to test

# Leaving Out

Same strategies work for running both clustering and feature allocation together

# Leaving Out

Same strategies work for running both clustering and feature allocation together

- Keep the best looking result?

# Leaving Out

Same strategies work for running both clustering and feature allocation together

- Keep the best looking result?
- Run clustering and feature allocation together, without !