Practical Time-Series Clustering for Messy Data in R

Jonathan Page
University of Hawaii Economic Research Organization
uhero.hawaii.edu

16 February, 2018

Problem Definition

GoFundMe-like company in Kenya needs to understand the typical activity patterns in campaign contributions.

Creating a list of campaign archetypes will allow them to reason about the effects of changes to their platform and campaign-specific interventions.

Outline

1. Messy data -> Matrix of trajectories
2. Dynamic Time Warping (DTW) clusters
3. k-Shape (or Shape-based) clusters

Messy Data

Data Cleaning Plan

1. Select tables and columns necessary for analysis
2. Remove test campaigns and users
3. Produce long table of features
4. Create trajectory matrix for each feature

Raw Data

Transactions

campaign_id	contributor_id	amount	payment_time
1	1	64	2016-01-01 00:00:01
1	2	128	2016-01-01 06:00:02
1	3	256	2016-01-02 12:34:56
1	3	512	2016-01-03 06:54:32
1	2	1024	2016-01-05 07:53:10
2	2	2048	2016-01-07 23:59:59

Campaigns

campaign_id	start_time
1	2016-01-01 00:00:00
2	2016-01-05 12:00:00

Intermediate Data

Combined and aggregated (in-time)

campaign_id	balance	amount	contributors	day_of_campaign
1	192	192	2	1
1	448	256	1	2
1	960	512	1	3
1	1984	1024	1	5
2	2048	2048	1	3

Matrix of trajectories

Balance trajectories

b_{it}

192	448	960	960	1984
0	0	2048	2048	2048

Amount trajectories

a_{it}

192	256	512	0	1024
0	0	2048	0	0

Data Cleaning

```r
library(tidyverse)
transactions <- read_csv("transactions.csv")
end_of_time <- max(transactions$payment_date)
campaigns <- read_csv(
    "campaigns.csv",
    na = c("", "0000-00-00 00:00:00")
  ) %>%
  mutate(
    days_old = as.numeric(difftime(
      end_of_time,
      date_created,
      units = "days"
    ))
  )
```

Data Cleaning

```r
transactions <- transactions %>%
  inner_join(
    campaigns %>%
      select(campaign_id, date_created)
  ) %>%
  mutate(
    day_of_campaign = as.numeric(difftime(
      payment_date,
      date_created,
      units = "days"
    ))
  ) %>%
  select(-date_created)
```

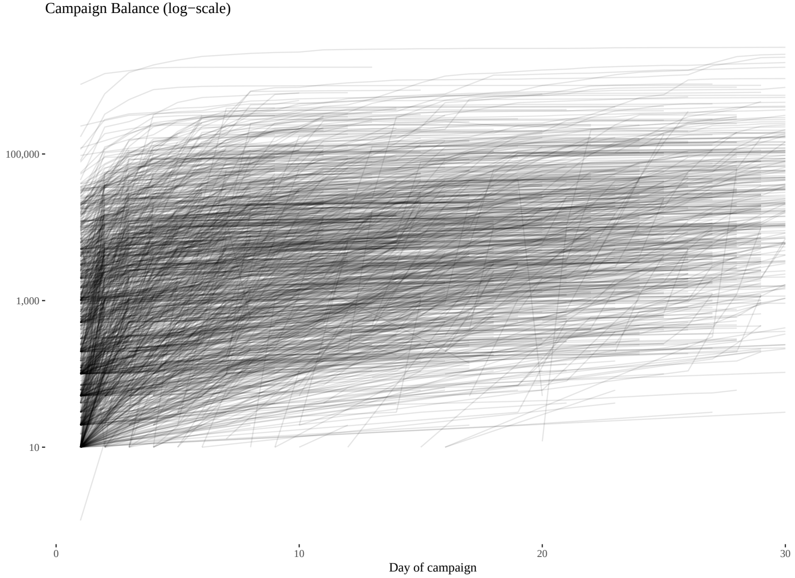
Data Cleaning

```r
clean_transactions <- campaigns %>%
  filter(days_old >= 30) %>%
  select(campaign_id) %>%
  left_join(transactions) %>%
  group_by(campaign_id) %>%
  summarize(
    transaction_count = n(),
    contributors = length(unique(contributor))
  ) %>%
  filter(contributors > 1) %>%
  select(campaign_id) %>%
  left_join(transactions) %>%
  group_by(campaign_id) %>%
  arrange(payment_date) %>%
  mutate(campaign_balance = cumsum(amount))
```

Daily Series

```r
daily_series <- clean_transactions %>%
  mutate(day_of_campaign = floor(day_of_campaign) + 1) %>%
  filter(day_of_campaign <= 30) %>%
  group_by(campaign_id, day_of_campaign) %>%
  arrange(id) %>%
  summarise(
    balance = last(campaign_balance),
    amount = sum(amount),
    transactions = n(),
  ) %>%
  filter(day_of_campaign > 0)
```

Long-form Balance Trajectories



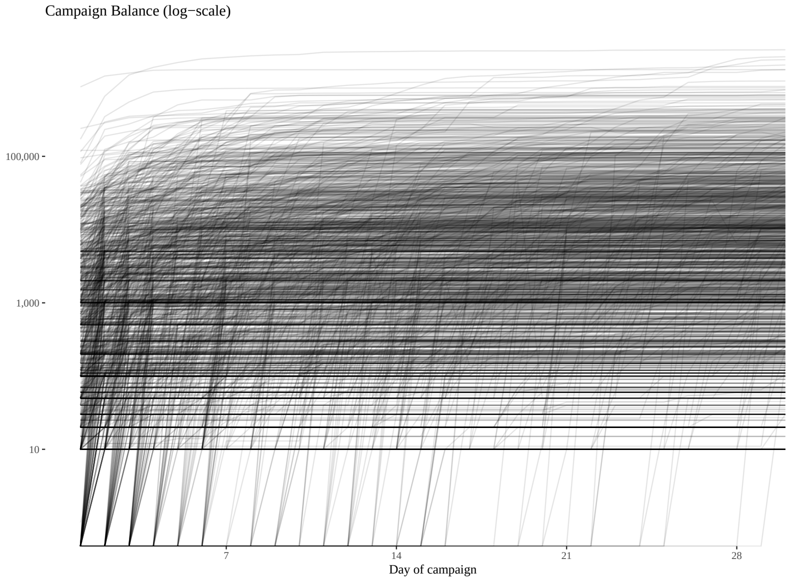
From sparse series to full matrix

```
balance_traj <- daily_series %>%
  filter(day_of_campaign %in% 1:30) %>%
  select(campaign_id, day_of_campaign, balance) %>%
  spread(day_of_campaign, balance) %>%
  mutate(`1` = coalesce(`1`, 0)) %>%
  remove_rownames() %>%
  column_to_rownames("campaign_id") %>%
  apply(1, FUN=zoo::na.locf) %>%
  t()
```

From sparse series to full matrix

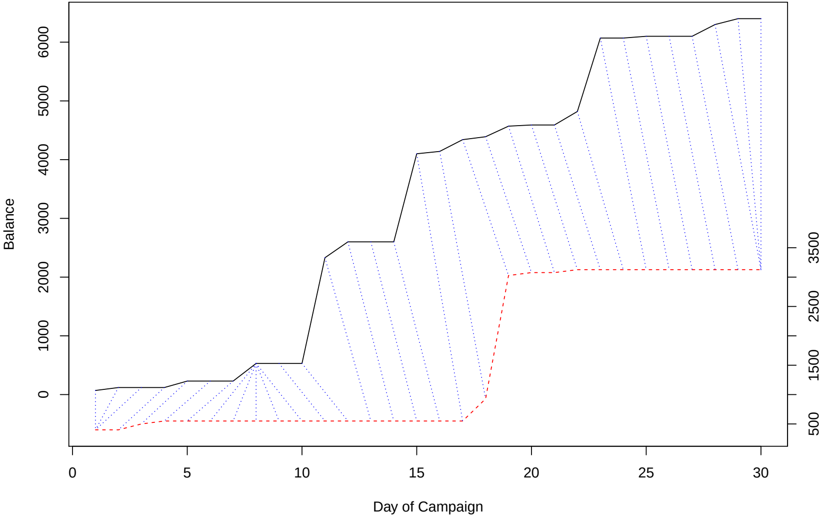
```
amount_traj <-  daily_series %>%
  filter(day_of_campaign %in% 1:30) %>%
  select(campaign_id, day_of_campaign, amount) %>%
  spread(day_of_campaign, amount, fill = 0) %>%
  remove_rownames() %>%
  column_to_rownames("campaign_id")
```

Balance Trajectories

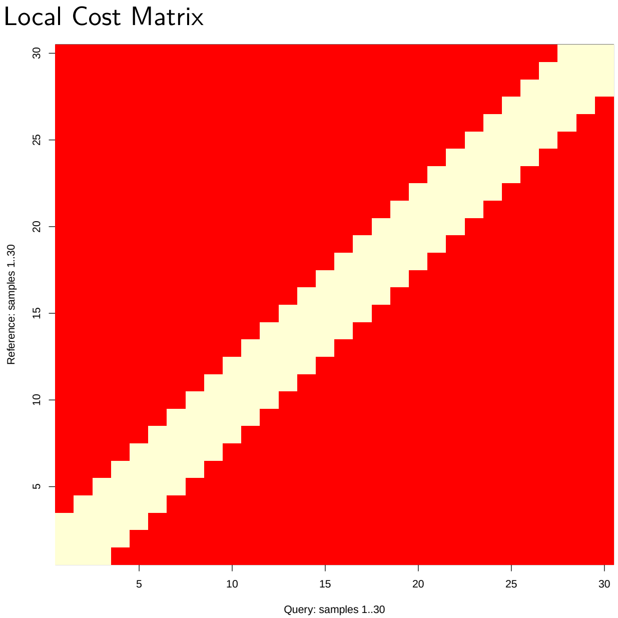


Dynamic Time Warping

Dynamic Time Warping



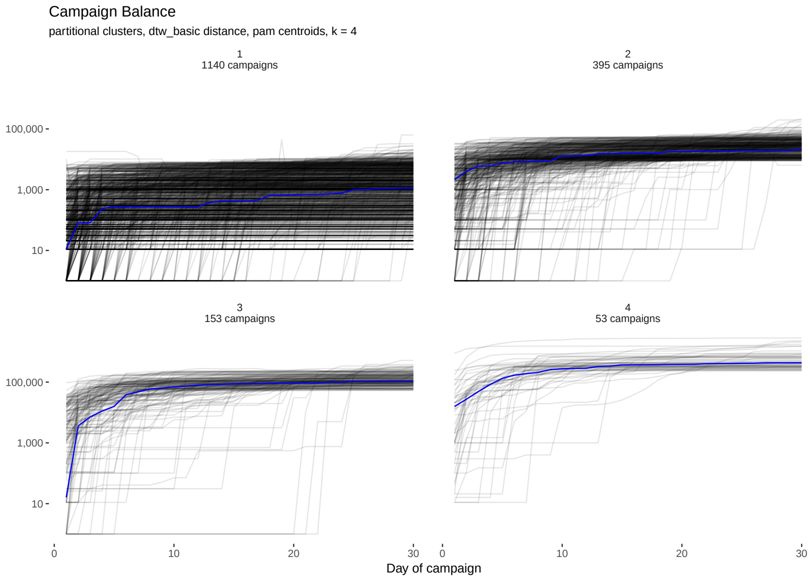
Sakoe-Chiba Window



DTW Code

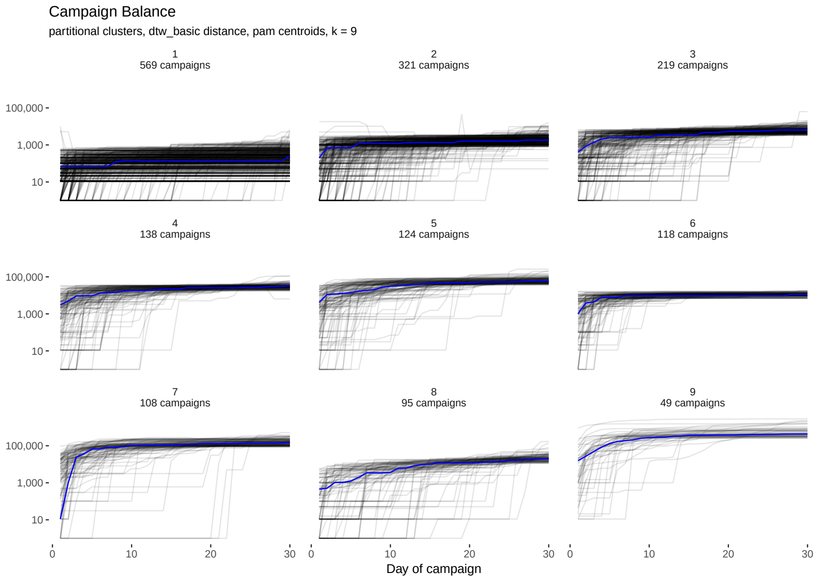
```r
pc_dtw <- tsclust(balance_traj,
                  k = 4L,
                  distance = "dtw_basic",
                  seed = 1234,
                  norm = "L2",
                  window.size = 2L)
```

Results, k=4



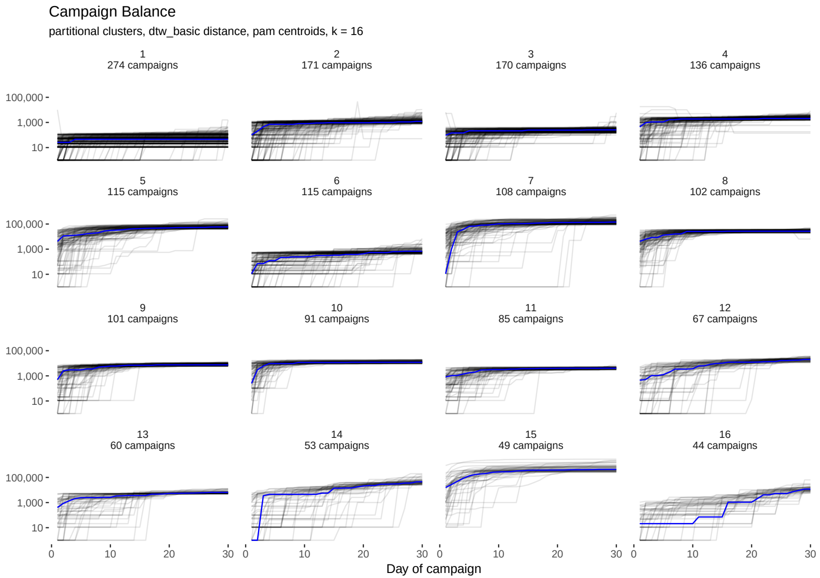
Campaign Balance
partitional clusters, dtw_basic distance, pam centroids, k = 4
1
1140 campaigns
2
395 campaigns
3
153 campaigns
4
53 campaigns
100,000
1,000
10
0
10
20
30
Day of campaign

Results, k=9



Campaign Balance
partitional clusters, dtw_basic distance, pam centroids, k = 9
1
569 campaigns
2
321 campaigns
3
219 campaigns
4
138 campaigns
5
124 campaigns
6
118 campaigns
7
108 campaigns
8
95 campaigns
9
49 campaigns
100,000
1,000
10
0
10
20
30
Day of campaign

Results, k=16


Campaign Balance
partitional clusters, dtw_basic distance, pam centroids, k = 16
1
274 campaigns
2
171 campaigns
3
170 campaigns
4
136 campaigns
5
115 campaigns
6
115 campaigns
7
108 campaigns
8
102 campaigns
9
101 campaigns
10
91 campaigns
11
85 campaigns
12
67 campaigns
13
60 campaigns
14
53 campaigns
15
49 campaigns
16
44 campaigns
100,000
1,000
10
0
10
20
30
Day of campaign

k-Shape Clustering

Shape-based distance

Cross-correlation with shift

$$SBD(\vec{x},\vec{y}) = 1 - \max_s \left(\frac{\vec{x}_{(s)} \cdot \vec{y}}{\sqrt{\|\vec{x}\|^2 \|\vec{y}\|^2}} \right)$$

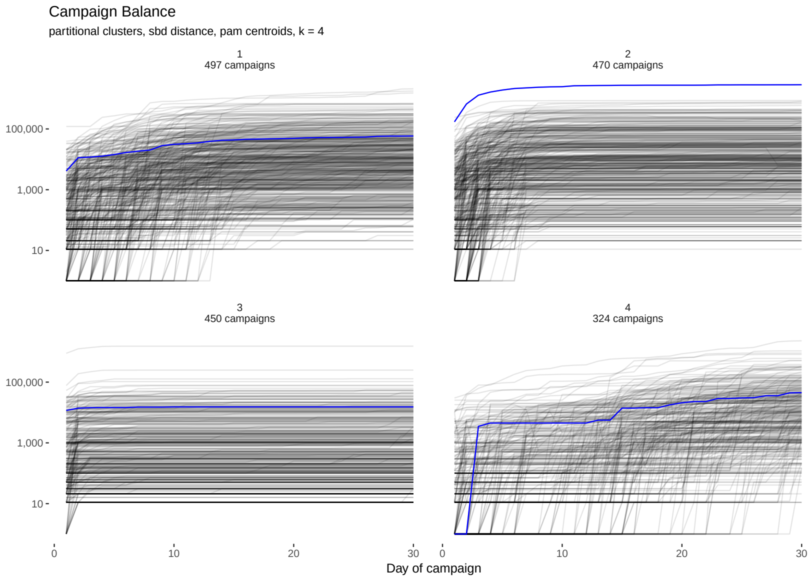
$$\vec{x}_{(s)} = \begin{cases} (\overbrace{0,\ldots,0}^{|s|}, x_1, x_2, \ldots, x_{m-s}), & s \geq 0 \\ (x_{1-s}, \ldots, x_{m-1}, x_m, \underbrace{0,\ldots,0}_{|s|}), & s < 0 \end{cases}$$

Paparrizos J, Gravano L (2015). "k-Shape: Efficient and Accurate Clustering of Time Series." In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, {*SIGMOD '15*}, *pp. 1855-1870. ACM, New York, NY, USA. ISBN 978-1-4503-2758-9. doi:10.1145/2723372.2737793.*

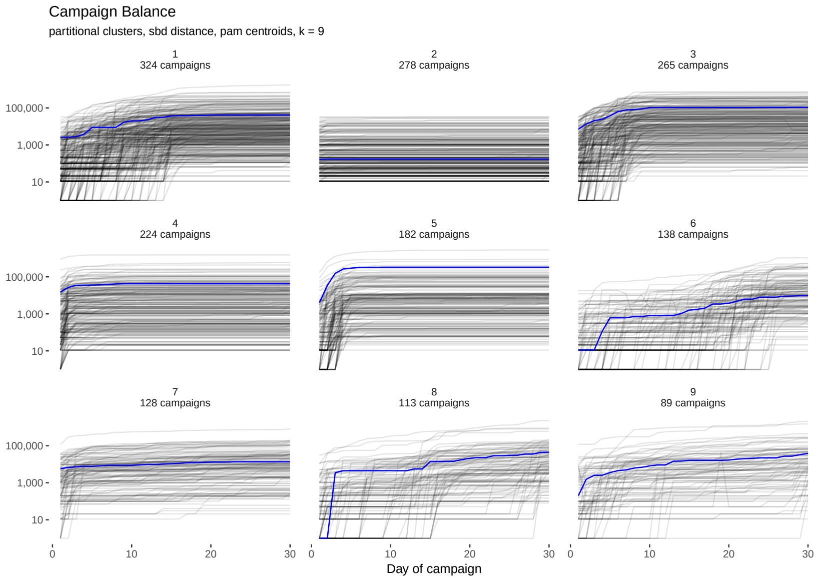
SBD Code

```r
pc_sbd4 <- tsclust(
  balance_traj,
  type = "p",
  k = 4L,
  seed = 1234,
  distance = "sbd"
)
```

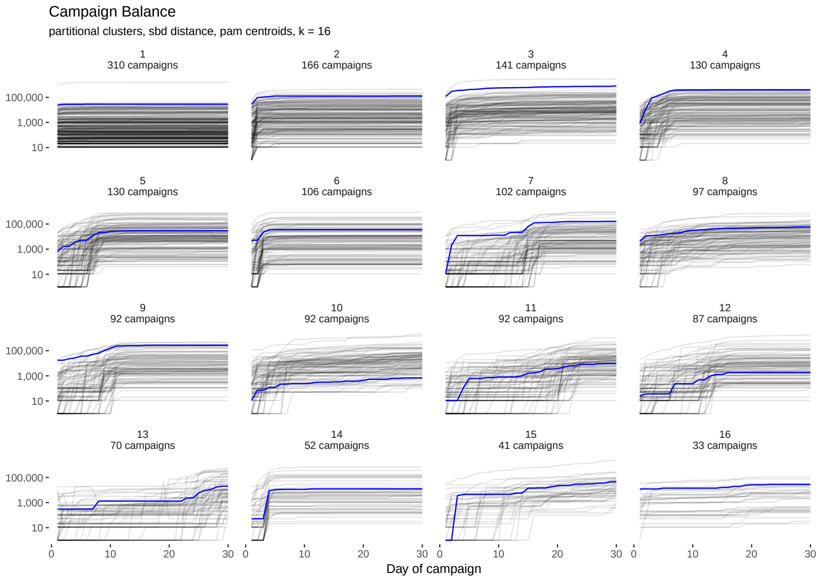
Results for k=4



Results for k=9



Results for k=16



Resources

R packages:

- `dtwclust`
- `tidyverse`

Handout Answers

1.21e+06	1.43e+07	2.56e+06	1.87e+06		1.21e−01	2.78e−02	1.89e−01	3.76e−02
2.29e+07	9.23e+06	2.74e+07	2.66e+07		1.76e−02	1.24e−01	7.08e−02	8.60e−03
4.18e+06	1.73e+07	1.96e+04	7.96e+05		3.47e−02	2.36e−01	1.06e−03	8.44e−02
3.33e+06	1.64e+07	8.08e+05	1.39e+05		1.01e−02	1.84e−01	4.07e−02	3.86e−02
4.19e+06	1.73e+07	2.64e+04	8.04e+05		1.14e−02	2.10e−01	1.61e−02	6.39e−02
4.18e+06	1.73e+07	1.84e+04	7.96e+05		3.44e−01	9.05e−02	4.06e−01	2.34e−01
4.13e+06	1.72e+07	3.10e+04	7.43e+05		3.47e−02	2.36e−01	1.06e−03	8.44e−02
4.18e+06	1.73e+07	1.89e+04	7.97e+05		4.34e−02	1.65e−01	1.02e−01	3.32e−02