Experiments Data Analysis

Yidan Chen

2022-05-06

read in data

```
algos_data <- read_csv("experiments.csv")
eps_data <- read_csv("randomized_eps.csv")
```

see if the approximation ratios are met (k=5, 3 datasets with known sp score, eps=0.1)

```
(algos_data %>%
   drop_na(exact_score) %>%
   mutate(actual_ratio=score/exact_score, max_expected_ratio=2-1/k) %>%
   group_by(Algorithm, l) %>%
   summarise(average_actual_ratio=mean(actual_ratio), max_expected_ratio=mean(max_expected_ratio)) %>%
   mutate(k=5) %>%
   relocate(k, .after = Algorithm) %>%
   ungroup() -> table_approx_ratio)
```

```
## # A tibble: 6 x 5
##   Algorithm        k     l average_actual_ratio max_expected_ratio
##   <chr>        <dbl> <dbl>                <dbl>              <dbl>
## 1 (2l-1)-stars     5     2                 1.09                1.6
## 2 (2l-1)-stars     5     3                 1.06                1.4
## 3 l-stars          5     2                 1.10                1.6
## 4 l-stars          5     3                 1.09                1.4
## 5 randomized       5     2                 1.10                1.6
## 6 randomized       5     3                 1.09                1.4
```

average running time and score for each algorithm with different k and l (seq_len=10, 5 datasets each, eps=0.1)

```
(stats_l_stars <- algos_data %>%
   filter(Algorithm=="l-stars" & k %in% c(7, 13) & seq_len==10) %>%
   group_by(k, l) %>%
   summarise(averge_time=mean(time), average_score=mean(score)) %>%
   ungroup())
```

```
## # A tibble: 6 x 4
##       k     l averge_time average_score
##   <dbl> <dbl>       <dbl>         <dbl>
```

```
## 1     7     2      0.0048          672.
## 2     7     3      0.435           659
## 3     7     4     13.0             636
## 4    13     2      0.0165         2605.
## 5    13     3      3.82           2609.
## 6    13     4    265.             2579.
```

```
(stats_2l_stars <- algos_data %>%
   filter(Algorithm=="(2l-1)-stars" & k %in% c(9, 13) & seq_len==10) %>%
   group_by(k, l) %>%
   summarise(averge_time=mean(time), average_score=mean(score)) %>%
   ungroup())
```

```
## # A tibble: 4 x 4
##       k     l averge_time average_score
##   <dbl> <dbl>       <dbl>         <dbl>
## 1     9     2        1.81         1159.
## 2     9     3      179.           1149.
## 3    13     2        5.93         2555
## 4    13     3      625.           2545.
```

```
(stats_randomized <- algos_data %>%
   filter(Algorithm=="randomized" & k %in% c(7, 13) & seq_len==10) %>%
   group_by(k, l) %>%
   summarise(averge_time=mean(time), average_score=mean(score)) %>%
   ungroup())
```

```
## # A tibble: 6 x 4
##       k     l averge_time average_score
##   <dbl> <dbl>       <dbl>         <dbl>
## 1     7     2      0.0528          672.
## 2     7     3      1.11            662.
## 3     7     4     16.7             633.
## 4    13     2      0.193          2605.
## 5    13     3      4.35           2608.
## 6    13     4     66.6            2601.
```

compare performance of different algorithms on the same datasets (k=9, len=15, 5 datasets each)

```
(table_comparison <- algos_data %>%
   filter(k==9 & seq_len==15) %>%
   group_by(Algorithm, k, l) %>%
   summarise(averge_time=mean(time), average_score=mean(score)) %>%
   arrange(l) %>%
   ungroup())
```

```
## # A tibble: 6 x 5
##   Algorithm        k     l averge_time average_score
##   <chr>        <dbl> <dbl>       <dbl>         <dbl>
## 1 (2l-1)-stars     9     2      5.74           1756.
## 2 l-stars          9     2      0.0163         1782.
## 3 randomized       9     2      0.231          1782.
## 4 (2l-1)-stars     9     3   1278.             1718
```

```
## 5 l-stars          9     3      3.33           1745.
## 6 randomized       9     3      5.63           1729
```

distribution of score for randomized algorithm when epsilon changes (k=13, len=15, l=3, 100 datasets each)

```
(table_eps <- eps_data %>%
   group_by(eps) %>%
   summarise(average_time=mean(time), average_score=mean(score)) %>%
   mutate(k=13, l=3) %>%
   relocate(k, l) %>%
   ungroup())
```

```
## # A tibble: 5 x 5
##       k     l   eps average_time average_score
##   <dbl> <dbl> <dbl>        <dbl>         <dbl>
## 1    13     3   0.1         14.1          3844.
## 2    13     3   0.3         10.1          3839.
## 3    13     3   0.5          9.09         3840.
## 4    13     3   0.7          8.09         3840.
## 5    13     3   0.9          7.11         3843.
```

```
eps_data %>%
  mutate(epsilon=as.factor(eps)) %>%
  ggplot(aes(x=score, color=epsilon)) + geom_density()
```