Frontpage Generation Segmentation

A Class Project for CMPT-740 Foundations of Data Mining

Yuanzhu Peter Chen

Lei Duan

School of Computing Science, Simon Fraser University

Talk Outline

- Motivation – *Yihoo!* CEO's New Idea
- Preliminaries – Segmentation Problem
- Formulation
- Segmentation Algorithms
- Experiments
- Conclusion

Episode I

Graphite and Paper

Motivation

- *Yihoo!*'s frontpage
  - Mundane listing of all categories
  - Order and number of headlines
- CEO's new idea
  - Individual layout
- CSO's trade-off
  - Segmentation

***Yihoo!* Frontpage**

- **Domestic Events**
  - H8 Summit Held in Dexas Ranch
  - Prepublican Leading 2004 Campaign
  - PairForce One Arrived at Dave Camp
- **Global Events**
  - Logic Bomb Blast in Cyberton Capital Claiming 13 Mainframes
  - Anti-Opposite-Sex Marriage Parade in Shanhai
- **Finance**
  - Air Candy Refused of Bankcrupcy Protection
  - 230,000 Nazda 6-Series Recalled
- **Arts & Entertainment**
  - Hypercube Reloaded – Movie Preview
  - Join Michelle Jackson in Everland Party
- **Tech**
  - Tracking Web Users by Biscuits
  - Macrohard Releases New OS Security Blackholes
- **Horny**
  - 101% Male Want Steel Implantation Out – Survey

Segmentation Problem

— A $-driven clustering

- Capitalists want to

$$\max_{x \in \mathcal{D}} f(x)$$

or when different clients are considered

$$\max_{x \in \mathcal{D}} \sum_{i \in \mathcal{C}} g(x, y_i).$$

- Greedy capitalists want to maximize

$$\sum_{i \in \mathcal{C}} \max_{x \in \mathcal{D}} g(x, y_i).$$

Segmentation Problem (cnt'd)

— A $-driven clustering (cnt'd)

- Merciful/realistic capitalists want to partition $\mathcal{C}$ into k parts $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$, so as to maximize the sum of the optima

$$\sum_{j=1}^{k} \max_{x \in \mathcal{D}} \sum_{\mathcal{C}_j} c_i \cdot x.$$

- Segmentation versions of many simple combinatorial problems are NP-hard.

Problem Formulation

- Quantize a user's habit and frontpage layout
- Define discrepancy and satisfaction functions

Quantizing a User Habit

- Pamela's web accessing pattern:

	Order	Frequency
Domestic	3	5%
Global	5	5%
Arts & Entert.	1	50%
Technology	2	15%
Business	4	25%
Horny	0	0%
Vectors	$u_o = \langle 3, 5, 1, 2, 4, 0 \rangle$	$u_f = \langle 4, 4, 1, 3, 2, 0 \rangle$

Quantizing a Frontpage

***Yihoo!* Frontpage**

- **Domestic Events**
  - H8 Summit Held in Dexas Ranch
  - Prepublican Leading 2004 Campaign
  - PairForce One Arrived at Dave Camp
- **Global Events**
  - Logic Bomb Blast in Cyberton Capital Claiming 13 Mainframes
  - Anti-Opposite-Sex Marriage Parade in Shanhai
- **Finance**
  - Air Candy Refused of Bankcrupcy Protection
  - 230,000 Nazda 6-Series Recalled
- **Arts & Entertainment**
  - Hypercube Reloaded – Movie Preview
  - Join Michelle Jackson in Everland Party
- **Tech**
  - Tracking Web Users by Biscuits
  - Macrohard Releases New OS Security Blackholes
- **Horny**
  - 101% Male Want Steel Implantation Out – Survey

- Category order vector w_o
- Headline number vector w_f

Discrepancy Function

- Weighted rank difference between user u and web page w, where $u = \langle u_1, u_2, \ldots, u_l \rangle$ and $w = \langle w_1, w_2, \ldots, w_l \rangle$.
- *i-discrepancy* between u and w:

$$d_i(u_i, w_i) = \begin{cases} 0 & \text{if } u_i = 0 \text{ (Note that } w_i > 0.\text{)} \\ \frac{1}{(u_i + w_i)^p} |u_i - w_i| & \text{else,} \end{cases}$$

- *Discrepancy function* of u and w to be:

$$d(u, w) = \sum_{i=1}^{l} d_i(u_i, w_i).$$

Satisfaction Function

- *Satisfaction function* s of a user u browsing a page w:

$$s : \mathbb{R}^l \times \mathbb{R}^l \longrightarrow \mathbb{R}, \text{ where } s(u, w) = e^{-d(u,w)}.$$

- $s(u, w) \in (0, 1]$.
- $s(u, w) = 1$ iff the two vectors are identical on all categories that the user cares.

Episode II

Silicon and Electron

Segmentation Heuristics

- Algo I – k-Means Based Algorithm
- Algo II – Modified k-Means Based Algorithm
- Algo III – Optimality Branch-and-Bound Algorithm

k-Means Based Algorithm

proc *k_means*(U) $\equiv$ //U is set of user vectors

begin

$centroids :=$ *initial_centroids*(U);

$new\_centroids := NULL$;

while $!(centroids = new\_centroids)$ **do**

$new\_centroids :=$ *iterate_k_means*$(U, centroids)$;

//Discrepancy function replacing distance function.

od //Until centroids don't improve.

$permutations :=$ *cent2perm*$(centroids)$;

//Convert centroids to permutations

end

Modified k-Means Based Algorithm

proc *modified_k_means*(U) $\equiv$ //U is set of user vectors

begin

$centroids :=$ *initial_centroids*(U);

$permutations :=$ *cent2perm*$(centroids)$;

//Convert centroids to permutations

$permutations := NULL$;

while $!(permutations = new\_permutations)$ **do**

$new\_centroids :=$ *iterate_mod_k_means*$(U, permutations)$;

//Discrepancy function replacing distance function.

$new\_permutations :=$ *cent2perm*$(centroids)$;

od //Until permutations don't improve.

end

Optimality Branch-and-Bound

proc *optimality_brandnbound*(U) $\equiv$ //U is set of user vectors

begin

$all\_trunc\_perms :=$ *generate_trunc_perms*(l, i);

//All i-permutations of l.

foreach k-subset $ktperms \subseteq all\_trunc\_perms$ **do**

$opt\_ktperms :=$ *record_optimal_k_perms*$(ktperms, U)$;

//Record best k truncated permutations so far.

od

foreach $p \in opt\_ktperms$ **do**

extend_permutation(p) //Extend p full according to its cluster

od

end

Experiments

— Data Source and Preprocessing

- MSNBC's page visit record

```
1 1
6
6 7 7 7 6 6 8 8 8 8
6 9 4 4 4 10 3 10 5 10 4 4 4
1 1 1 11 1 1 1
```

- Transform each user sequence into two vectors u_o and u_f.

Experimental Settings

- Number of users

$$n = 1000, 2000, 4000, 8000, 16000, 32000, 64000, \text{and} 128000$$

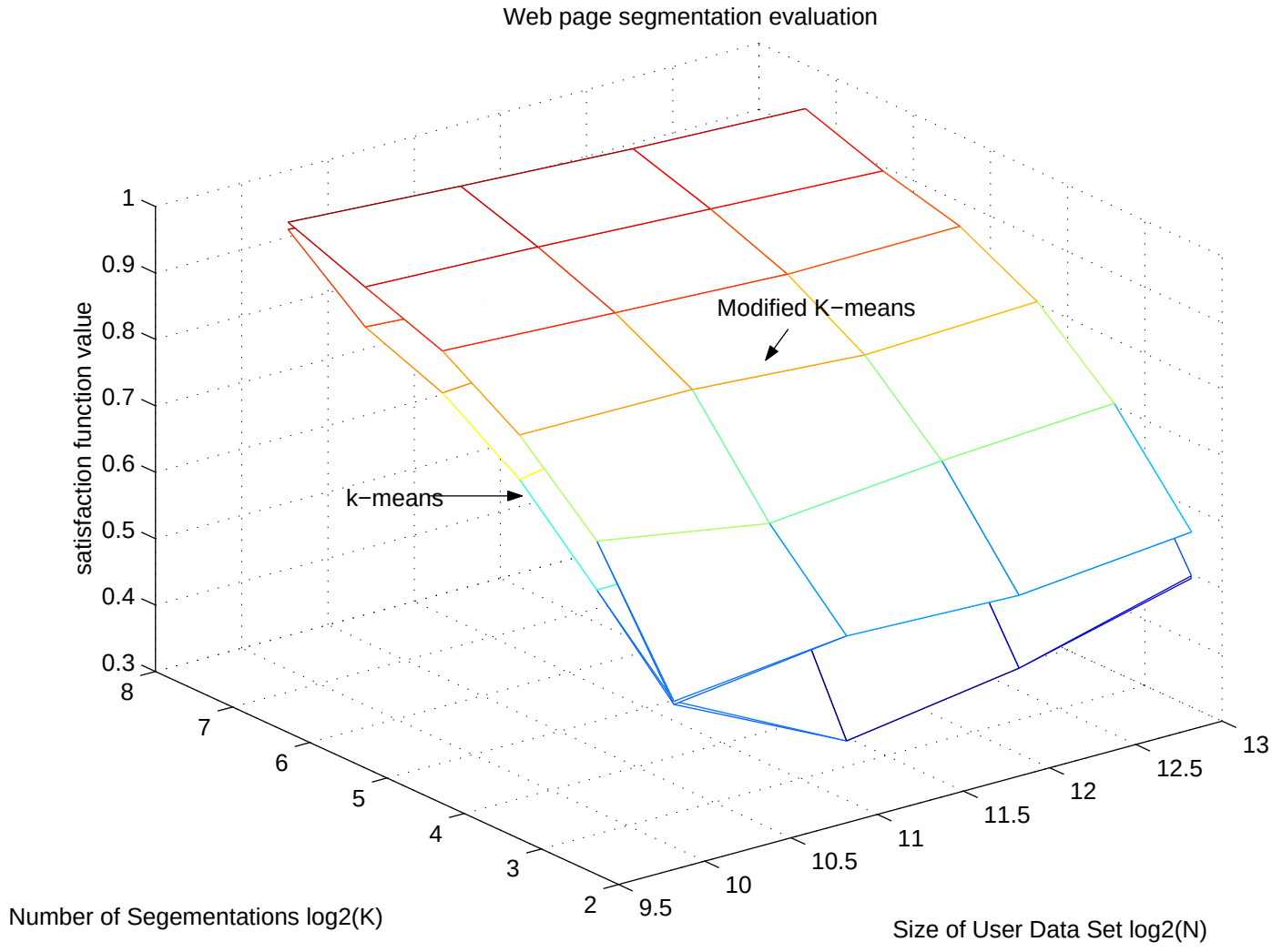
- Number of frontpages to generate

$$k = 5, 10, 20, 40, 80, 160$$

- All combinations of the above n and k's

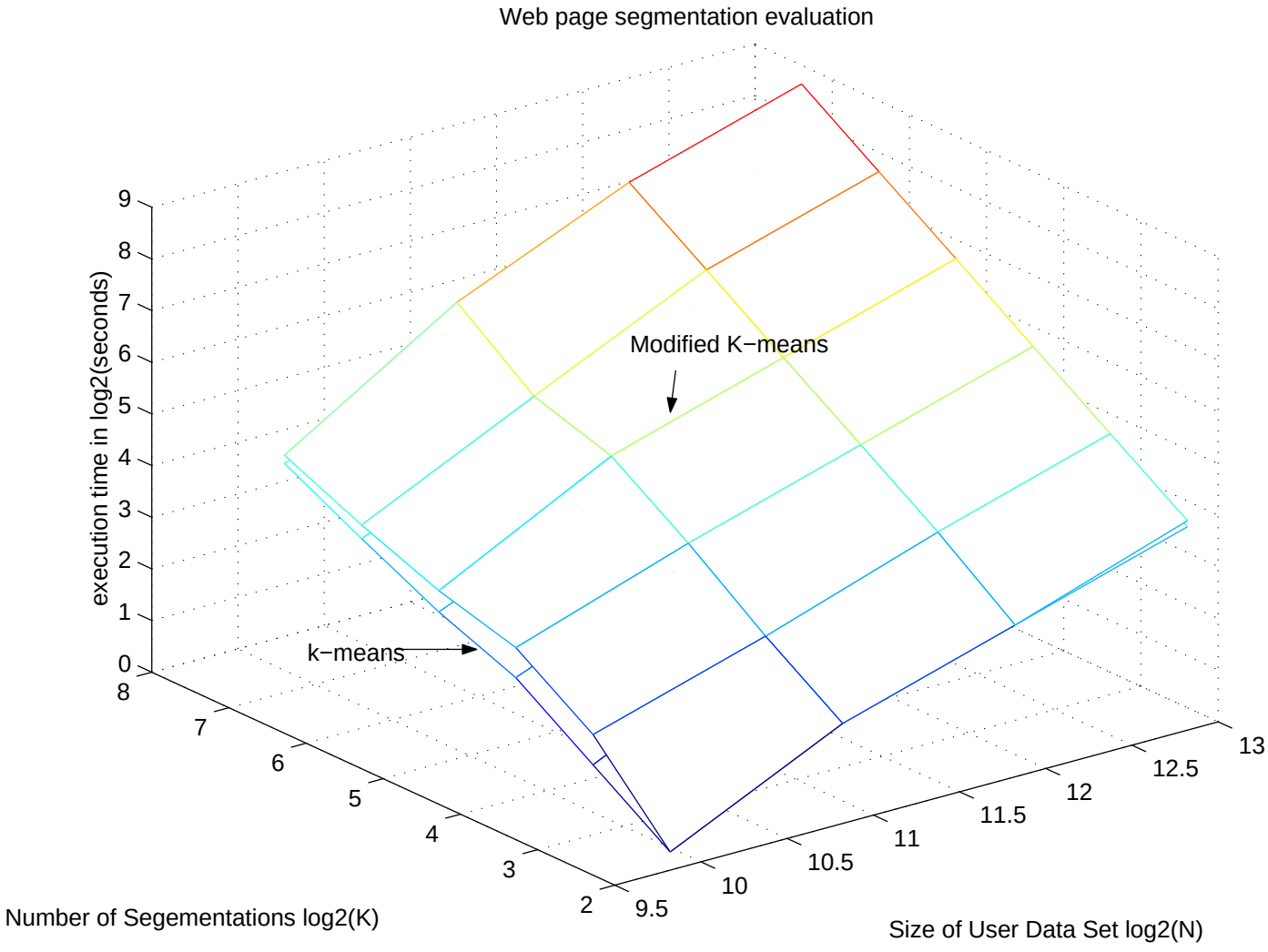
Experiment Results

- Satisfaction



Experiment Results (cnt'd)

- Execution time



Conclusion

- The frontpage generation segmentation problem can be formulated using integer permutations and defining a discrepancy function and satisfaction function between permutations.
- Two k-means based algorithms are proposed, among others, and are shown being able to generate segmentations with high satisfaction at controllable costs.

Open Problems

- What is the computational complexity of this problem? (Very likely NP-hard.)
- Does the problem itself or some of its special cases allow good approximation algorithms with provable bounds?
- Would other formulation and discrepancy/satisfaction definitions have the nicer computational complexity properties?

References

References

[1] Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. A microeconomics view of data mining. *Journal of Data Mining and Knowledge Discovery*, 2(4):311–324, December 1998.

[2] Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. Segmentation problems. In *Proceedings of 30th ACM Symposium on Theory of Computing (STOC)*, pages 473–482, 1998.