Modeling Item Sequences by Overlapped Markov Embeddings

Cheng-Hsuan Tsai (蔡誠軒)

Advisor: Pu-Jen Cheng, Ph.D. (鄭卜壬 博士)

Computer Science & Information Engineering
National Taiwan University

Outline

Section 1

Problem definition

What to predict?

Given

- A set of items $S = \{s_1, s_2, \ldots, s_{|S|}\}$.
- A set of item sequences (ex. $p = s_i \rightarrow s_j \rightarrow s_k$.)

We want to predict the *transition probability*:

$$P(s_b|s_a) \quad \forall s_a, s_b \in S$$

which is the probability of s_b following s_a.

Applications

Recommend the next item based on current item.

- Music playlists.
- Online shopping.
- News reading list.

Section 2

Limitations of current methods

Possible solutions

- We want to find solutions depending purely on item sequences.
- Two directions from our survey...

Direction 1: Markov Chain

- N-gram model[1,2], Random-walk with Restart[3,4].
- Pro: Efficiency.
- Con: Highly depends on observed data.

---

[1]Vlado Keselj, Computational Linguistics 2009

[2]B. McFee et al. ISMIR'11

[3]Robert Ragno et al., MIR'05

[4]Hanghang Tong, ICDM'06

Direction 2: Latent Vector Space

- Matrix Factorization[5,6], Logistic Markov Embedding[7].
- Pro: Able to derive similarity between any two vectors.
- Con: Computation cost is high.

---

[5] Yehuda Koren et al., Computer'09

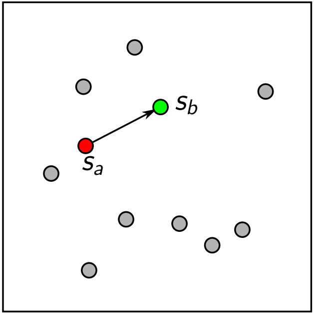
[6] Natalie Aizenberg et al., WWW'12

[7] Shuo Chen et al., KDD'12

Logistic Markov Embedding[8] (LME)

$s_i \in S \Rightarrow \vec{s_i}$ in Euclidean space

$$P(s_b|s_a) = P_{\text{LME}}(s_b|s_a) = \frac{e^{-\|\vec{s_b}-\vec{s_a}\|^2}}{\sum_{s_i \neq s_a} e^{-\|\vec{s_i}-\vec{s_a}\|^2}}$$



2D Euclidean space

[8]Shuo Chen et al. "Playlist prediction via metric embedding". In: *KDD*. 2012, pp. 714–722.

Logistic Markov Embedding

Optimize the vectors by maximizing log-likelihood

$$\max \sum_{s_a \xrightarrow{w} s_b \ \in \ \text{training}} w \cdot \ln P(s_b | s_a)$$

- w is the number of occurrence of $s_a \rightarrow s_b$.
- Use gradient decent for optimization.

Why LME?

- Able to derive transition probability between any two items.
- $P(s_b|s_a)$ and $P(s_a|s_b)$ can have different values.

Problem of LME

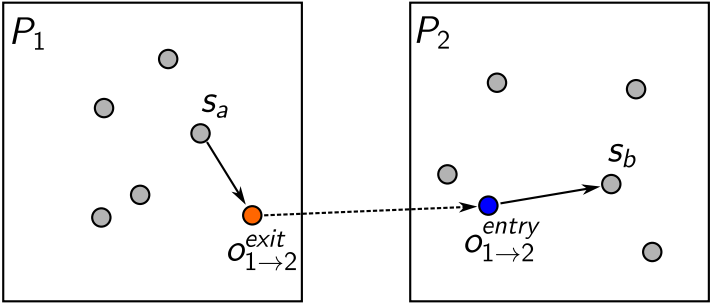
Training time complexity is too high: $\Theta(|S|^2)$

A speed-up approach: Multi-LME[9]

- Divide S into k balance-sized partitions.
- Add $2(k-1)$ *portals* to each partition.
- Learn a LME for each partition.

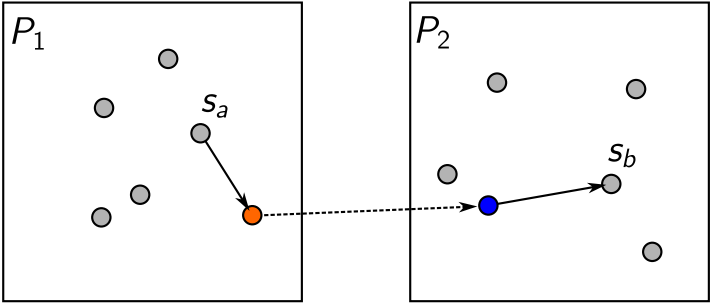
[9] Shuo Chen et al. "Multi-space probabilistic sequence modeling". In: *KDD*. 2013, pp. 865–873.

Portal



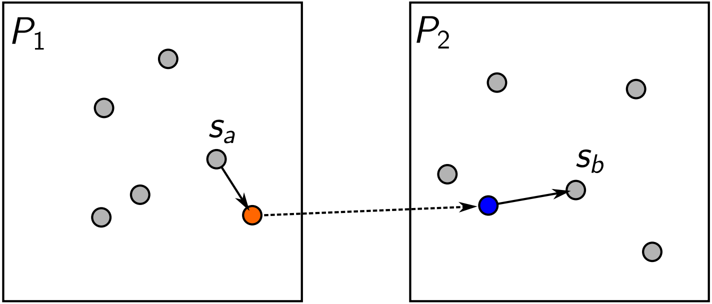
$$P(s_b|s_a) = P_{\mathrm{LME}(C_1)}(o_{1\to 2}^{exit}|s_a) \cdot P_{\mathrm{LME}(C_2)}(s_b|o_{1\to 2}^{entry})$$

Problem of Multi-LME



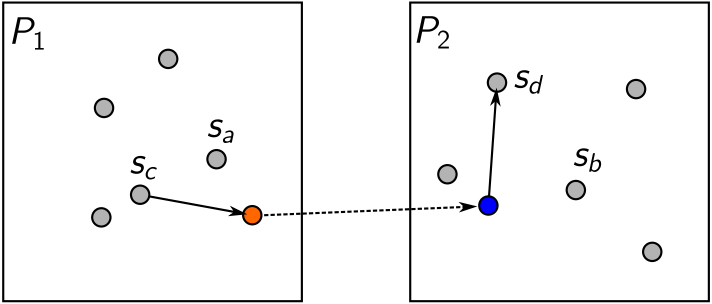
Observed transitions: $s_a \to s_b$

Problem of Multi-LME



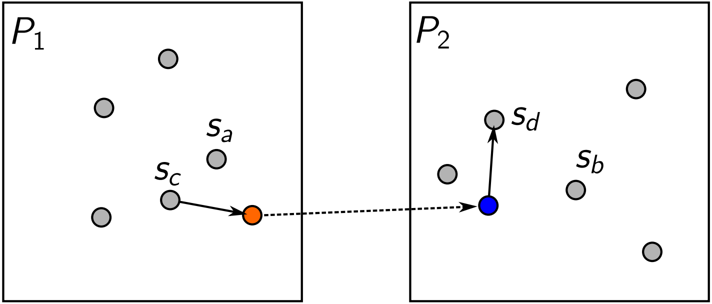
Observed transitions: $s_a \rightarrow s_b$

Problem of Multi-LME



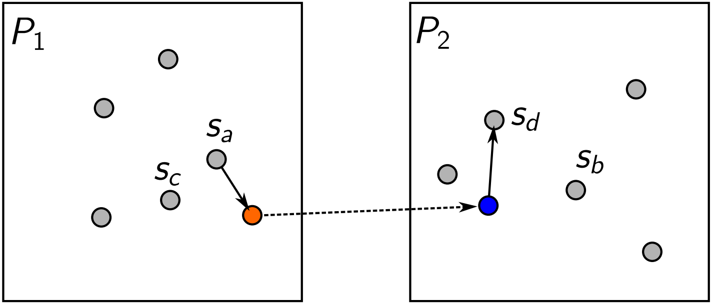
Observed transitions: $s_a \rightarrow s_b$, $s_c \rightarrow s_d$

Problem of Multi-LME



Observed transitions: $s_a \rightarrow s_b$, $s_c \rightarrow s_d$

Problem of Multi-LME



$P(s_d|s_a)$ is also increased, but it's not reasonable from the independent observations $s_a \rightarrow s_b$ and $s_c \rightarrow s_d$.

Problem of Multi-LME

- The prediction becomes inaccurate if the number of "crossing transitions" is large.
- Multi-LME uses state-of-the-art vertex-partitioning algorithm (ex. METIS) on the *transition graph* to minimize the number of crossing transitions.

Time complexity of Multi-LME

$$\overbrace{O(m + k\log k)}^{\text{partitioning}} + \overbrace{O(k(\frac{|S|}{k} + k)^2)}^{\text{training LMEs}}$$

- m is the number of edges in transition graph.
- k requires a careful adjustment.

Section 3

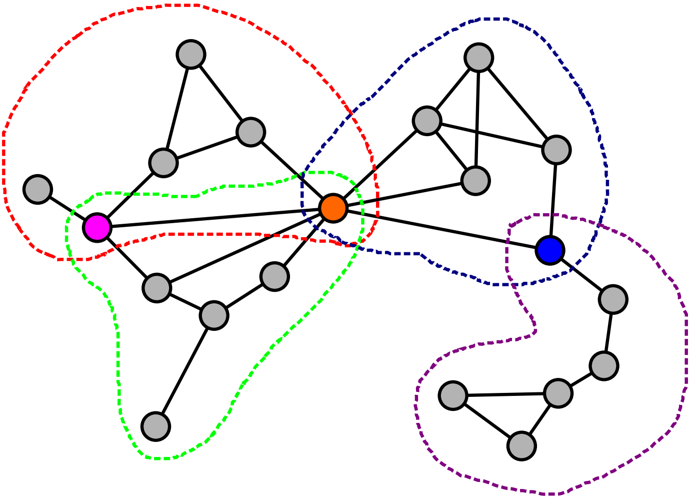
Our new solution: Overlapped-LME

Overlapped-LME

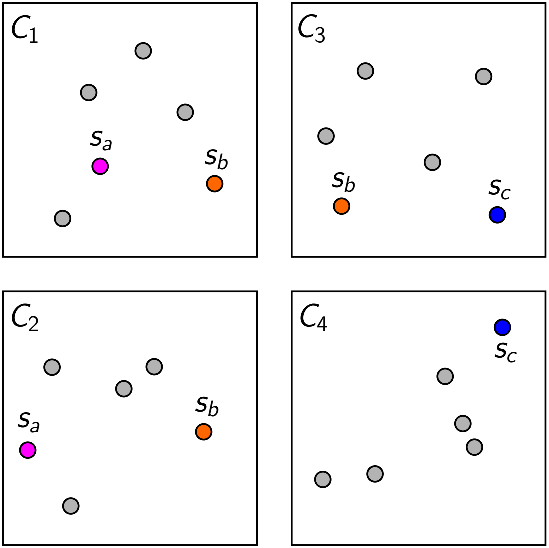
Transition graph $G \quad \Rightarrow \quad$ Clusters $\{C_1, \ldots, C_\ell\}$

- One vertex can reside in multiple clusters.
- Each edge will reside in at least one cluster.
- $|C_i| \leq N, \forall i$ ($|C_i| =$ number of vertices in C_i)
- $\ell \leq m$ ($m =$ number of edges in G)

Transition Graph ⇒ Clusters



Learn a LME for each cluster



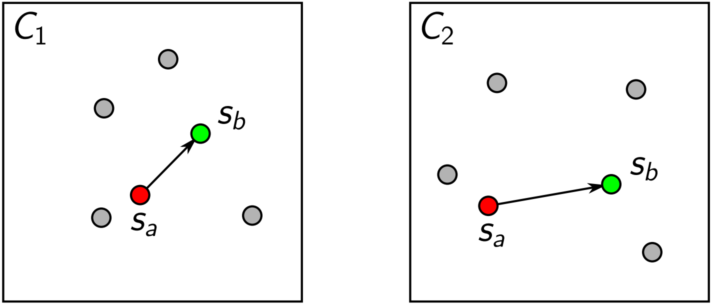
Design of $P(s_b|s_a)$

Given the clusters and corresponding LME's, how do we model the value of

$$P(s_b|s_a) = \ ?$$

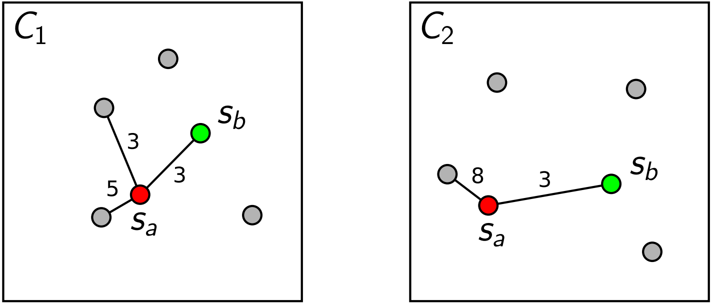
for any two items $s_a, s_b \in S$.

One-step probability



$$\begin{aligned} P_{one}(s_b|s_a) &= p(s_a, C_1) \times P_{\mathrm{LME}(C_1)}(s_b|s_a) \\ &+ p(s_a, C_2) \times P_{\mathrm{LME}(C_2)}(s_b|s_a) \end{aligned}$$

One-step probability



$$p(s_a, C_1) = \frac{5+3+3}{(5+3+3)+(8+3)+\cdots}$$

$P(s_b|s_a)$ version 1

$$P(s_b|s_a) = P_{one}(s_b|s_a)$$

Problem: If none of the clusters contain both s_a and s_b, then $P(s_b|s_a) = 0$.

Background probability

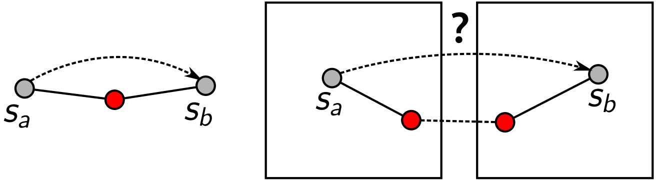
To ease the problem of version 1, add a small background probability for each transition.

$$P_{bg} = \frac{1}{|S| - 1}$$

$P(s_b|s_a)$ version 2

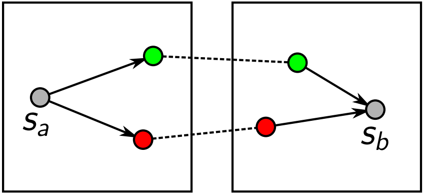
$$P(s_b|s_a) = \alpha \cdot P_{one}(s_b|s_a) + (1-\alpha) \cdot P_{bg}$$

Two-steps probability



- Cannot be modeled by $P_{one}(s_b|s_a)$.
- Should be stronger than P_{bg}.

Two-steps probability



$$P_{two}(s_b|s_a) = \sum_{s_i \neq s_a, s_b} P_{one}(s_i|s_a) \cdot P_{one}(s_b|s_i)$$

- Can be extended to three or more steps.

$P(s_b|s_a)$ version 3 (final version)

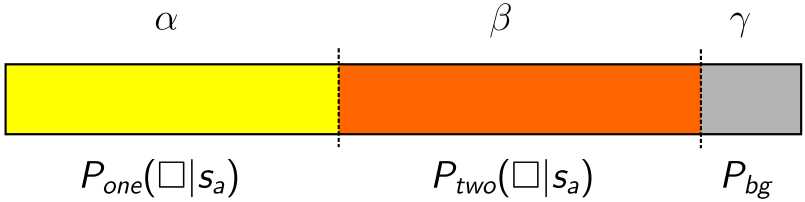
Given $\alpha + \beta + \gamma = 1$,

$$\begin{aligned} P(s_b|s_a) = & \, (\alpha + \beta \times R_{one}(s_a)) \times P_{one}(s_b|s_a) \\ & + [\![s_b \in S \setminus X(s_a) \setminus \{s_a\}]\!] \times \\ & \qquad \beta \times (1 - R_{one}(s_a) - R_{self}(s_a)) \times P_{two}(s_b|s_a) \\ & + (\beta \times R_{self}(s_a) + \gamma) \times P_{bg} \end{aligned}$$

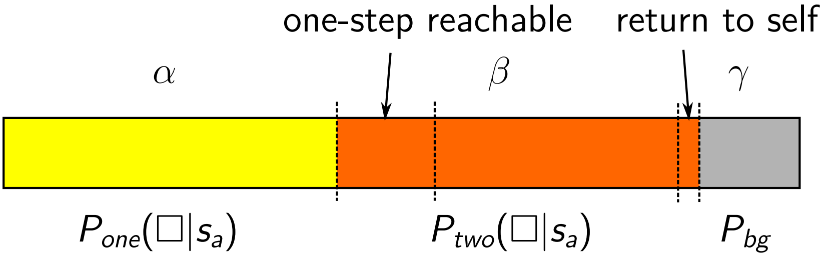
$P(s_b|s_a)$ version 3 (final version)

$$
\begin{aligned}
X(s_a) &= \{s_i | P_{one}(s_i|s_a) > 0\} \\
R_{one}(s_a) &= \sum_{s_i \in X(s_a)} P_{two}(s_i|s_a) \\
R_{self}(s_a) &= P_{two}(s_a|s_a)
\end{aligned}
$$

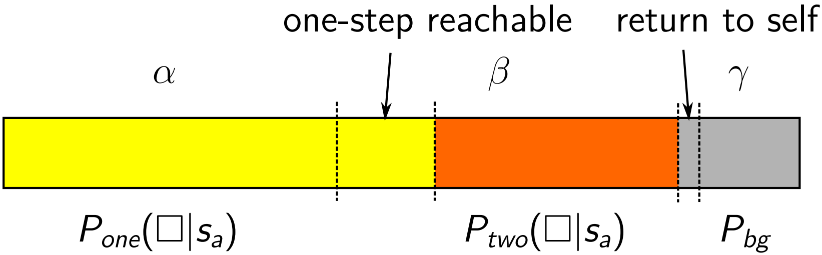
$P(s_b|s_a)$ version 3 (final version)



$P(s_b|s_a)$ version 3 (final version)



$P(s_b|s_a)$ version 3 (final version)



Clustering algorithms

How to convert transition graph into clusters?

- Baseline method: Random clustering
- For edge density: Density clustering
- For weight sum: Weight clustering

Baseline method: Random clustering

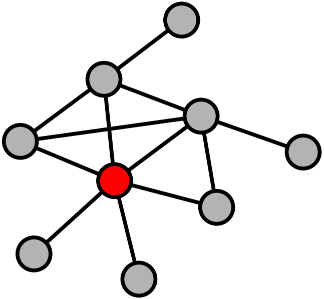
while *there's still edges remain in* G **do**
  keep picking edges from G until the number of unique vertices incident to these edges reach limit N;
  create a cluster by vertices incident to these edges;
  remove these edges from G;
end

Algorithm 1: Density clustering

Maximize the number of edges in each cluster.

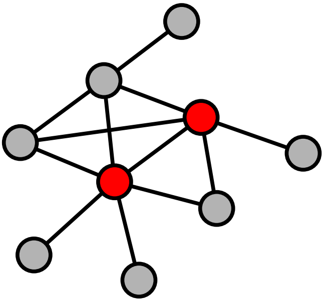
- Expect more accurate LME.
- Less clusters and shorter training time.

Density clustering



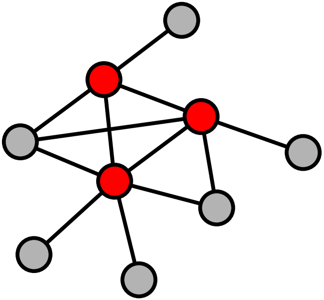
Start from the vertex with largest degree.

Density clustering

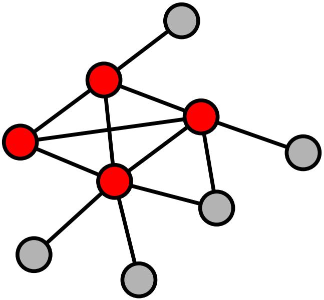


From all neighbors of red vertices, pick the one that will contribute most edges.

Density clustering

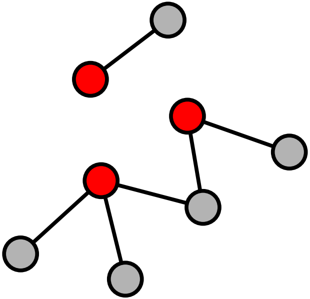


Density clustering



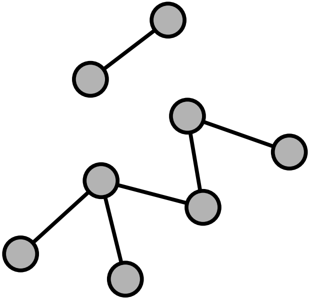
When limit is reached, create a cluster from red vertices.

Density clustering



Remove the edges and isolated vertices.

Density clustering



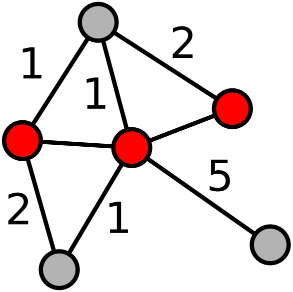
Find the next cluster from remaining graph.

Algorithm 2: Weight clustering

Maximize the sum of edges' weights in each cluster.

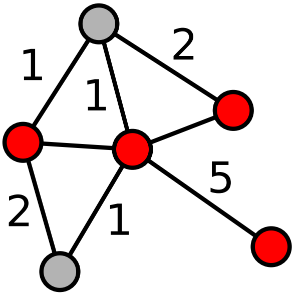
- Assign strongly related items (ex. connected by a heavy path on G) to the same cluster.
- Expect more accurate and higher transition probability between related items.

Weight clustering



From the neighbors of red vertices, pick the one that will contribute most weight.

Weight clustering



From the neighbors of red vertices, pick the one that will contribute most weight.

Time complexity of Overlapped-LME

$$\overbrace{O(m)}^{\text{clustering}} + \overbrace{O(m \times N^2)}^{\text{training LMEs}}$$

- N: cluster's size limit, m: number of edges.
- Take N as constant, no parameter to tune.

Section 4

Experiments

Data sets

	Data type	Songs	Training transitions	Testing transitions
Yes.com	radio playlists	9,775	172,510	1,602,079
KKBOX	user logs	233,501	5,543,451	5,878,953

- Yes.com is the same dataset tested by Multi-LME (KDD'13).

Evaluation

Use average log-likelihood on testing data as evaluation:

$$\frac{1}{T} \sum_{s_a \xrightarrow{w} s_b \ \in \ \text{testing}} w \cdot \ln P(s_b | s_a)$$

where

$$T = \sum_{s_a \xrightarrow{w} s_b \ \in \ \text{testing}} w$$

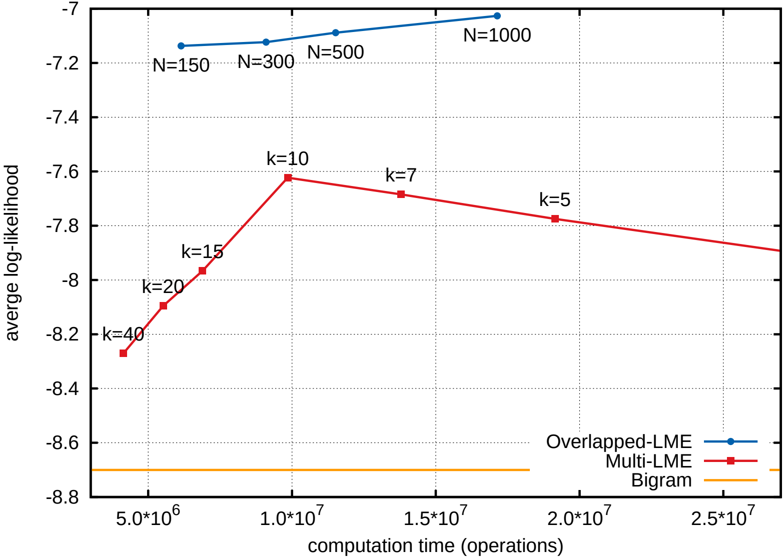
Detail settings

- Fix the parameters of LME.
- Measure the computation time by the total number of operations.
- Assume training a LME with n items takes n^2 operations.

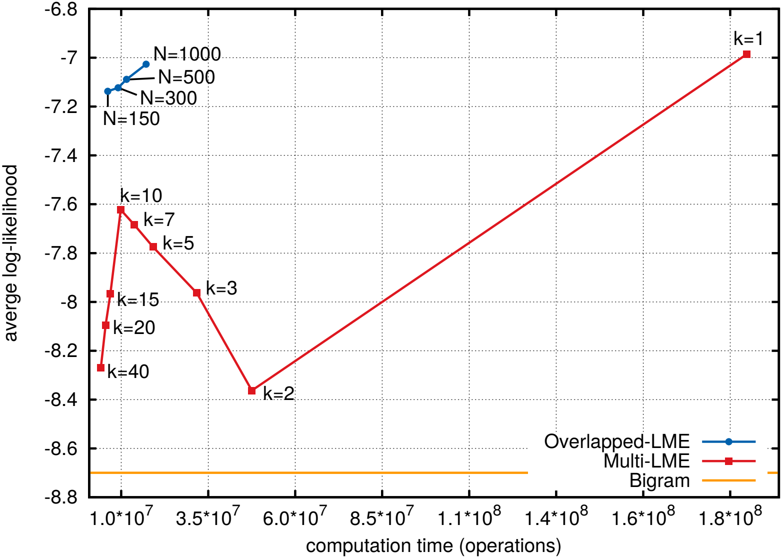
Experiments

1. Compare the performance between Overlapped-LME and Multi-LME.
2. Check the effect of two-steps probability.
3. Check the effect of different clustering algorithms.

Overlapped-LME v.s. Multi-LME (Yes.com)



Overlapped-LME v.s. Multi-LME (Yes.com)

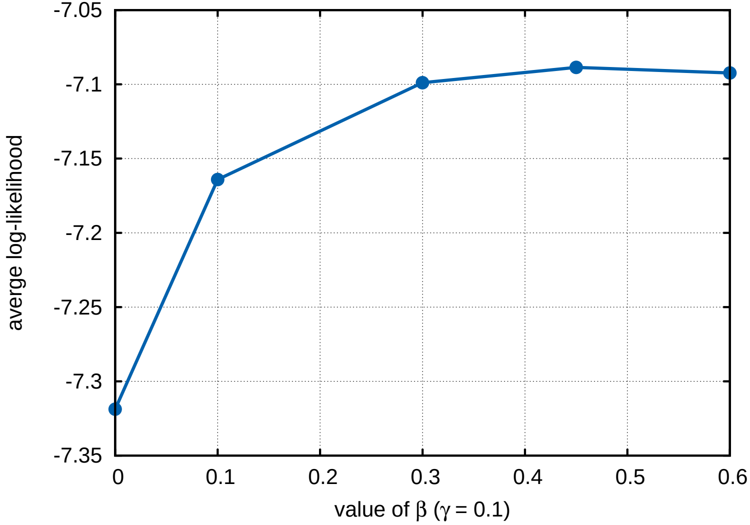


Overlapped-LME v.s. Multi-LME (KKBOX)

	Average log-likelihood	Operations	Time
Overlapped-LME (N=500)	-9.694	5.22×10^8	15hr
Multi-LME (k=300)	-10.451	5.68×10^8	16hr
Bigram	-10.606		

- Overlapped-LME is evaluated without two-steps probability.
- Bigram performs better due to higher similarity between training and testing data. (Yes.com: 19.3%, KKBOX: 35.5%)

Effect of two-steps probability

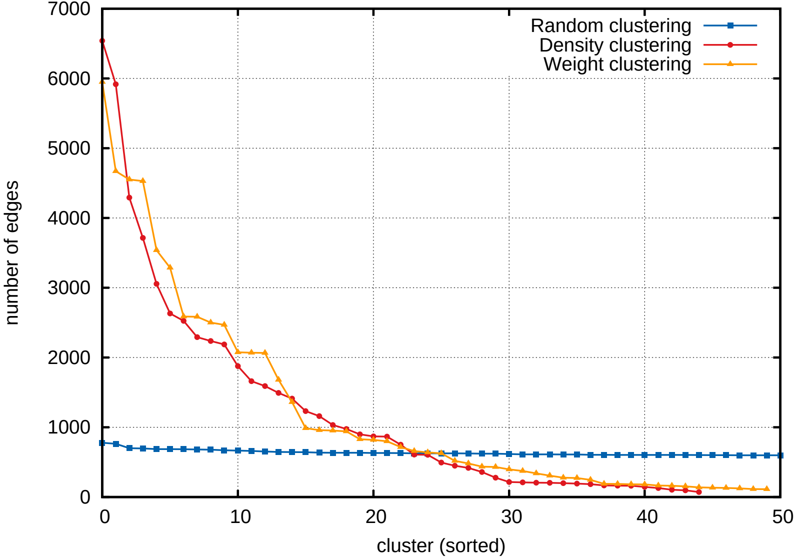


Effect of different clustering algorithms

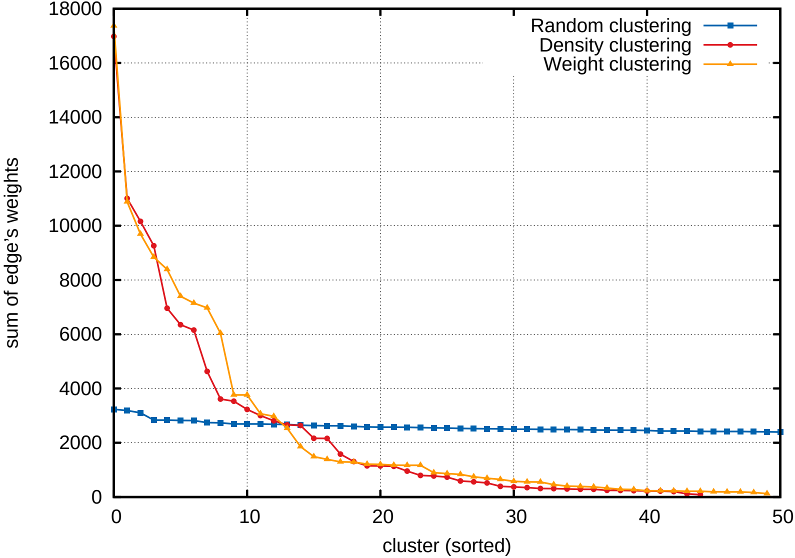
	Random	Density	Weight
Average log-likelihood	-7.573	-7.318	-7.186
Duplicated items	256,359	19,264	19,166
LME operations	13.2×10^7	1.15×10^7	1.28×10^7

- Size limit: $N = 500$.
- Without two-steps probability.

Edge density



Sum of edges' weights



Section 5

Conclusions & Future works

Conclusions

- A simpler interface for user (less parameter-tuning).
- Outperforms the current best speed-up approach on the same dataset (Yes.com).
- Tested on the larger scale, currently growing, user-crafted dataset (KKBOX) with good result.
- Introduces the possibility of *specific-purpose clusters*.

Future works

- Our overlapped design introduces the possibility of specific-purpose clusters (ex. items with similar observed features, items in the same time slot), and is able to combine these perspectives from different clusters to form the transition probability.

Thank you!

Pishen Tsai (Cheng-Hsuan Tsai)
pishen02@gmail.com