On Analyzing Graphs with Motif-Paths

Supplemental Material

XIAODONG LI[†], REYNOLD CHENG[†], KEVIN CHANG[‡], CAIHUA SHAN[†], CHENHAO MA[†], HONGTAI CAO[‡], [†]Department of Computer Science, University of Hong Kong, Hong Kong SAR
[‡]Department of Computer Science, University of Illinois at Urbana-Champaign, USA

ACM Reference Format:
Xiaodong Li[†], Reynold Cheng[†], Kevin Chang[‡], Caihua Shan[†], Chenhao Ma[†], Hongtai Cao[‡]. 2020. On Analyzing Graphs with Motif-Paths: Supplemental Material. 1, 1 (December 2020), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Because of the page limit in the paper, we report more details in the supplemental materials. In Section 1, we attach the proofs for lemmas (cf. Section 1.1), algorithm complexities (cf. Section 1.2). In Section 2, We report more efficiency and effectiveness evaluations, including:
•Fig. 1 MOD-Indexing time and size as graph density increases;
•Fig. 2 query time of SMP and ESMP as graph density increases;
•Fig. 3 breakdown of the offline indexing cost;
•Fig. 5 evaluation of generic motifs for link prediction on GAVI, EXTE and AMAZ;
•Tab. 2 link prediction performance with running time of each algorithm on each dataset;
•Tab. 3 local graph clustering performance with running time of each algorithm on each dataset;
•Fig. 4 local graph clustering performance comparison between MLGC-b and MLGC-c.;

1 PROOF

1.1 Proof of the Lemmas

Proof of Lemma 1:

Proof. For c1, there is no need to add motif-instances containing a "searched" node since all motif-instances around node marked as "searched" have been found and added into candidates for $\mathcal{P}_{s,t}$. For c2, for the motif-instances in $\mathcal{P}_{s,t}$, there are only two status of the nodes: "searched" and "discovered". We only select "discovered" node as next seed because the "searched" nodes have been used as seed before and thus using them as next seed will find duplicates. For c3, in the incremental search manner, $\mathcal{P}_{s,v}$ is found for the "undiscovered" node v when v is covered by any motif-instance for the first time. Therefore, we only add motif-instances which contain at least one node marked as "undiscovered" to push the incremental search forward. □

Author's address: Xiaodong Li[†], Reynold Cheng[†], Kevin Chang[‡], Caihua Shan[†], Chenhao Ma[†], Hongtai Cao[‡], [†]Department of Computer Science, University of Hong Kong, Hong Kong SAR
[‡]Department of Computer Science, University of Illinois at Urbana-Champaign, USA, {xdli,ckcheng,chshan,chma2}@cs.hku.hk;{kcchang, hongtai2}@illinois.edu.

XXXX-XXXX/2020/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Proof of Lemma 2:

Proof. Since $\mathcal{P}_{s,t}$ is the shortest sequence of motif-instances from m_s to m_t. For each motif-instance in the sequence, we pick the edge which links s, t and the nodes shared by the neighboring motif-instances, the path is the shortest one on W. Vise versa. □

Proof of Lemma 3:

Proof. Assume that $\exists(i,j) \in V \times V$, $W_{i,j} = 1$ but there is no motif-instance of $\bar{\tau}$, then there must be motif-instance m such that $m \simeq \bar{\tau}' \& (i,j) \in E_m$, where $\bar{\tau}'$ is another motif-orbit of τ with seed $s \in V_m$. By switching the seed node, $m \simeq \bar{\tau}$ with seed $s' \in V_m$, which is contradictory to the assumption. □

Proof of expansive degree:

1.2 Time and Space Complexities

Table 1. Summary of algorithm complexities.

Alg.	Time	Space
BASE	$\mathcal{O}(N_\tau^3)$	$\mathcal{O}(N_\tau^2)$
MODC	$\mathcal{O}(\sum_{s\in V}\sum_{\tau'\in B_k} D_{\tau'}^{k})$	$\mathcal{O}(I)$
MODCt	$\mathcal{O}(\lvert V\rvert \times d_{\max}^2)$	$\mathcal{O}(\lvert V\rvert \times \binom{d_{\max}}{2})$
MODQ	$\mathcal{O}(D_\tau^{\phi_\tau - k})$	$\mathcal{O}(I)$
SMP	$\mathcal{O}(\lvert V\rvert \times D_\tau^{\phi_\tau - k} + N_\tau)$	$\mathcal{O}(\lvert V\rvert + N_\tau + I)$
ESMP	$\mathcal{O}(\lvert V\rvert \times D_\tau^{\phi_\tau - k} + \lvert E\rvert + N_\tau)$	$\mathcal{O}(\lvert V\rvert + \lvert E\rvert + N_\tau + I)$
MGD	$\mathcal{O}(\lvert V\rvert \times D_\tau^{\phi_\tau - k} + N_\tau)$	$\mathcal{O}(\lvert V\rvert + N_\tau + I)$
MKI	$\mathcal{O}((\lvert V\rvert \times D_\tau^{\phi_\tau - k})^{2L} + N_\tau)$	$\mathcal{O}(\lvert V\rvert^{2L} + N_\tau + I)$
MLGC	$\mathcal{O}(\lvert V\rvert \times \hat{k} \times D_\tau^{\phi_\tau - k} + N_\tau)$	$\mathcal{O}(d_{\max} \times \hat{k} + \binom{\hat{k}}{\lvert V_\tau\rvert} + I)$
MBET	$\mathcal{O}(\lvert V\rvert \times (D_\tau^{\phi_\tau - k} + \lvert V\rvert + \lvert E_W\rvert) + N_\tau)$	$\mathcal{O}(\binom{d_{\max}^{\phi_\tau}}{\lvert V_\tau\rvert} + \lvert V\rvert^2 + I)$

Here $N_\tau = \binom{|V|}{|V_\tau|}$, $D_\tau = \binom{d_{\max}}{\tau.d_e}$ and $I = \sum_{s\in V}\sum_{\tau'\in B_k}\binom{d_{\max}^{\phi_{\tau'}}}{|V_{\tau'}|-1}$.

2 SUPPLEMENTAL EVALUATIONS

Fig. 1. MOD-Indexing time and space cost as graph density grows, with shortest motif-path query time reported.

Fig. 2. Query time of SMP and ESMP as graph density increases.

Fig. 3. Breakdown of the offline indexing cost.

Table 2. MKI/MGD performance with AUC and running time reported. Numbers of top-3 highest AUC are marked bold.

Method	**GAVI**		**EXTE**		**DBLP**		**AMAZ**		**YOUT**	
	AUC	Time	AUC	Time	AUC	Time	AUC	Time	AUC	Time
CN	0.72	1.2ms	0.56	1.7ms	0.77	12.9ms	0.62	13.5ms	0.54	0.2s
JC	0.7	1.1ms	0.48	1.6ms	0.55	14.4ms	0.52	12.8ms	0.44	0.2s
AA	0.75	1.2ms	0.57	1.6ms	**0.81**	12.0ms	0.65	12.6ms	0.52	0.2s
PA	0.59	1.2ms	0.76	1.7ms	0.64	13.4ms	0.63	14.9ms	0.76	0.2s
FM	0.65	1.2ms	0.65	1.5ms	0.59	14.0ms	0.64	15.4ms	0.69	0.2s
HT	0.6	1.4ms	0.7	1.7ms	0.64	13.5ms	0.59	15.2ms	0.53	70.1s
RPR	0.61	1.5ms	0.51	1.6ms	0.76	12.7ms	0.62	13.4ms	0.48	72.0s
MCN	0.67	1.0s	0.62	3.2s	0.75	38.8s	0.61	2.4s	0.65	16.6m
MLP+GB	**0.89**	7.0m	**0.83**	76.9m	**0.82**	35.9m	**0.72**	1.4m	**0.83**	70.4h
KI	0.69	36.2ms	0.6	0.5s	0.69	2.2s	0.6	97.5ms	0.63	1.1m
MKI	0.71	0.1s	0.66	1.2s	0.74	23.5s	0.63	6.1s	0.66	5.7m
MKI-c	0.62	6.4m	0.67	18.0m	-	-	-	-	-	-
MKI-b	**0.76**	0.1s	**0.87**	2.0s	**0.75**	31.2s	**0.73**	10.6s	**0.76**	7.5m
GD	0.5	2.2ms	0.5	3.1ms	0.5	23.0ms	0.5	26.7ms	0.5	1.8s
MGD	0.67	50.1ms	0.63	0.2s	0.65	2.5s	0.66	1.8s	0.55	1.4m
MGD-c	0.52	1.3m	0.51	3.2m	-	-	-	-	-	-
MGD-b	**0.75**	32.7ms	**0.84**	30.6ms	0.72	2.8s	**0.81**	0.9s	**0.87**	1.8s

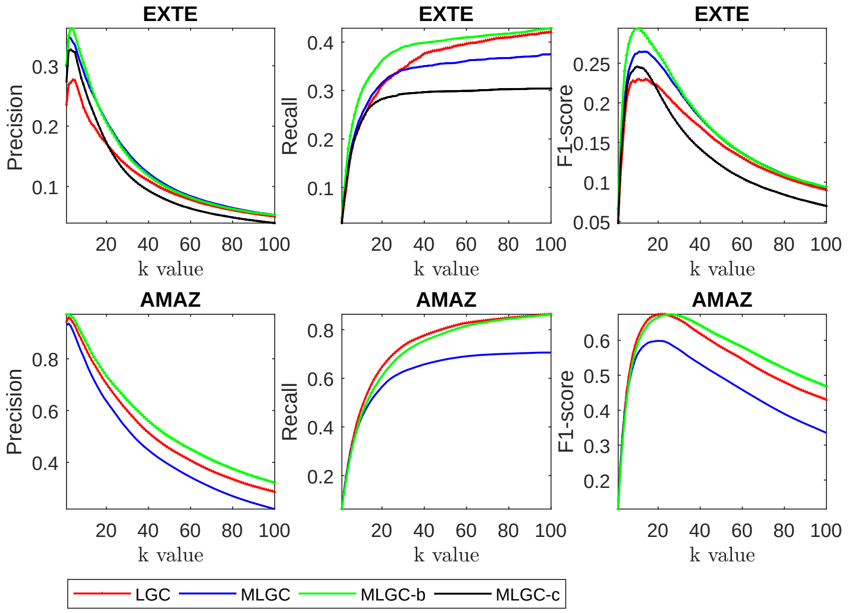

Fig. 4. Local graph clustering results on EXTE and AMAZ Precision, Recall and F1-score reported.

Table 3. MLGC performance with F1-score and running time reported. Numbers of top-3 highest F1-score are marked bold.

Method	GAVI		EXTE		DBLP		AMAZ		YOUT	
	F1	Time	F1	Time	F1	Time	F1	Time	F1	Time
TECTONIC	0.39	7.0s	**0.44**	24.6s	-	-	0.37	-	-	-
MAPPR	0.39	0.2s	**0.42**	0.1s	**0.34**	5.1s	0.35	4.4s	0.15	16.8s
EdMot	0.33	21.1s	0.38	1.1m	-	-	-	-	-	-
LGC	**0.42**	9ms	0.36	12.2ms	0.33	1.3s	**0.63**	89.6ms	**0.17**	0.7s
MLGC	**0.41**	3.1ms	0.3	8.7ms	**0.35**	1.3s	**0.59**	7.8ms	**0.16**	8.7s
MLGC-c	0.39	23.1s	0.29	1.1m	-	-	-	-	-	-
MLGC-b	**0.42**	3.7ms	**0.38**	9.9ms	**0.35**	1.5s	**0.65**	8.8ms	**0.23**	13.8s

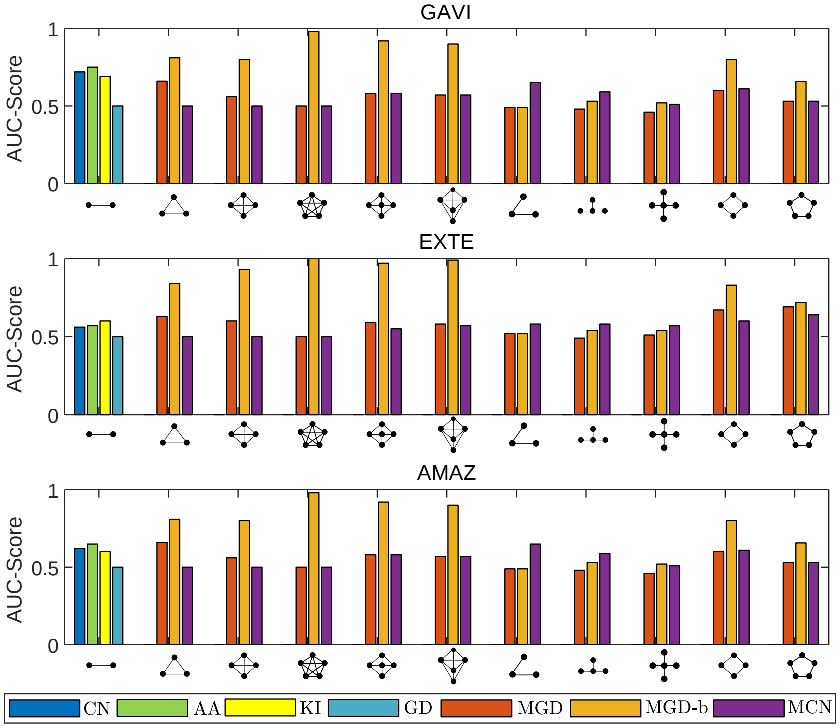

Fig. 5. Evaluation of generic motifs (edge, cliques, quasi-cliques, stars and cycles) for link prediction on GAVI, EXTE and AMAZ.