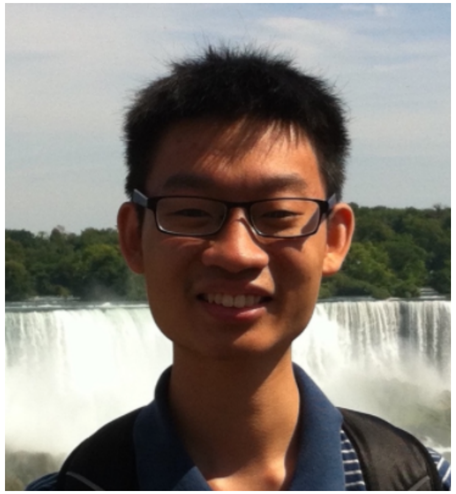
Autonomous Graph Mining Algorithm Search with Best Speed/Accuracy Trade-off

Minji Yoon
Carnegie Mellon University

Théophile Gervet
Carnegie Mellon University

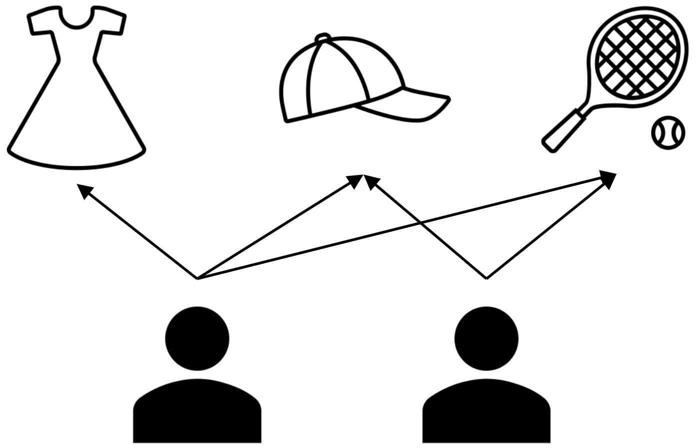


Bryan Hooi
National University of Singapore

Christos Faloutsos
Carnegie Mellon University

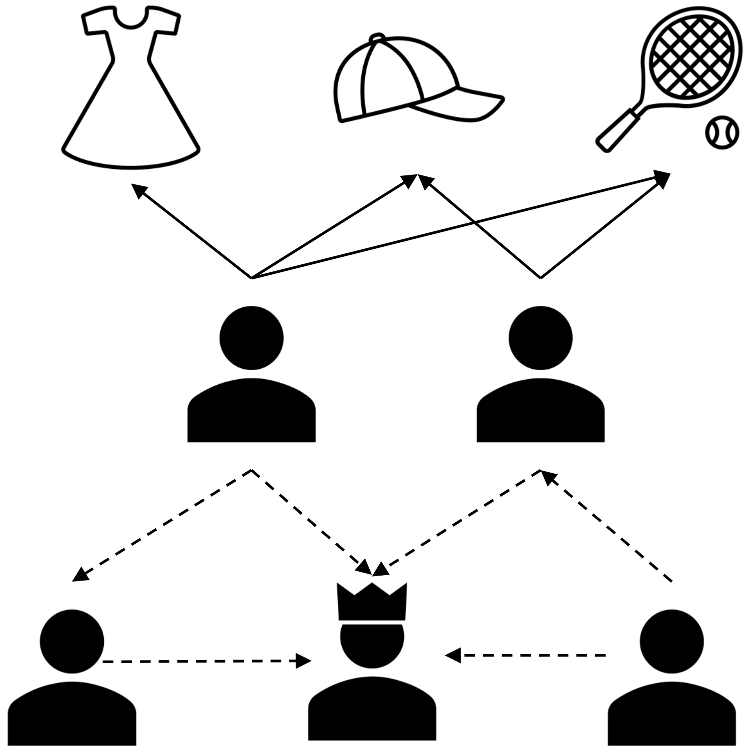
Graph Data is Ubiquitous

- Who-buys-which-products
- Who-follows-whom
- Who-pays-whom
- Protein relationship



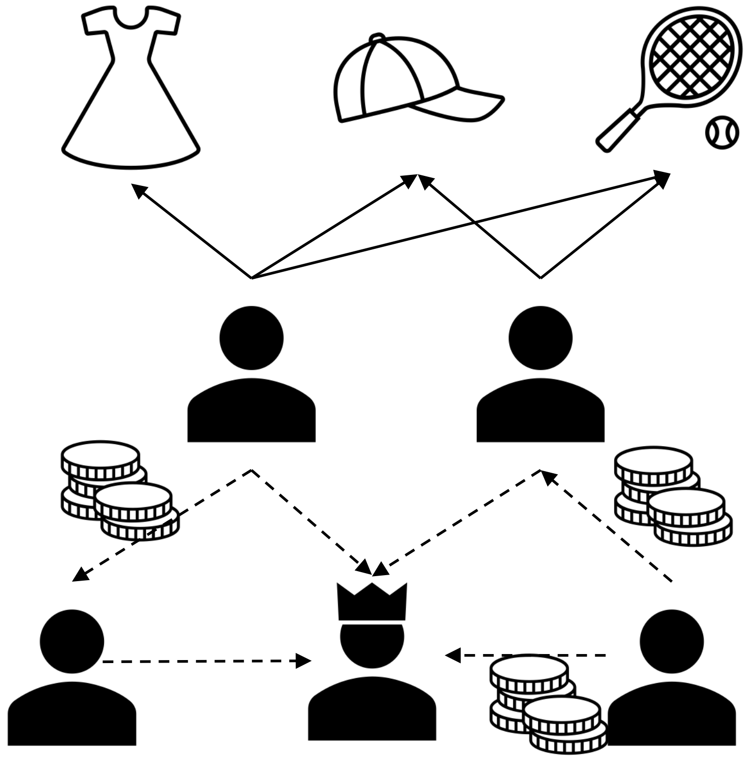
Graph Data is Ubiquitous

- Who-buys-which-products
- Who-follows-whom
- Who-pays-whom
- Protein relationship



Graph Data is Ubiquitous

- Who-buys-which-products
- Who-follows-whom
- Who-pays-whom
- Protein relationship

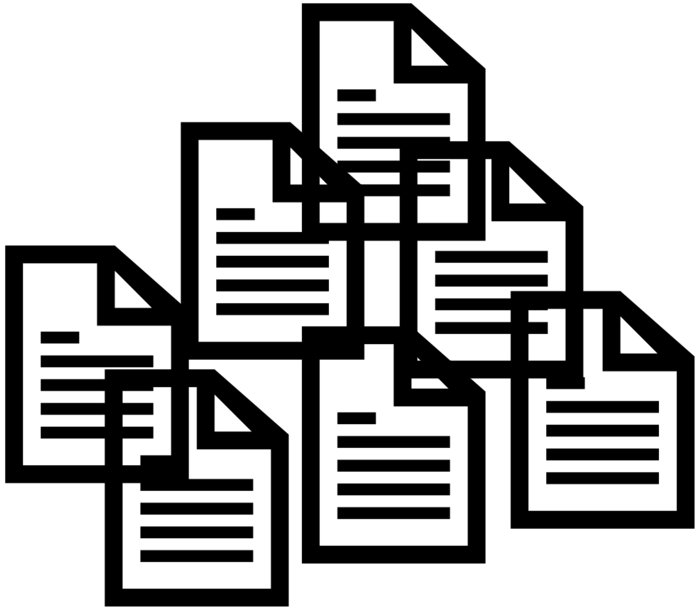


Graph Mining Algorithms are diverse

- Classification of web documents
- Clustering in market segmentation
- Recommendation in streaming services
- Fraud detection in banking

Graph Mining Algorithms are diverse

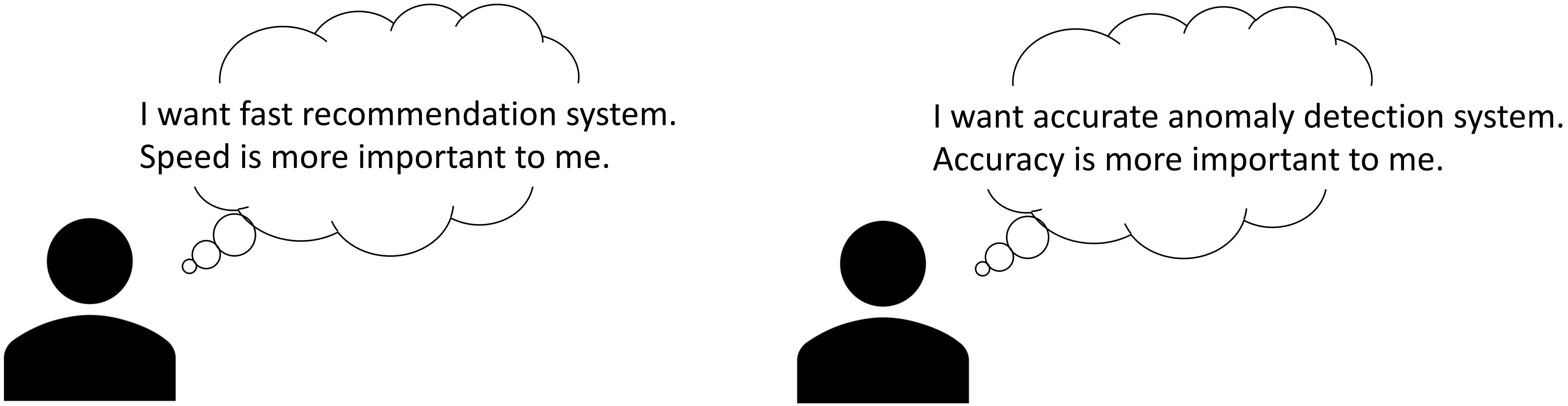
- Classification of web documents
- Clustering in market segmentation
- Recommendation in streaming services
- Fraud detection in banking



200+ graph mining papers published in 2019

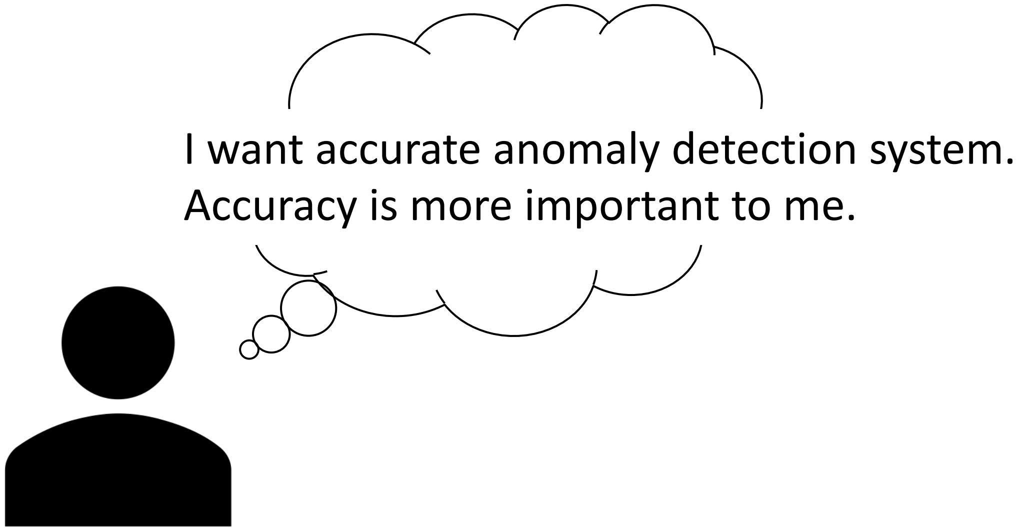
Lack of Unity in Graph Mining

- Which graph mining algorithm should we choose?



Lack of Unity in Graph Mining

- Distinct problem definitions and conceptual formulations
- Require expert experience and brute-force search



We propose AutoGM

- **Unify** various graph mining algorithms in one framework

We propose AutoGM

- **Unify** various graph mining algorithms in one framework
- **Automate** the generation of graph mining algorithm

We propose AutoGM

- **Unify** various graph mining algorithms in one framework
- **Automate** the generation of graph mining algorithm
- Deploy graph mining algorithms **tailored** to their scenarios

Overview

1. Motivation
2. **UnifiedGM:** Unified Graph Mining Framework
3. **AutoGM:** Automation of Graph Mining Algorithm Development
4. **Experiments**
5. **Conclusion**

Unification

- Message Passing
- UnifiedGM
- Reproduction of Existing Algorithms
- Conventional GM vs. GNNs

Message Passing (1)

- What is Graph Mining?
  - **GIVEN**: global graph information
  - e.g., edge structure and feature information from other nodes
  - **ANSWER**: queries at the node level
  - e.g., node clustering, classification, or recommendation

Message Passing (2)

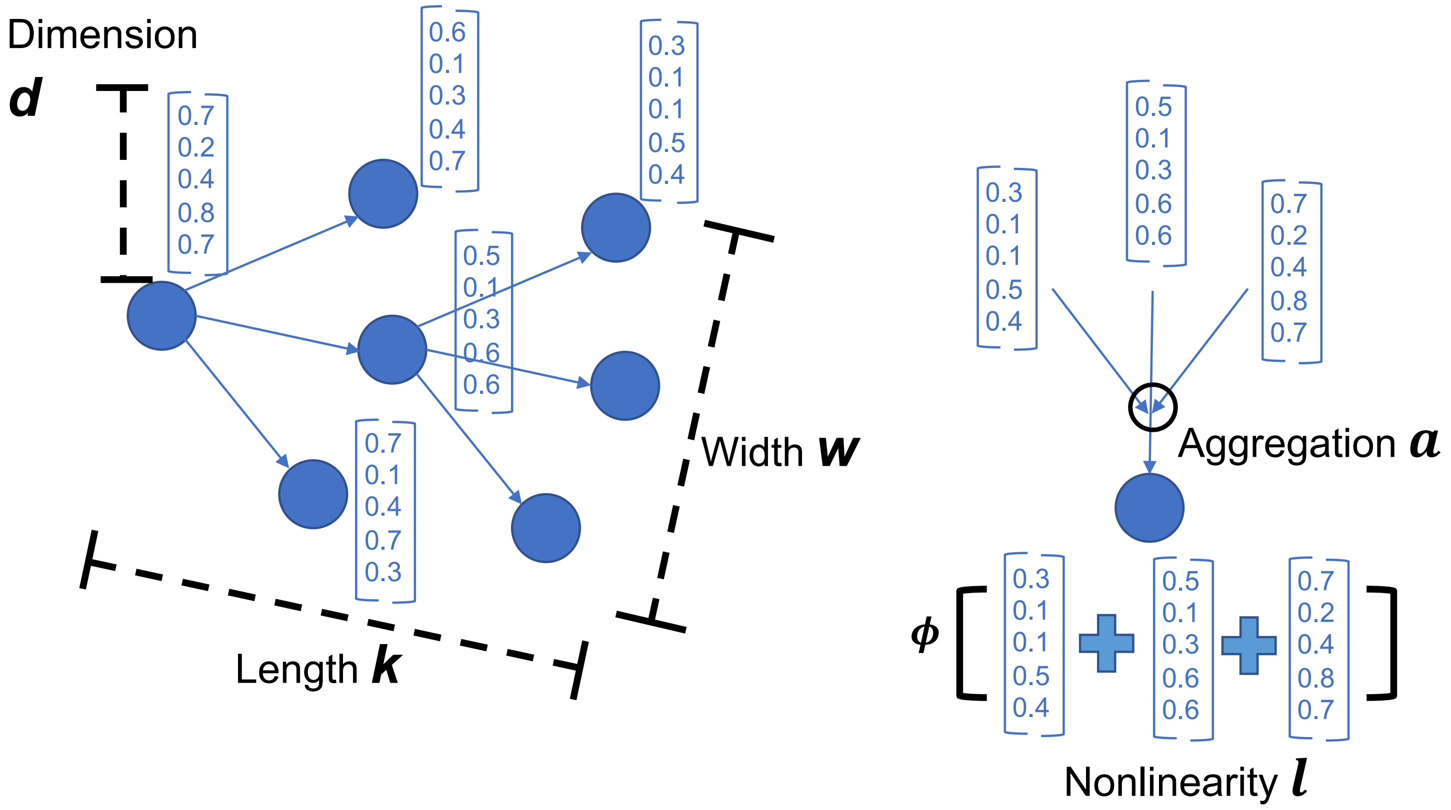
- What is Graph Mining?
  - GIVEN: global graph information
  - e.g., edge structure and feature information from other nodes
  - ANSWER: queries at the node level
  - e.g., node clustering, classification, or recommendation
- **Target graph algorithms**
  - Message passing mechanism
  - Transmit the information necessary to answer such queries

Message Passing (3)

- Ex1: classical graph mining algorithms
  - PageRank
  - nodes *propagate scalar scores to* their neighbors
- Ex2: graph neural networks
  - GCN
  - nodes *aggregate feature vectors from* their neighbors

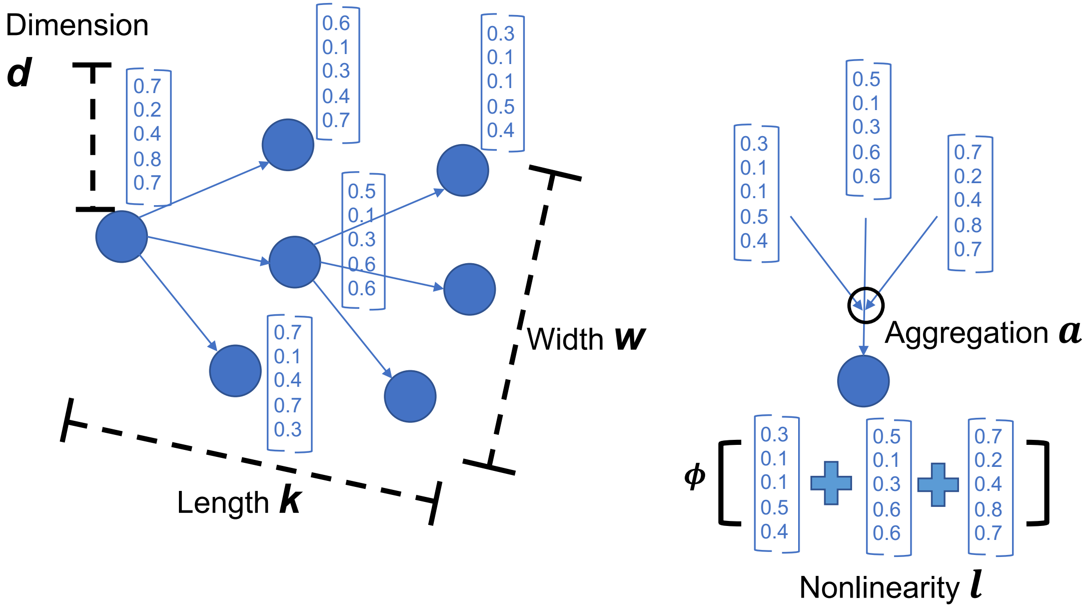
UnifiedGM (1)

- Dimension $\boldsymbol{d}$
- Width $\boldsymbol{w}$
- Length $\boldsymbol{k}$
- Nonlinearity $\boldsymbol{l}$
- Aggregation strategy $\boldsymbol{a}$



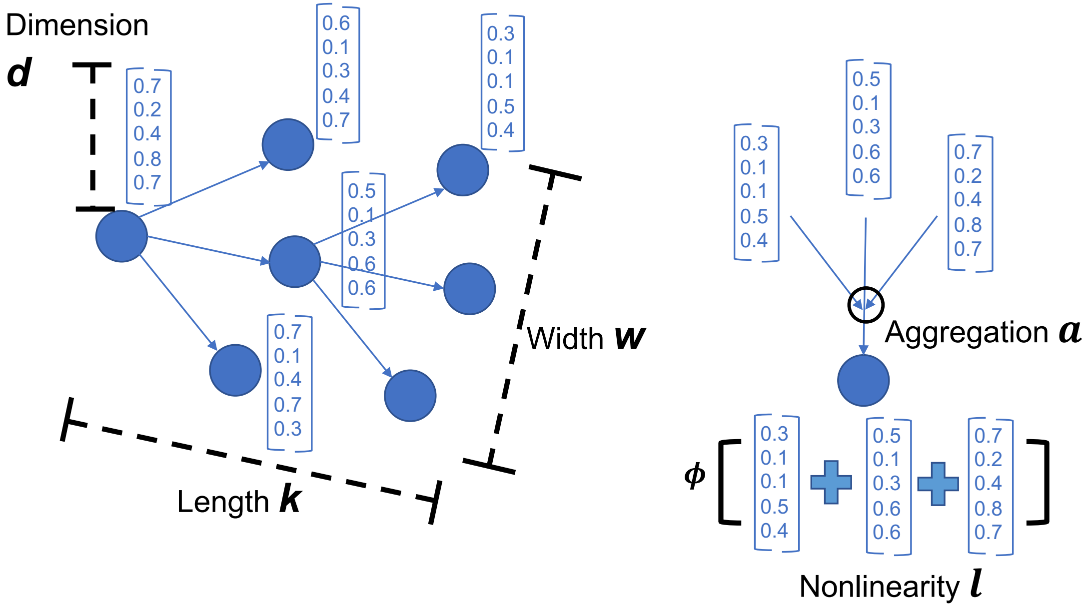
UnifiedGM (1)

- Dimension $\boldsymbol{d}$ of passed messages
  - $\boldsymbol{d}$=1, messages are scalar scores
- Width $\boldsymbol{w}$
- Length $\boldsymbol{k}$
- Nonlinearity $\boldsymbol{l}$
- Aggregation strategy $\boldsymbol{a}$



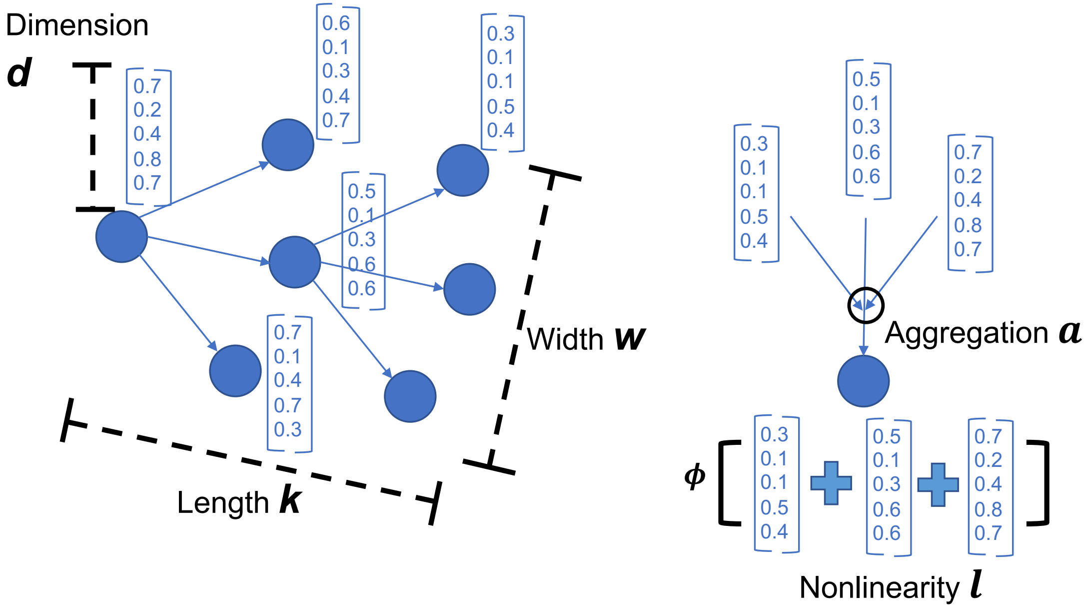
UnifiedGM (1)

- Dimension $\boldsymbol{d}$
- Width $\boldsymbol{w}$
  - Number of neighbors
  - $\boldsymbol{w}$=-1, all neighbors
- Length $\boldsymbol{k}$
- Nonlinearity $\boldsymbol{l}$
- Aggregation strategy $\boldsymbol{a}$



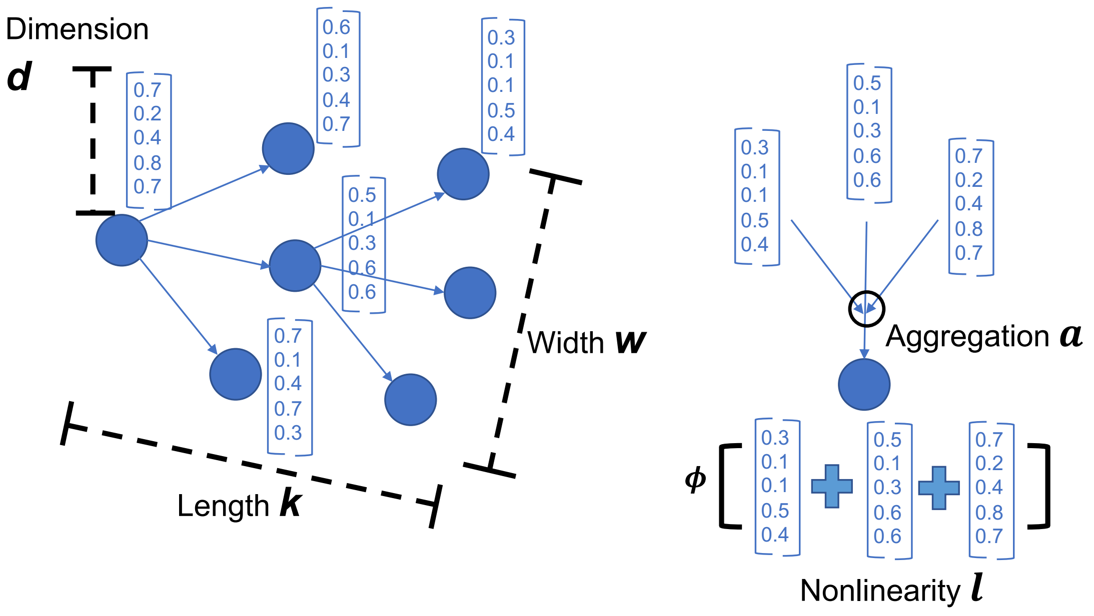
UnifiedGM (1)

- Dimension $\boldsymbol{d}$
- Width $\boldsymbol{w}$
- Length $\boldsymbol{k}$
  - Number of message passing steps
- Nonlinearity $\boldsymbol{l}$
- Aggregation strategy $\boldsymbol{a}$



UnifiedGM (1)

- Dimension $\boldsymbol{d}$
- Width $\boldsymbol{w}$
- Length $\boldsymbol{k}$
- Nonlinearity $\boldsymbol{l}$
  - Nonlinearity in the message passing
- Aggregation strategy $\boldsymbol{a}$



UnifiedGM (1)

- Dimension $\boldsymbol{d}$
- Width $\boldsymbol{w}$
- Length $\boldsymbol{k}$
- Nonlinearity $\boldsymbol{l}$
- Aggregation strategy $\boldsymbol{a}$

	Self-loop (S)	No-self-loop (N)
Asymmetric (A)	$D^{-1}(A+I_n)$	$D^{-1}A$
Symmetric (S)	$D^{-1/2}(A+I_n)D^{-1/2}$	$D^{-1/2}AD^{-1/2}$
No-normalization (N)	$(A+I_n)$	A

UnifiedGM (2)

Width w

Aggregation a

$$
\begin{aligned}
A_{\text{samp}} &= \mathit{Sample}(A) \\
A_{\text{agg}} &= \mathit{Aggregate}(A_{\text{samp}}) \\
X_k &= f_k(X_{k-1}) = \phi(A_{\text{agg}} X_{k-1} W_k) \\
&= f_k(f_{k-1}(\dots f_1(X_0)))
\end{aligned}
$$

Dimension d

Nonlinearity l

Length k

X_0: $(n \times d_0)$ initial statistic vectors

X_i: $(n \times d)$ statistic vectors at step i

$(i = 1, \cdots, k)$

W_1: $(d_0 \times d)$ transformation matrix

W_i: $(d \times d)$ transformation matrix

$(i = 2, \cdots, k)$

Reproduction of Existing Algorithms

Algorithm	Original message passing equation	Initial node statistics	d	k	w	l	a
PageRank [19]	$X_k = c(D^{-1}A)X_{k-1}$	$\frac{1}{n}$ for all nodes	1	∞	-1	False	NA
Pixie [6]	$X_k(u) = \sum_{v \in \mathrm{N}(u)} X_{k-1}(v)$	1 for seeds, 0 others	1	sample	$\frac{2000}{k}$	False	NN
GCN [11]	$X_k = ReLU\left((D^{-\frac{1}{2}}(A+I_n)D^{-\frac{1}{2}})X_{k-1}W_k\right)$	feature vectors	64	2	-1	True	SS
GraphSAGE [8]	$X_k(u) = ReLU\left(\frac{1}{\lvert\mathrm{N}(u)\rvert+1} \sum_{v \in \mathrm{N}(u) \cup u} X_{k-1}(v)W_k\right)$	feature vectors	64	2	25	True	SA
SGCN [25]	$X_k = D^{-\frac{1}{2}}(A+I_n)D^{-\frac{1}{2}}X_{k-1}W_k$	feature vectors	64	2	-1	False	SS

- Initial node statistics X_0
- Parameters of UnifiedGM $(\mathbf{d}, \mathbf{k}, \mathbf{w}, \boldsymbol{l}, \boldsymbol{a})$

Reproduction of Existing Algorithms

Algorithm	Original message passing equation	Initial node statistics	d	k	w	l	a
PageRank [19]	$X_k = c(D^{-1}A)X_{k-1}$	$\frac{1}{n}$ for all nodes	1	∞	-1	False	NA
Pixie [6]	$X_k(u) = \sum_{v \in \mathrm{N}(u)} X_{k-1}(v)$	1 for seeds, 0 others	1	sample	$\frac{2000}{k}$	False	NN
GCN [11]	$X_k = ReLU\left((D^{-\frac{1}{2}}(A+I_n)D^{-\frac{1}{2}})X_{k-1}W_k\right)$	feature vectors	64	2	-1	True	SS
GraphSAGE [8]	$X_k(u) = ReLU\left(\frac{1}{\lvert\mathrm{N}(u)\rvert+1}\sum_{v \in \mathrm{N}(u) \cup u} X_{k-1}(v)W_k\right)$	feature vectors	64	2	25	True	SA
SGCN [25]	$X_k = D^{-\frac{1}{2}}(A+I_n)D^{-\frac{1}{2}}X_{k-1}W_k$	feature vectors	64	2	-1	False	SS

- Initial node statistics X_0
- Parameters of UnifiedGM $(\mathbf{d}, \mathbf{k}, \mathbf{w}, \boldsymbol{l}, \boldsymbol{a})$

Reproduction of Existing Algorithms

Algorithm	Original message passing equation	Initial node statistics	d	k	w	l	a
PageRank [19]	$X_k = c(D^{-1}A)X_{k-1}$	$\frac{1}{n}$ for all nodes	1	∞	-1	False	NA
Pixie [6]	$X_k(u) = \sum_{v \in N(u)} X_{k-1}(v)$	1 for seeds, 0 others	1	sample	$\frac{2000}{k}$	False	NN
GCN [11]	$X_k = ReLU\left((D^{-\frac{1}{2}}(A+I_n)D^{-\frac{1}{2}})X_{k-1}W_k\right)$	feature vectors	64	2	-1	True	SS
GraphSAGE [8]	$X_k(u) = ReLU\left(\frac{1}{\lvert N(u)\rvert+1} \sum_{v \in N(u) \cup u} X_{k-1}(v)W_k\right)$	feature vectors	64	2	25	True	SA
SGCN [25]	$X_k = D^{-\frac{1}{2}}(A+I_n)D^{-\frac{1}{2}} X_{k-1}W_k$	feature vectors	64	2	-1	False	SS

- Initial node statistics X_0
- Parameters of UnifiedGM $(\mathbf{d}, \mathbf{k}, \mathbf{w}, \boldsymbol{l}, \boldsymbol{a})$

Conventional GM vs. GNNs (1)

Conventional Graph Mining (e.g., PageRank)		
Graph Neural Networks (e.g., GCN)		

Conventional GM vs. GNNs (1)

	Node feature information	Semi-supervised learning
Conventional Graph Mining (e.g., PageRank)		
Graph Neural Networks (e.g., GCN)		

Conventional GM vs. GNNs (2)

	Node feature information	Semi-supervised learning
Conventional Graph Mining (e.g., PageRank)	A set of seed nodes to initialize with scores (d = 1)	No training phase
Graph Neural Networks (e.g., GCN)	Node features (d > 1)	Transformation matrix W is trained using node labels

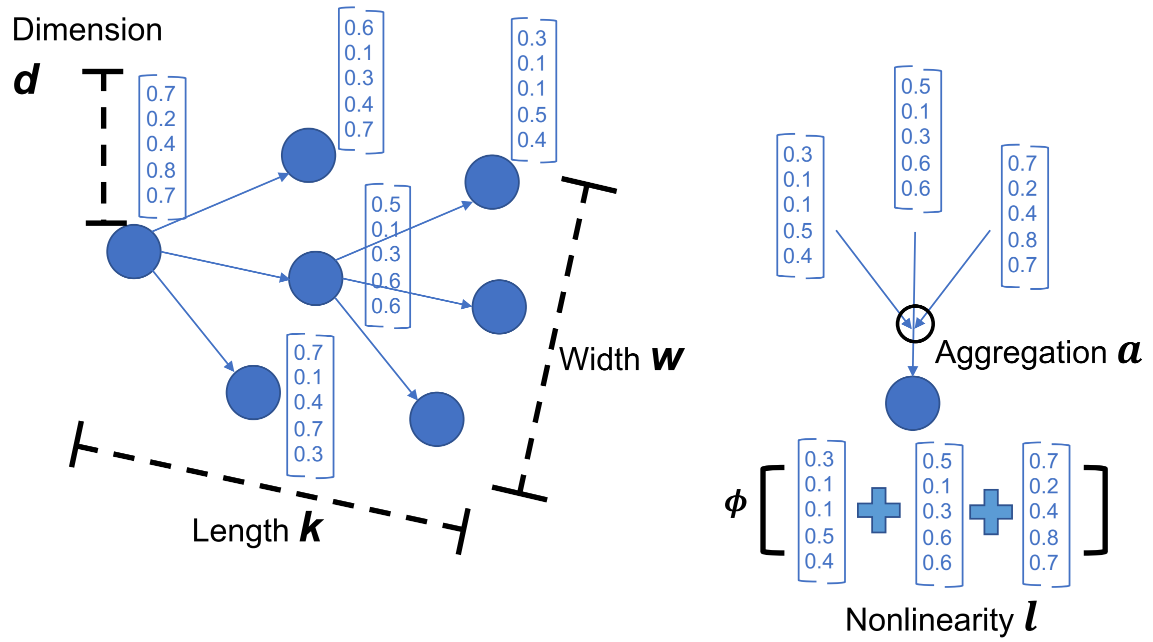
Conventional GM vs. GNNs (3)

	Node feature information	Semi-supervised learning
Conventional Graph Mining (e.g., PageRank)	**We can use node features !!!** • **Initial input dimension d_0= # feature** • **1st layer transformation matrix W_1 of size ($d_0 \times 1$)**	No training phase
Graph Neural Networks (e.g., GCN)	Node features (d > 1)	Transformation matrix W is trained using node labels

Conventional GM vs. GNNs (4)

	Node feature information	Semi-supervised learning
Conventional Graph Mining (e.g., PageRank)	We can use node features • Initial input dimension d_0= # feature • 1st layer transformation matrix W_1 of size $(d_0 \times 1)$	**We are learnable !!!** • **Decaying coefficient c corresponds to (1×1) transformation matrix W** • **Set heuristically (e.g., $c = 0.85$)** • **Trainable with gradient descent**
Graph Neural Networks (e.g., GCN)	Node features (d > 1)	Transformation matrix W is trained using node labels

Parameter Selection



Given an application,

how could we choose proper parameters $(\mathbf{d}, \mathbf{k}, \mathbf{w}, \boldsymbol{l}, \boldsymbol{a})$?

Overview

1. **Motivation**
2. **UnifiedGM:** Unified Graph Mining Framework
3. **AutoGM:** Automation of Graph Mining Algorithm Development
4. **Experiments**
5. **Conclusion**

AutoGM (1)

Given a user's scenario

- Generate an optimal graph algorithm autonomously

AutoGM (2)

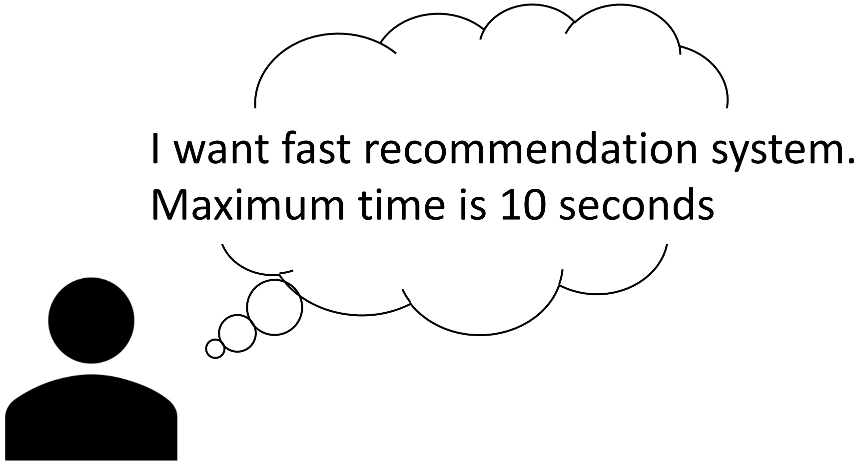
Given a user's scenario

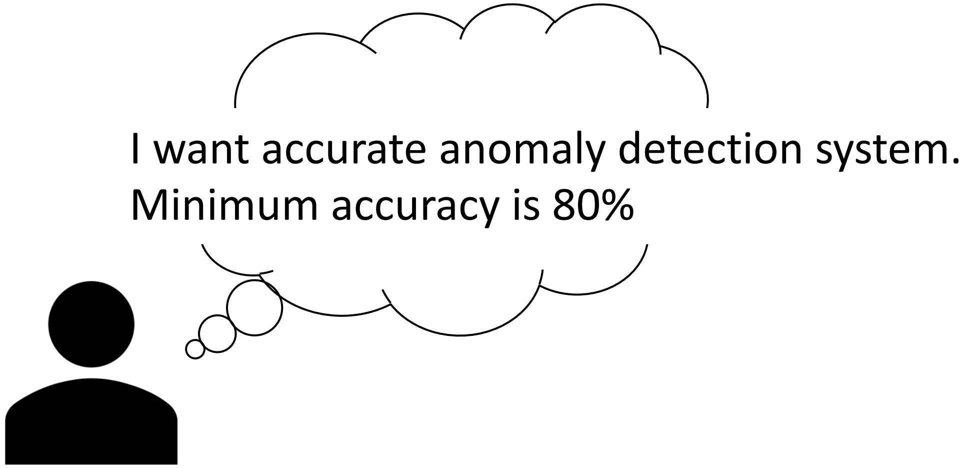
- ~~Generate an optimal graph algorithm autonomously~~
- Find the optimal parameters $(\mathbf{d}, \mathbf{k}, \mathbf{w}, \boldsymbol{l}, \boldsymbol{a})$ of UnifiedGM autonomously

AutoGM (3)

Given a user's scenario

- ~~Generate an optimal graph algorithm autonomously~~
- Find the optimal parameters $(\mathbf{d}, \mathbf{k}, \mathbf{w}, \boldsymbol{l}, \boldsymbol{a})$ of UnifiedGM autonomously



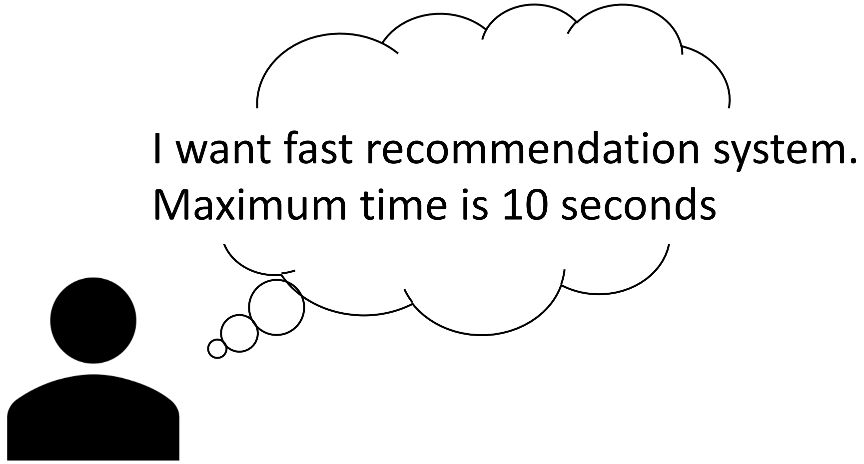


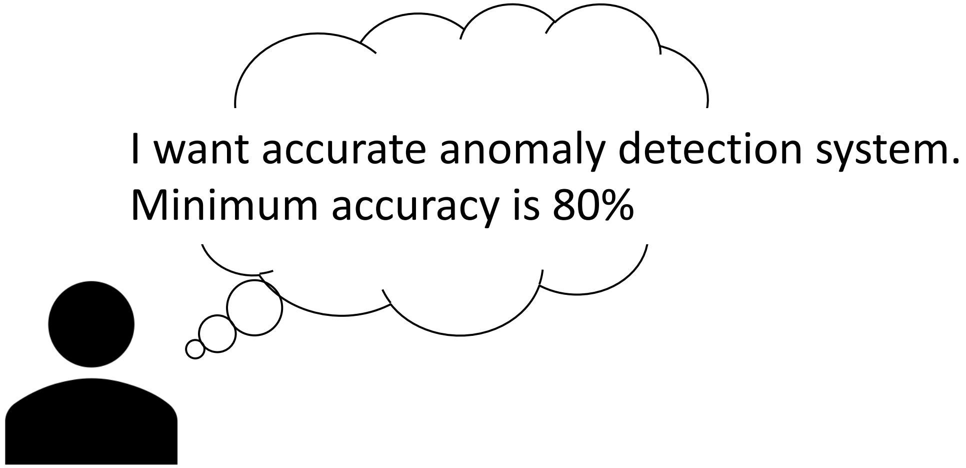
AutoGM (3)

~~Given a user's scenario~~

Given a user's budget on **computation time and accuracy**

- ~~Generate an optimal graph algorithm autonomously~~
- Find the optimal parameters $(\mathbf{d}, \mathbf{k}, \mathbf{w}, \boldsymbol{l}, \boldsymbol{a})$ of UnifiedGM autonomously





Budget-aware objective function (1)

$$x_{opt} = \text{argmin}_x \; g(x) \text{ subject to } h(x) - h_{\min} \geq 0$$

- **Graph algorithm $\boldsymbol{x} = (\mathbf{d}, \mathbf{k}, \mathbf{w}, \boldsymbol{l}, \boldsymbol{a})$**

- **Minimum accuracy constraint $\boldsymbol{acc_{min}}$**
  - $g(x) = time$
  - $h(x) - h_{min} = acc - acc_{min}$

- **Maximum inference time $\boldsymbol{time_{max}}$**
  - $g(x) = -acc$
  - $h(x) - h_{min} = time_{max} - time$

Budget-aware objective function (2)

$$x_{opt} = \operatorname{argmin}_x \ g(x) \text{ subject to } h(x) - h_{\min} \geq 0$$

Barrier Method

$$f_{GM}(x) = g(x) - \lambda \log(h(x) - h_{\min})$$
$$x_{opt} = \operatorname{argmin}_x \ f_{GM}(x)$$

- **Graph algorithm $\boldsymbol{x} = (\mathbf{d}, \mathbf{k}, \mathbf{w}, \boldsymbol{l}, \boldsymbol{a})$**
- **Minimum accuracy constraint acc_{min}**
  - $g(x) = time$
  - $h(x) - h_{min} = acc - acc_{min}$
- **Maximum inference time $time_{max}$**
  - $g(x) = -acc$
  - $h(x) - h_{min} = time_{max} - time$

AutoGM with Bayesian Optimization

- **Graph G**
- **User's Budget**

(acc_{min} or $time_{max}$)

AutoGM

(Bayesian Optimization)

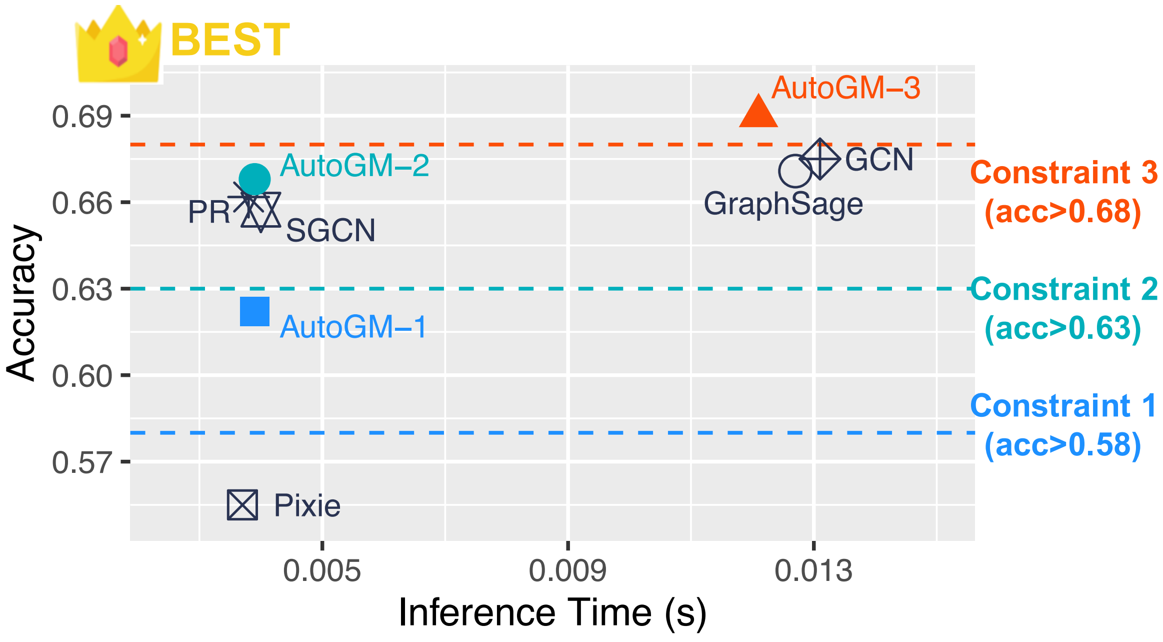
Optimal graph algorithm

$x = (\mathrm{d}, \mathrm{k}, \mathrm{w}, l, a)$ for Graph G

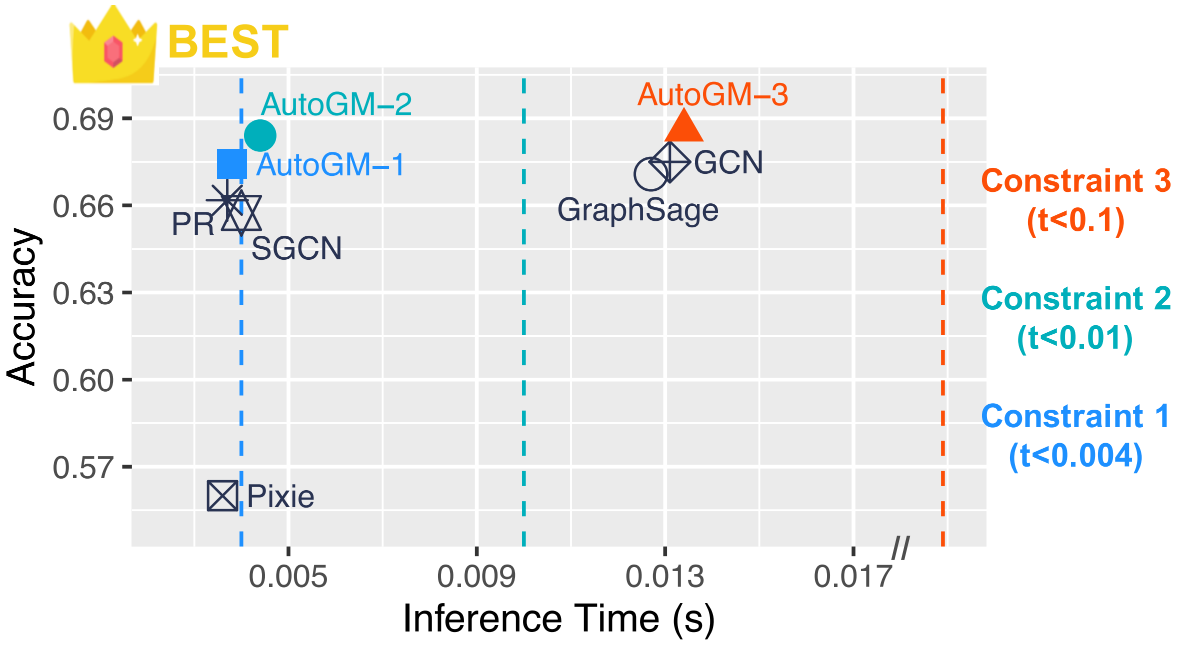
Overview

1. **Motivation**
2. **UnifiedGM:** Unified Graph Mining Framework
3. **AutoGM:** Automation of Graph Mining Algorithm Development
4. **Experiments**
5. **Conclusion**

Effectiveness of AutoGM



(a) Accuracy constraints on the Citeseer dataset



(b) Time constraints on the Citeseer dataset

Efficiency of AutoGM (1)

Dataset	Search(s)	Min.Acc.	Fastest Inference (s) AutoGM	Fastest Inference (s) Random	Accuracy AutoGM	Accuracy Random	Max.Time(s)	Highest Accuracy AutoGM	Highest Accuracy Random	Inference (s) AutoGM	Inference (s) Random
Cora	450	0.78	**0.0034**	-	0.79	-	0.004	**0.77**	**0.77**	0.0036	0.0033
Citeseer	800	0.67	**0.0039**	**0.0039**	0.67	0.67	0.004	**0.67**	-	0.0039	-
Pubmed	1,800	0.75	**0.021**	-	0.77	-	0.004	**0.76**	0.71	0.0036	0.0039
AmazonC	5,700	0.85	**0.032**	0.033	0.89	0.87	0.04	**0.85**	-	0.032	-
AmazonP	18,000	0.93	**0.047**	0.065	0.94	0.93	0.05	**0.94**	-	0.048	-
CoauthorC	2,500	0.8	**0.015**	0.016	0.8	0.82	0.02	**0.83**	0.75	0.015	0.02
CoauthorP	1,500	0.9	**0.01**	-	0.91	-	0.01	**0.92**	0.86	0.01	0.01

Efficiency of AutoGM (1)

Dataset	Search(s)	Min.Acc.	Fastest Inference (s)		Accuracy		Max.Time(s)	Highest Accuracy		Inference (s)	
			AutoGM	Random	AutoGM	Random		AutoGM	Random	AutoGM	Random
Cora	450	0.78	**0.0034**	-	0.79	-	0.004	**0.77**	**0.77**	0.0036	0.0033
Citeseer	800	0.67	**0.0039**	**0.0039**	0.67	0.67	0.004	**0.67**	-	0.0039	-
Pubmed	1,800	0.75	**0.021**	-	0.77	-	0.004	**0.76**	0.71	0.0036	0.0039
AmazonC	5,700	0.85	**0.032**	0.033	0.89	0.87	0.04	**0.85**	-	0.032	-
AmazonP	18,000	0.93	**0.047**	0.065	0.94	0.93	0.05	**0.94**	-	0.048	-
CoauthorC	2,500	0.8	**0.015**	0.016	0.8	0.82	0.02	**0.83**	0.75	0.015	0.02
CoauthorP	1,500	0.9	**0.01**	-	0.91	-	0.01	**0.92**	0.86	0.01	0.01

- Random Search fails to find any algorithm satisfying the given constraints

Efficiency of AutoGM (2)

Dataset	Search(s)	Min.Acc.	Fastest Inference (s)		Accuracy		Max.Time(s)	Highest Accuracy		Inference (s)	
			AutoGM	Random	AutoGM	Random		AutoGM	Random	AutoGM	Random
Cora	450	0.78	**0.0034**	-	0.79	-	0.004	**0.77**	**0.77**	0.0036	0.0033
Citeseer	800	0.67	**0.0039**	**0.0039**	0.67	0.67	0.004	**0.67**	-	0.0039	-
Pubmed	1,800	0.75	**0.021**	-	0.77	-	0.004	**0.76**	0.71	0.0036	0.0039
AmazonC	5,700	0.85	**0.032**	0.033	0.89	0.87	0.04	**0.85**	-	0.032	-
AmazonP	18,000	0.93	**0.047**	0.065	0.94	0.93	0.05	**0.94**	-	0.048	-
CoauthorC	2,500	0.8	**0.015**	0.016	0.8	0.82	0.02	**0.83**	0.75	0.015	0.02
CoauthorP	1,500	0.9	**0.01**	-	0.91	-	0.01	**0.92**	0.86	0.01	0.01

- Performance is still lower than the algorithms found by AutoGM

Efficiency of AutoGM (3)

Dataset	Search(s)	Min.Acc.	Fastest Inference (s) AutoGM	Fastest Inference (s) Random	Accuracy AutoGM	Accuracy Random	Max.Time(s)	Highest Accuracy AutoGM	Highest Accuracy Random	Inference (s) AutoGM	Inference (s) Random
Cora	450	0.78	**0.0034**	-	0.79	-	0.004	**0.77**	**0.77**	0.0036	0.0033
Citeseer	800	0.67	**0.0039**	**0.0039**	0.67	0.67	0.004	**0.67**	-	0.0039	-
Pubmed	1,800	0.75	**0.021**	-	0.77	-	0.004	**0.76**	0.71	0.0036	0.0039
AmazonC	5,700	0.85	**0.032**	0.033	0.89	0.87	0.04	**0.85**	-	0.032	-
AmazonP	18,000	0.93	**0.047**	0.065	0.94	0.93	0.05	**0.94**	-	0.048	-
CoauthorC	2,500	0.8	**0.015**	0.016	0.8	0.82	0.02	**0.83**	0.75	0.015	0.02
CoauthorP	1,500	0.9	**0.01**	-	0.91	-	0.01	**0.92**	0.86	0.01	0.01

- By exhausting the budget, AutoGM brings the best trade-off

Overview

1. **Motivation**
2. **UnifiedGM:** Unified Graph Mining Framework
3. **AutoGM:** Automation of Graph Mining Algorithm Development
4. **Experiments**
5. **Conclusion**

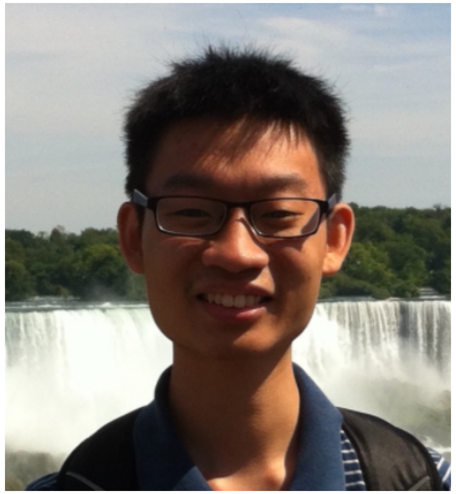
Conclusion

- Unification of Graph Mining algorithms
- Automation of Graph Mining algorithm developments
- Budget awareness

Conclusion

- Unification of Graph Mining algorithms
- Automation of Graph Mining algorithm developments
- Budget awareness

Empower practitioners without much expertise to deploy Graph Mining algorithms



Minji Yoon
Carnegie Mellon University

Théophile Gervet
Carnegie Mellon University

Bryan Hooi
National University of Singapore

Christos Faloutsos
Carnegie Mellon University

Thank you

Paper: https://minjiyoon.xyz

Github: https://github.com/minjiyoon/ICDM20-AutoGM