Deal or Deceit: Detecting Cheating in Distribution Channels

Kai Shu[1,2], Ping Luo[1]

Li Wan[2], Peifeng Yin[3], Linpeng Tang[4]

1 中国科学院计算技术研究所 Institute of Computing Technology,Chinese Academy of Sciences

2 CHONGQING UNIVERSITY

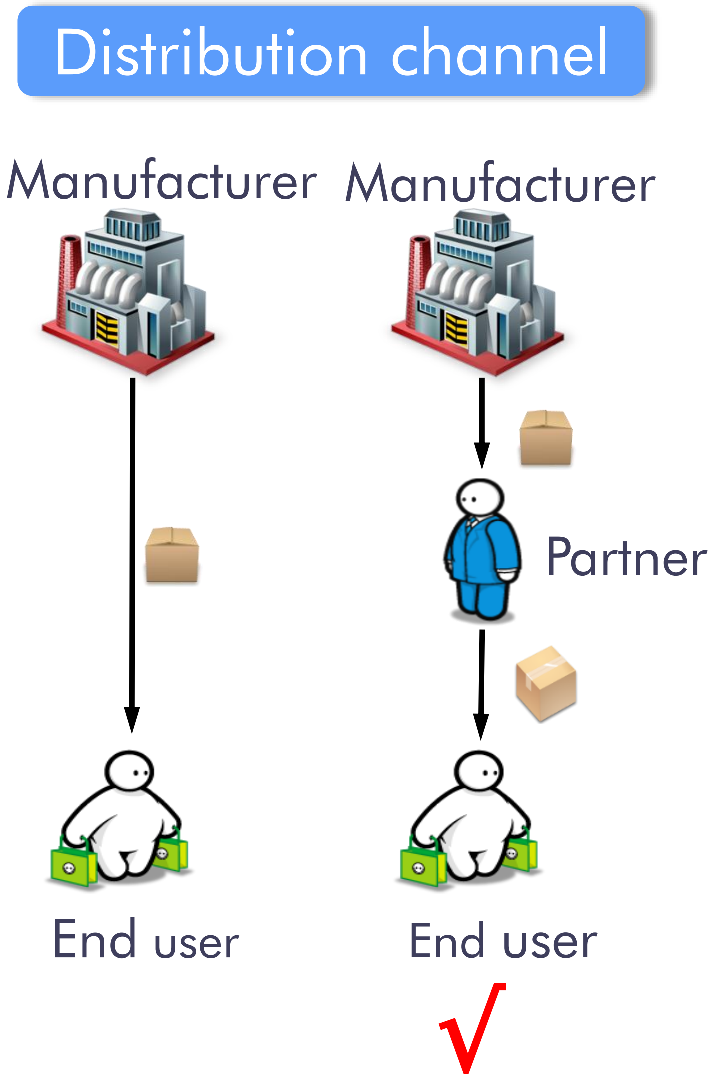
3 PENNSTATE

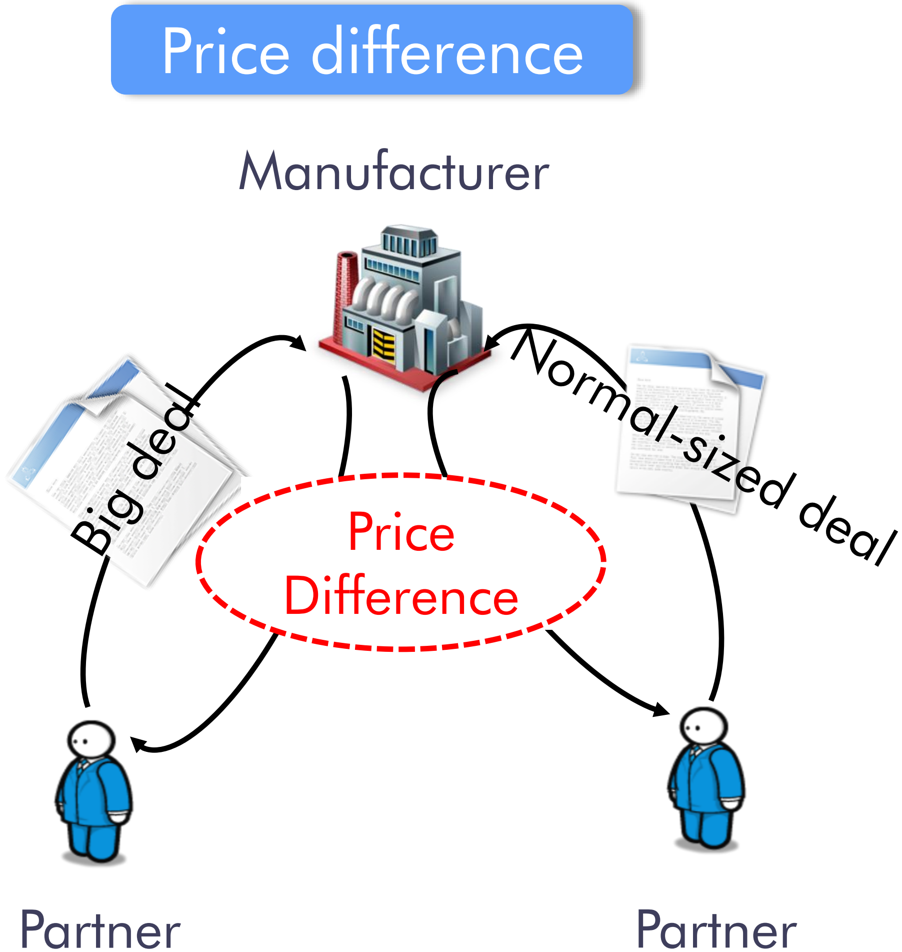
4 PRINCETON UNIVERSITY

Outline

- ➢ Introduction
- Motivation
- Problem Formulation
- Detection Framework
- Experimental results
- Conclusion

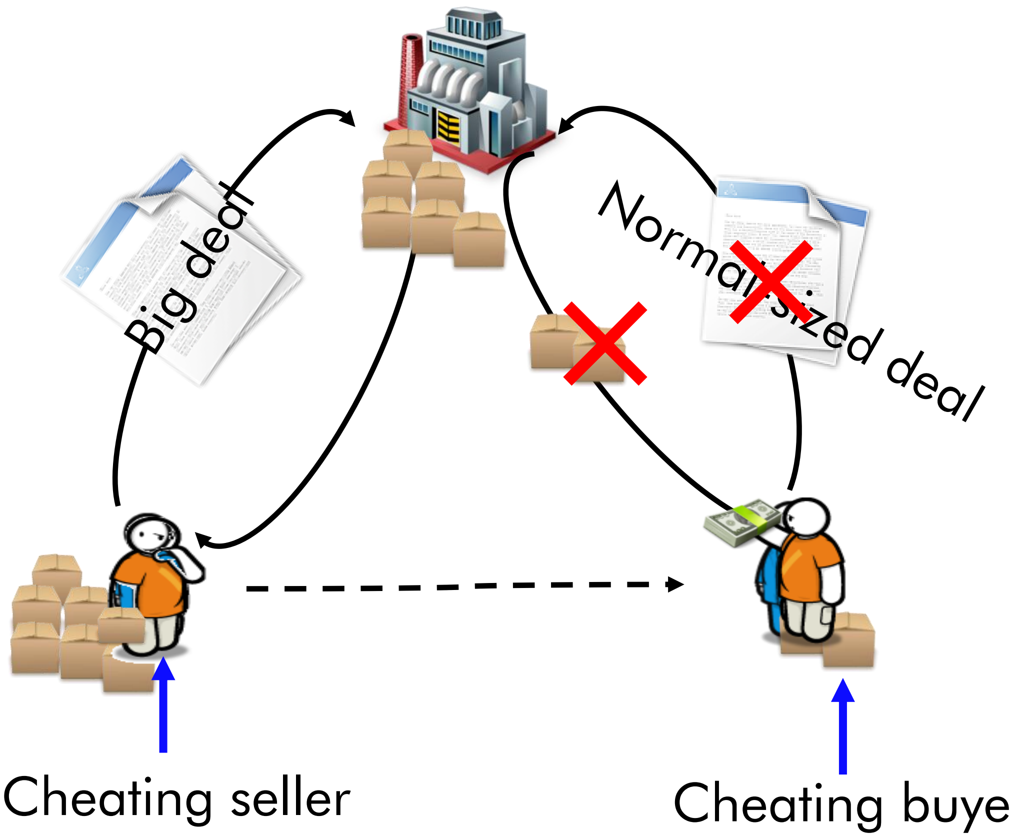
Introduction





Introduction

A typical cheating scenario



Detect cheating in distribution channel

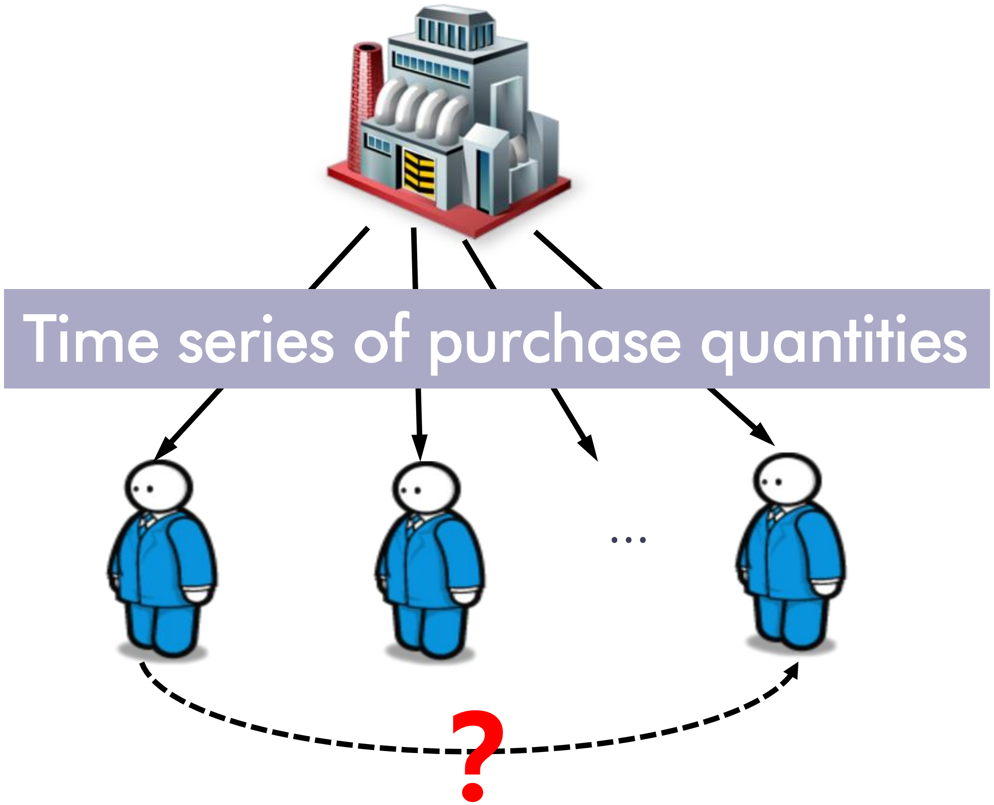
Outline

- Introduction
- ➢ Motivation
- Problem Formulation
- Detection Framework
- Experimental results
- Conclusion

Motivation

Data: from the perspective of manufactory

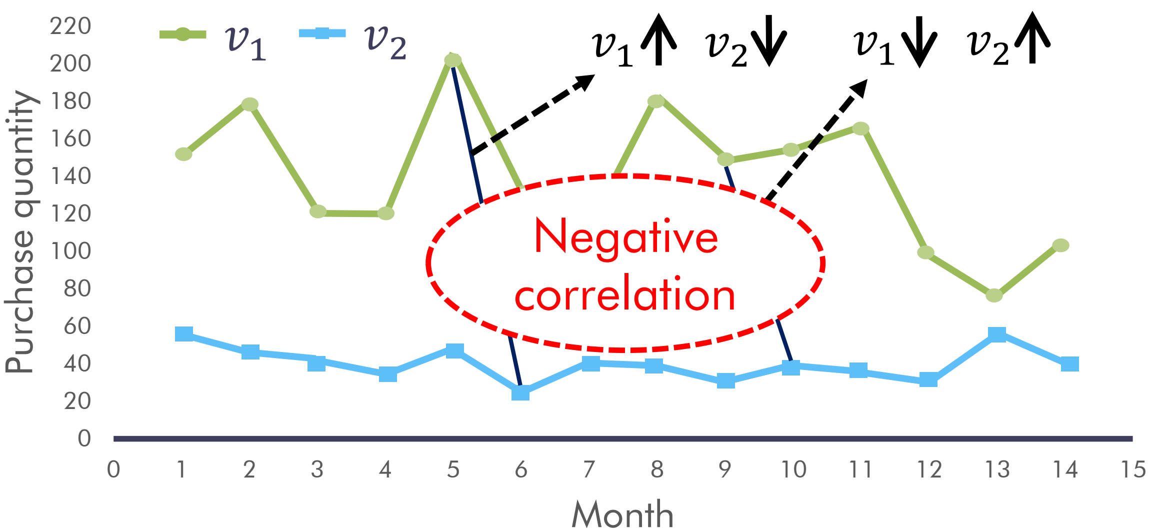
Time series of purchase quantities for each partner

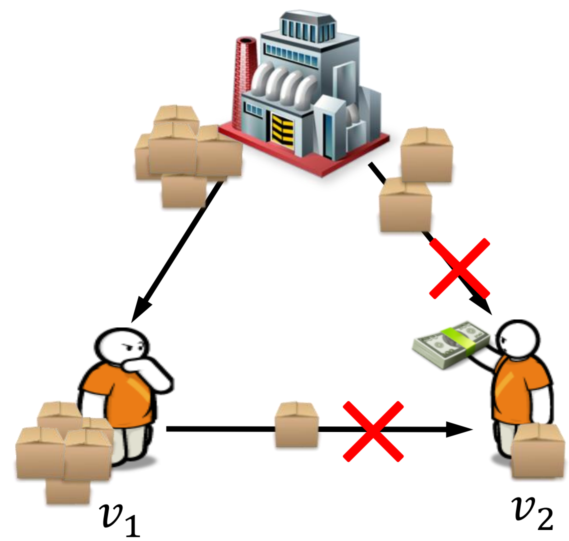


Motivation (2)

Observation

The purchase quantities of v_1 and v_2 change collectively

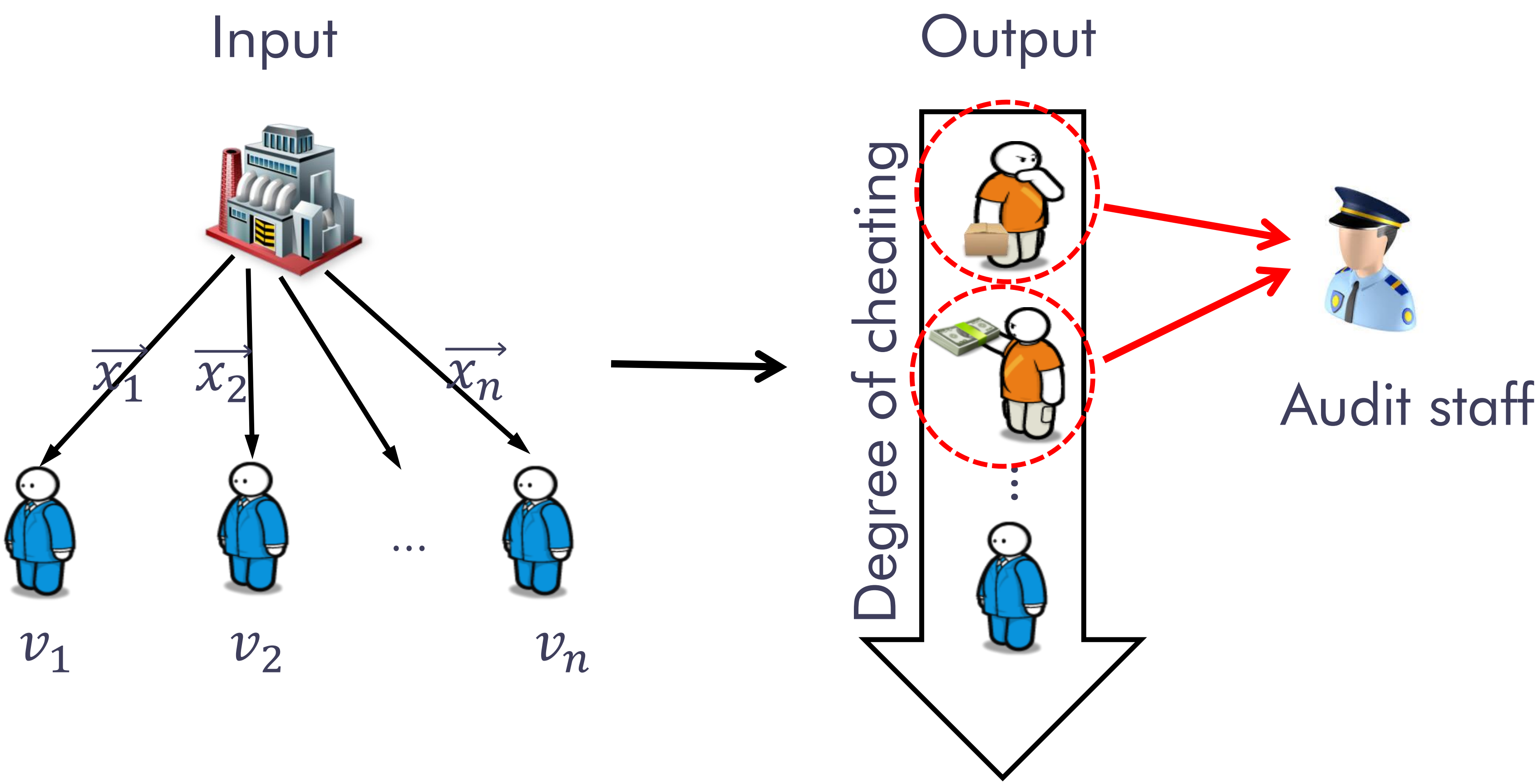




Outline

- Introduction
- Motivation
- ➢ Problem Formulation
- Detection Framework
- Experimental results
- Conclusion

Problem Formulation



Outline

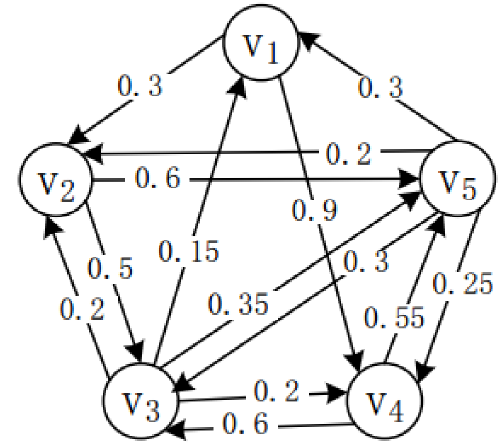
- Introduction
- Motivation
- Problem Formulation
- ➢ Detection Framework
- Experimental results
- Conclusion

Detection Framework

Step 1

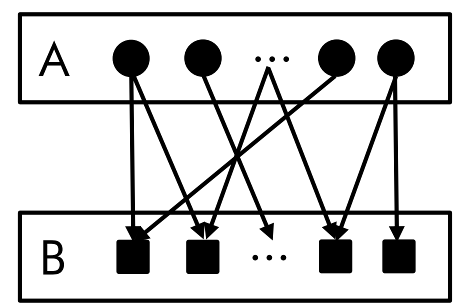
Building Partner Correlation Graph (PCG)

all purchase volume sequences $(\vec{x_i} | i = 1,2, \dots, n)$



Step 2

Partitioning



Step 3

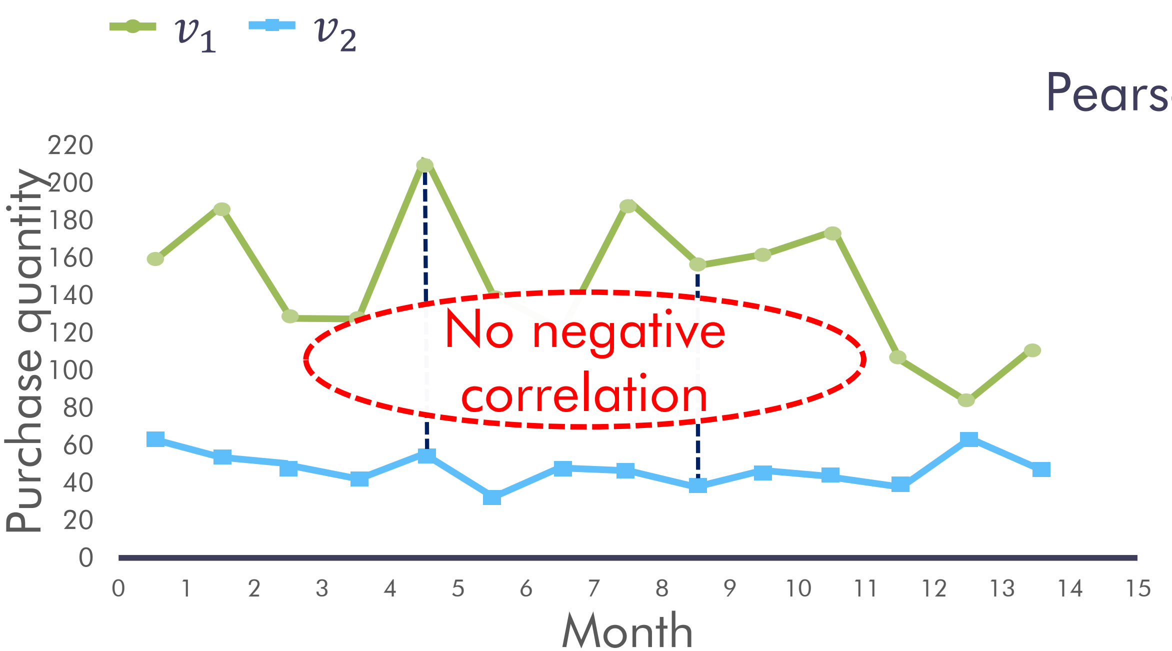
Ranking the partners by probabilistic model

Ranking of cheating partners

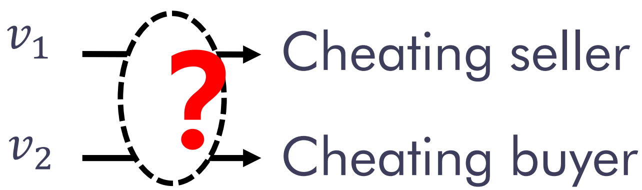
Detection Framework (Step 1)

Pearson correlation

- Tick-to-tick correspondence
- Symmetric measure



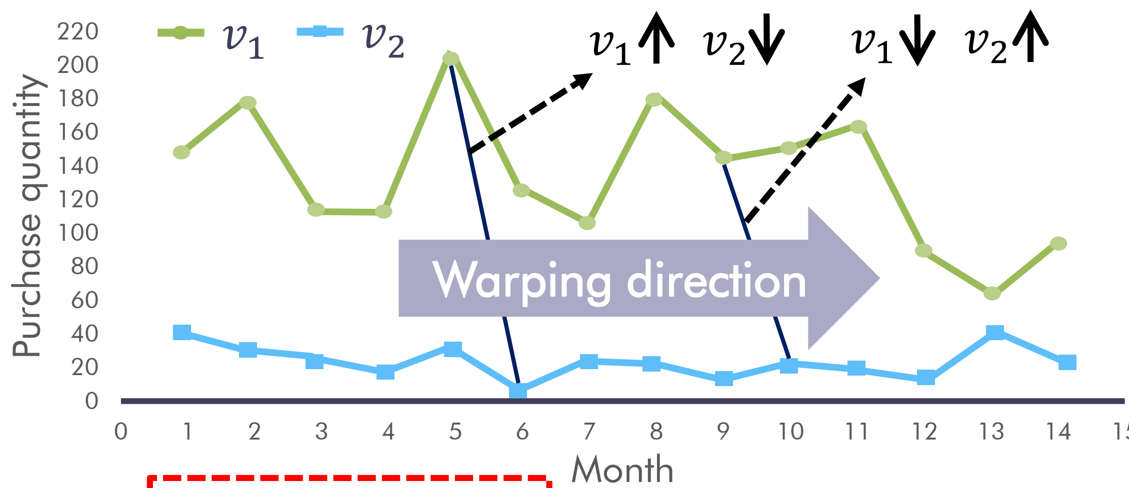
$$\text{Pearson}(v_1, v_2) = \text{Pearson}(v_2, v_1)$$



Detection Framework (Step 1)

Dynamic time warping

Learn the correspondence relationship of elements between two time series to minimize the Pearson correlation.



Cheating buyer

Directed Pearson Correlation (DPC)

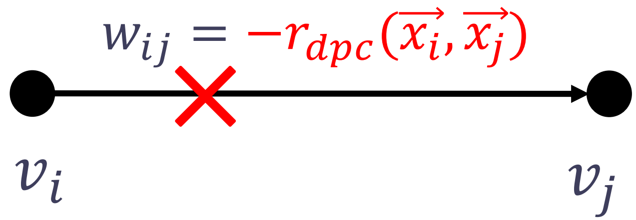
$r_{dpc}(\overrightarrow{x_1},\overrightarrow{x_2}) \neq r_{dpc}(\overrightarrow{x_2},\overrightarrow{x_1})$

Cheating seller

$$r_{dpc}(\overrightarrow{x_1},\overrightarrow{x_2}) := \min\{\frac{1}{L}\sum_{l=1}^{L}(\frac{\overrightarrow{x_1}(i_l)-\overline{x_1}}{\sigma_{\overrightarrow{x_1}}})(\frac{\overrightarrow{x_2}(j_l)-\overline{x_2}}{\sigma_{\overrightarrow{x_2}}})\}$$

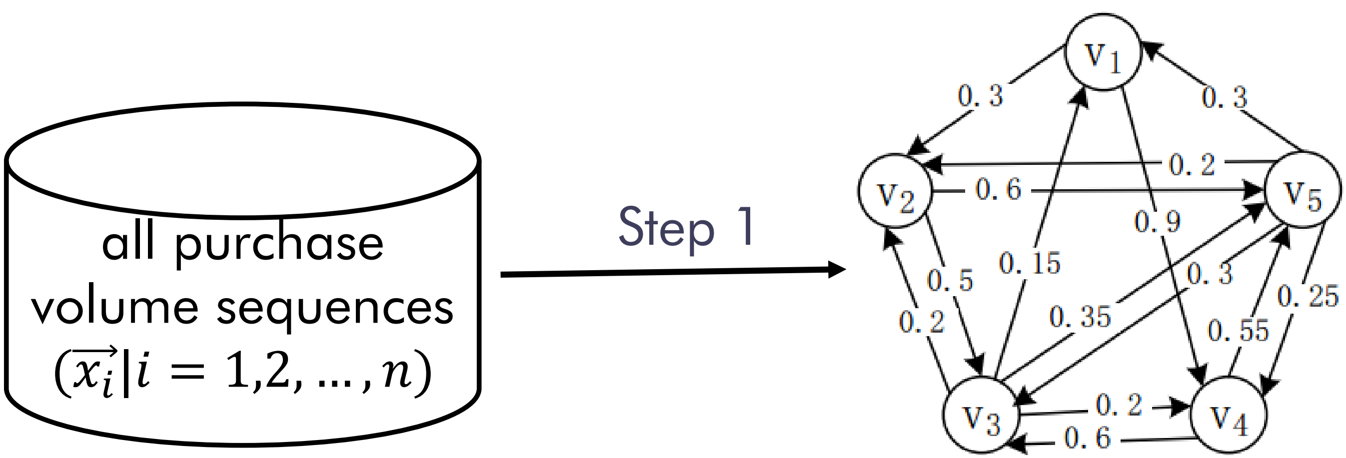
Where (i_l, j_l) is the element of a warping path P and $L = |P|$.

Detection Framework (Step 1)



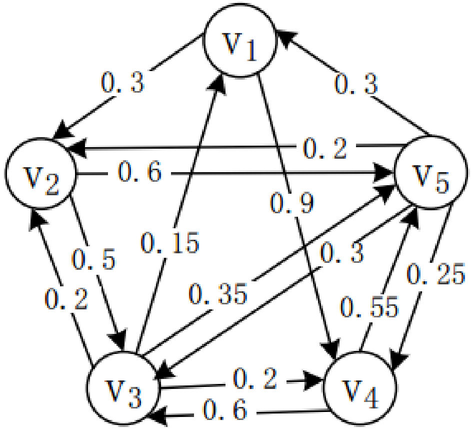
If $w_{ij} \geq \eta$, keep the edge

If $w_{ij} < \eta$, remove the edge

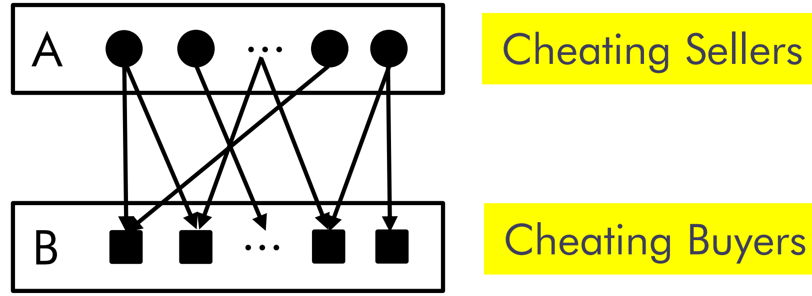


PCG

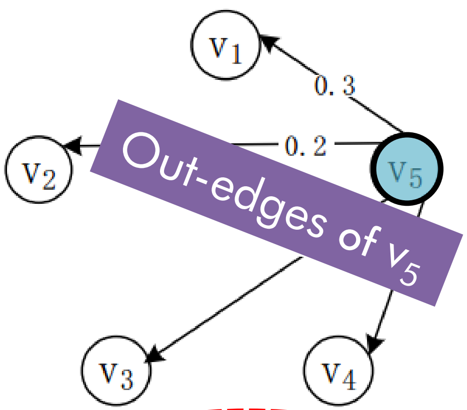
Detection Framework (Step 2)

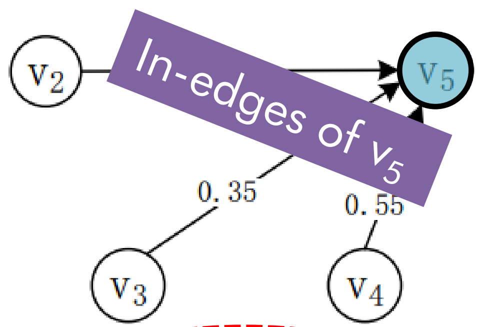


PCG



Bipartite Graph





Cheating Seller and Cheating Buyer

Not true

A partner can only be cheating seller or cheating buyer

Detection Framework (Step 2)

Classical graph cut problem (MAX DICUT)

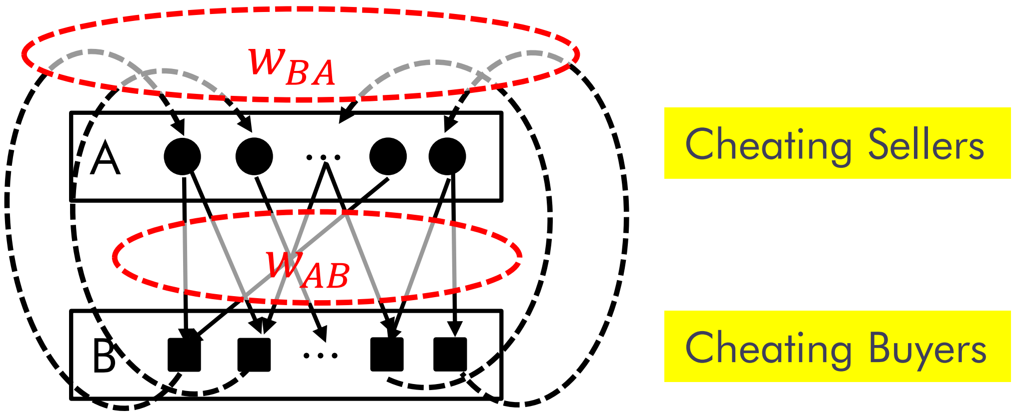
$$\text{Maximize } w_{AB}$$

New graph cut problem

$$\text{Maximize } 3w_{AB} - w_{BA}$$

Solution algorithm

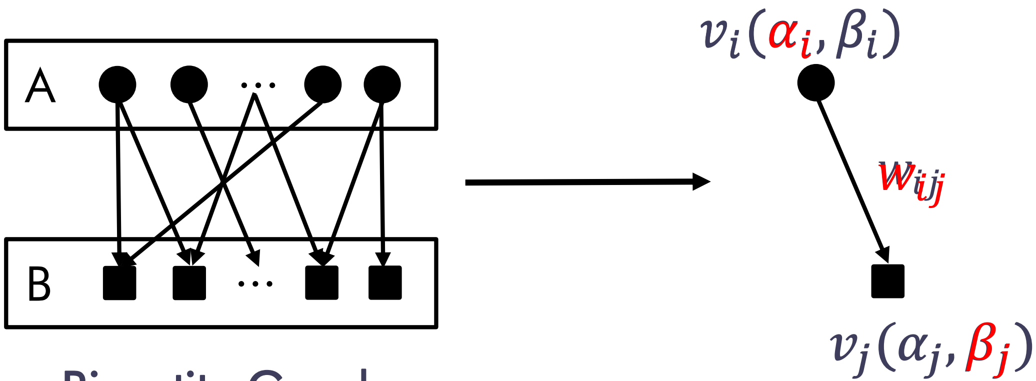
We propose a greedy algorithm to solve this new NP-hard problem.



Detection Framework (Step 3)

Probabilistic model

Generate the edges and the weights in resultant bipartite graph.



Bipartite Graph

$$Beta(w_{ij}; \alpha_i, \beta_j)$$

$$\alpha_i \quad \beta_j \rightarrow w_{ij}$$

$$\alpha_i \quad \beta_j \rightarrow w_{ij}$$

Probabilistic model

$$\text{Maxmize } \log p(w_{ij}, \boldsymbol{\theta}) + \lambda \cdot C(\boldsymbol{\theta})$$

$\boldsymbol{\theta}$ is the set of all parameters

Detection Framework (Step 3)

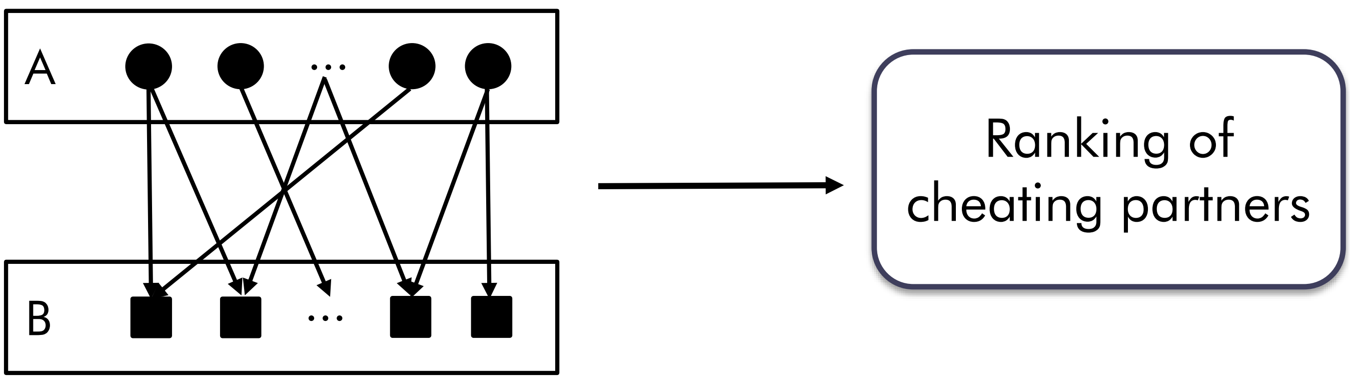
Probability weight π_{ij}

$$\pi_{ij} = \frac{\alpha_i}{\alpha_i + \beta_j}$$

π_{ij} is the expected value of beta distribution with parameters (α_i, β_j)

Ranking score function

$$score(i) = \begin{cases} \pi_{iB} - \pi_{*i}, & \text{if } v_i \in A \\ \pi_{Ai} - \pi_{i*}, & \text{if } v_i \in B \end{cases}$$



Bipartite Graph

Outline

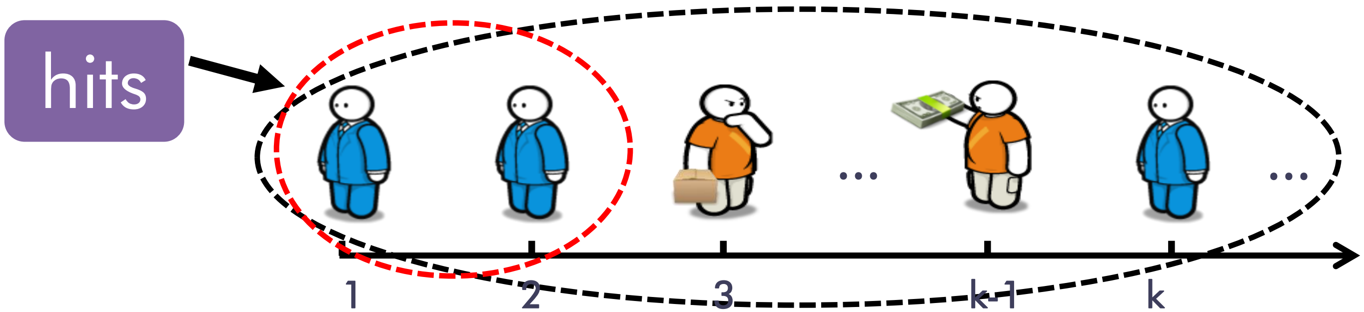
- Introduction
- Motivation
- Problem Formulation
- Detection Framework
- ➢ Experimental results
- Conclusion

Experimental results

Dataset

	Gold	Silver	All
# Total partners	104	424	528
# Cheating partners	17	85	102

Ground truth



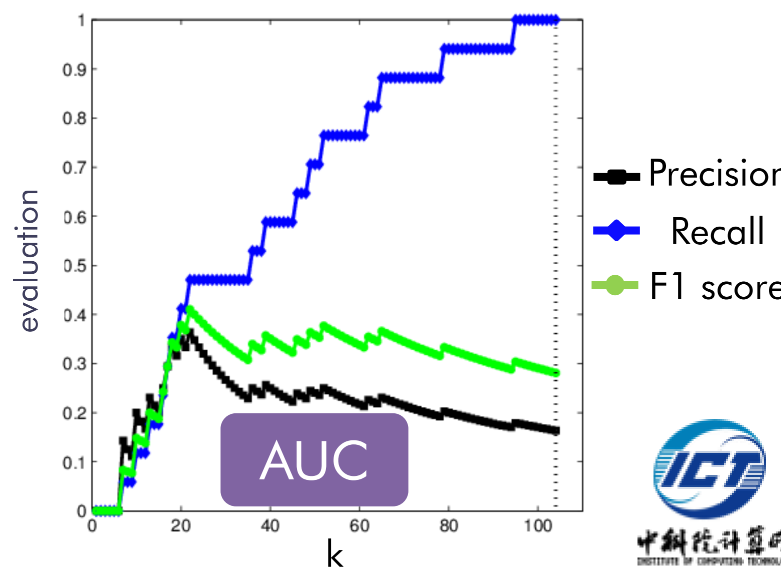
Evaluation measure

For the top-k ranking list, giving a number k,we count the number of hits in it.

$$Precision@k = \frac{\#\text{hits in top}-\text{k list}}{k}$$

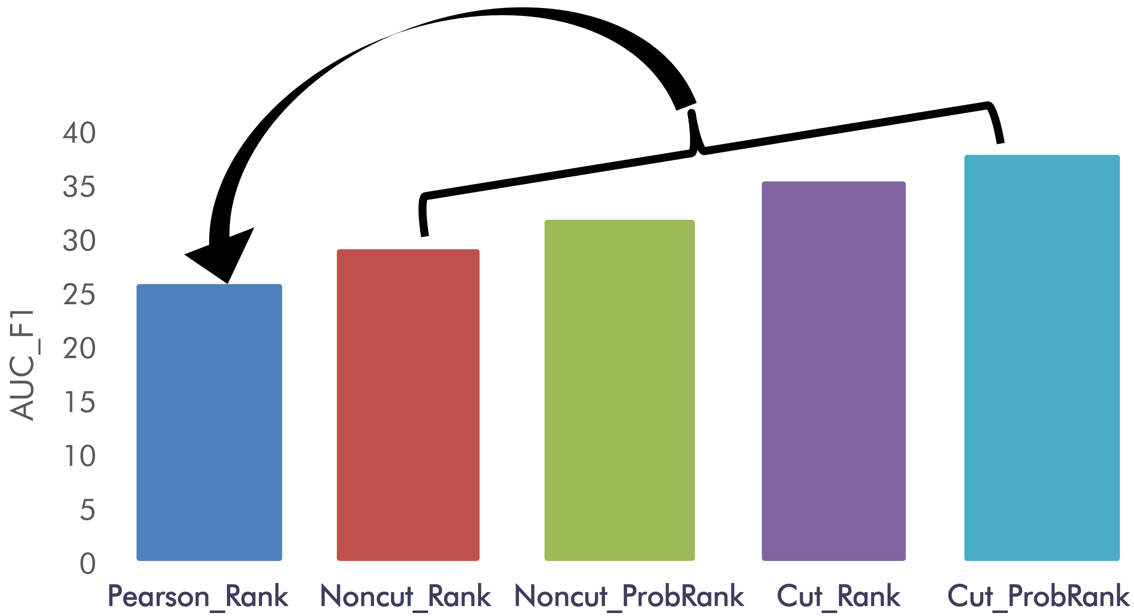
$$Recall@k = \frac{\#\text{hits in top}-\text{k list}}{M}$$

$$F1@k = 2 \cdot \frac{Precision@k \cdot Recall@k}{Precision@k + Recall@k}$$



Experimental results (1)

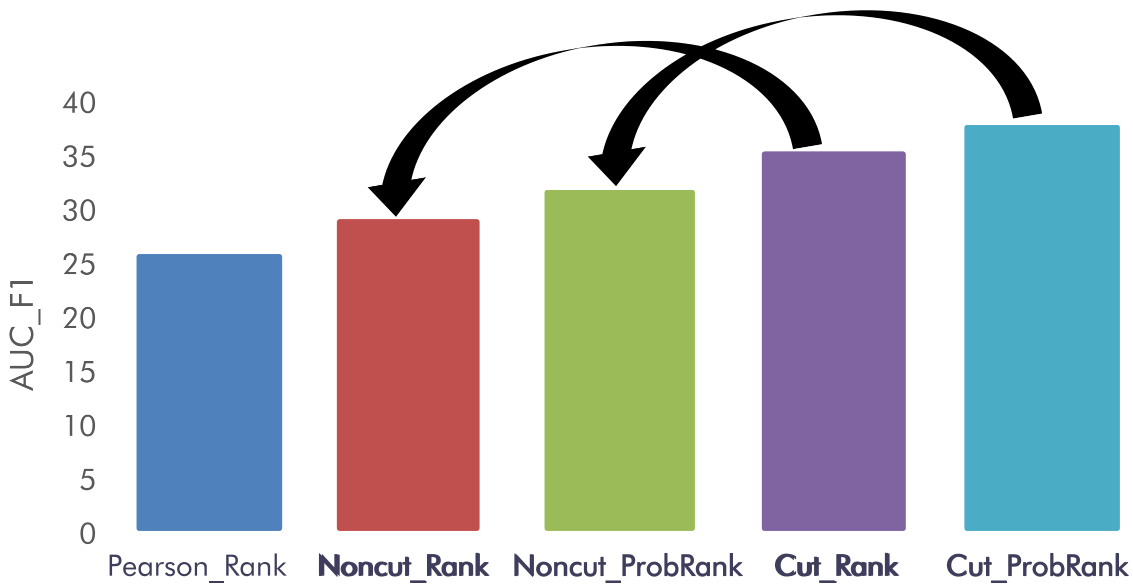
	PCG		Partition		Ranking	
	Pearson correlation	DPC	No	Yes	Edge weight	Probability
Noncut_Rank		√	√		√	
Noncut_ProbRank		√	√			√
Cut_Rank		√		√	√	
Cut_ProbRank		√		√		√
Pearson_Rank	√		√		√	



- DPC improves performance

Experimental results (2)

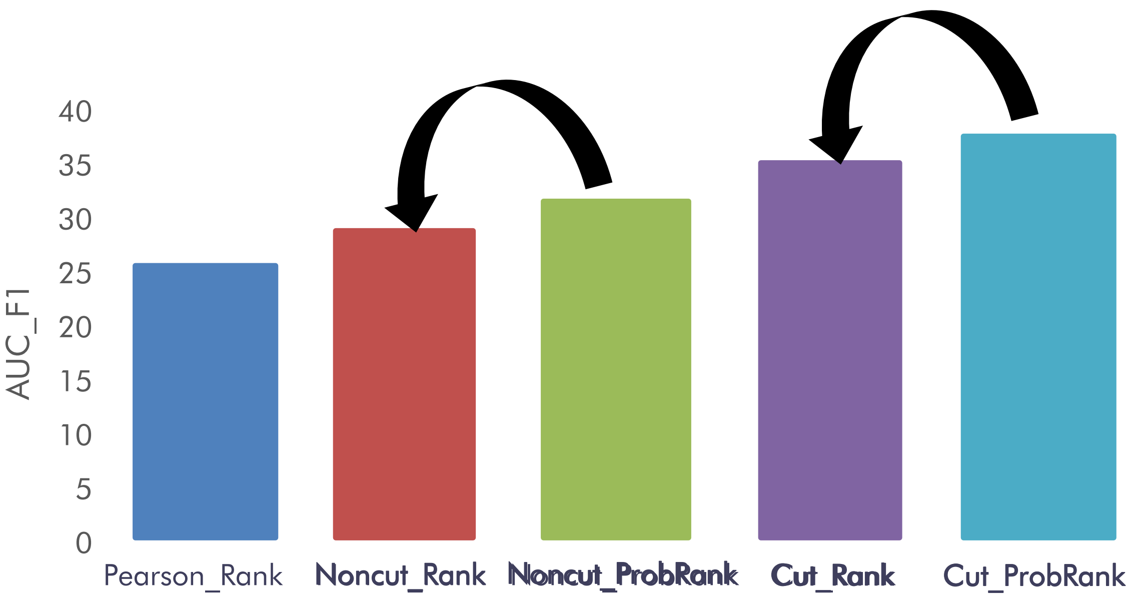
	PCG		Partition		Ranking	
	Pearson correlation	DPC	No	Yes	Edge weight	Probability
Noncut_Rank		√	√		√	
Noncut_ProbRank		√	√			√
Cut_Rank		√		√	√	
Cut_ProbRank		√		√		√
Pearson_Rank	√		√		√	



- DPC improves performance
- Partitioning improves performance

Experimental results (3)

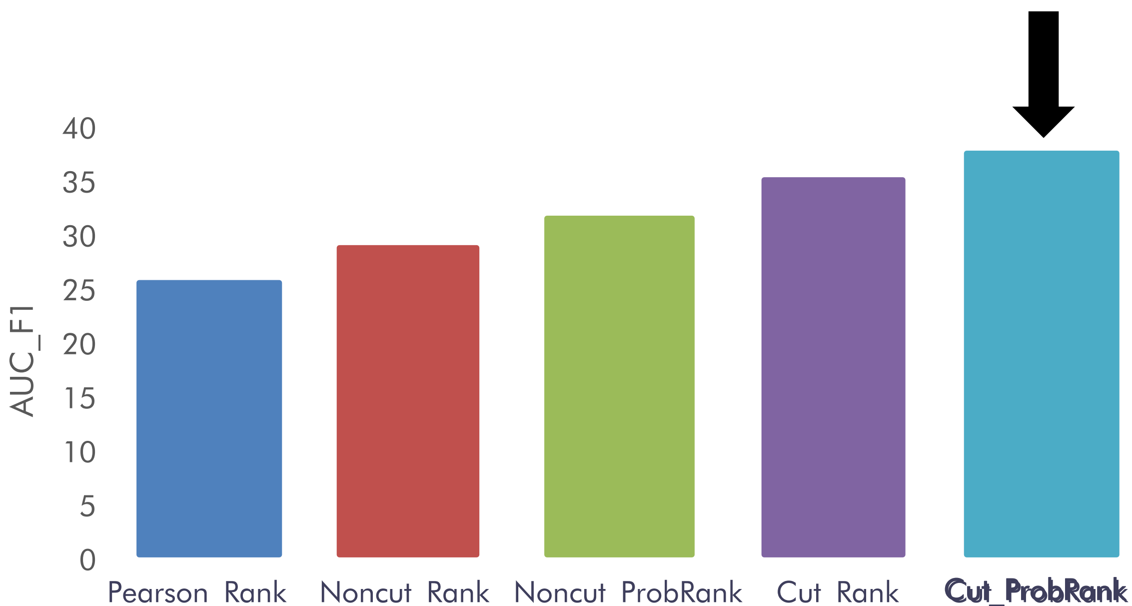
	PCG		Partition		Ranking	
	Pearson correlation	DPC	No	Yes	Edge weight	Probability
Noncut_Rank		√	√		√	
Noncut_ProbRank		√	√			√
Cut_Rank		√		√	√	
Cut_ProbRank		√		√		√
Pearson_Rank	√		√		√	



- DPC improves performance
- Partitioning improves performance
- Probability ranking improves performance

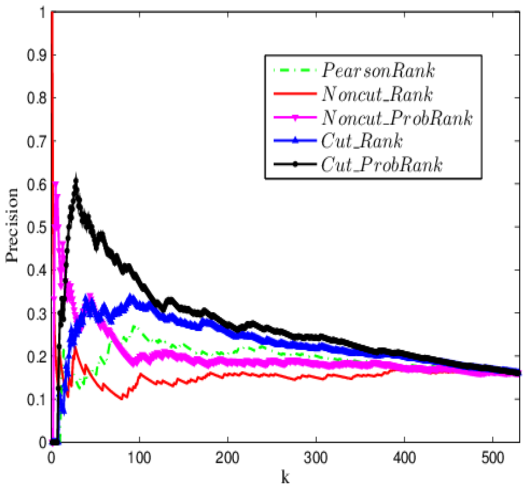
Experimental results (4)

	PCG		Partition		Ranking	
	Pearson correlation	DPC	No	Yes	Edge weight	Probability
Noncut_Rank		√	√		√	
Noncut_ProbRank		√	√			√
Cut_Rank		√		√	√	
Cut_ProbRank		√		√		√
Pearson_Rank	√		√		√	

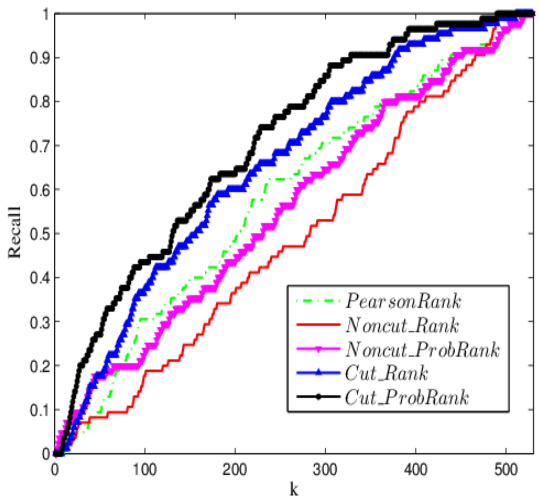


- DPC improves performance
- Partitioning improves performance
- Probability ranking improves performance
- Cut_ProbRank performs best

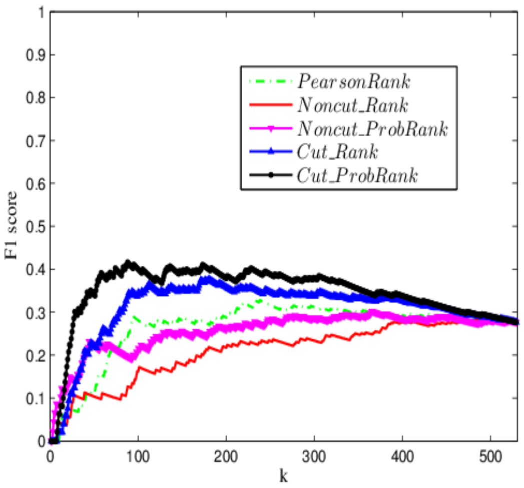
Experimental results (5)



Precision



Recall



F1 score

Cut_ProbRank always performs the best

Outline

- Introduction
- Motivation
- Problem Formulation
- Detection Framework
- Experimental results
- ➢ Conclusion

Conclusion

The first quantitative work to detect cheating in distribution channels.

- A new correlation method (i.e. DPC) by Introducing dynamic time warping technique.
- A new graph cut problem for graph partitioning.
- A probability model for node ranking.

Thanks