Privacy Preservation in Social Networks with Sensitive Edge Weights

Lian Liu (speaker), Jie Wang, Jinze Liu, and Jun Zhang

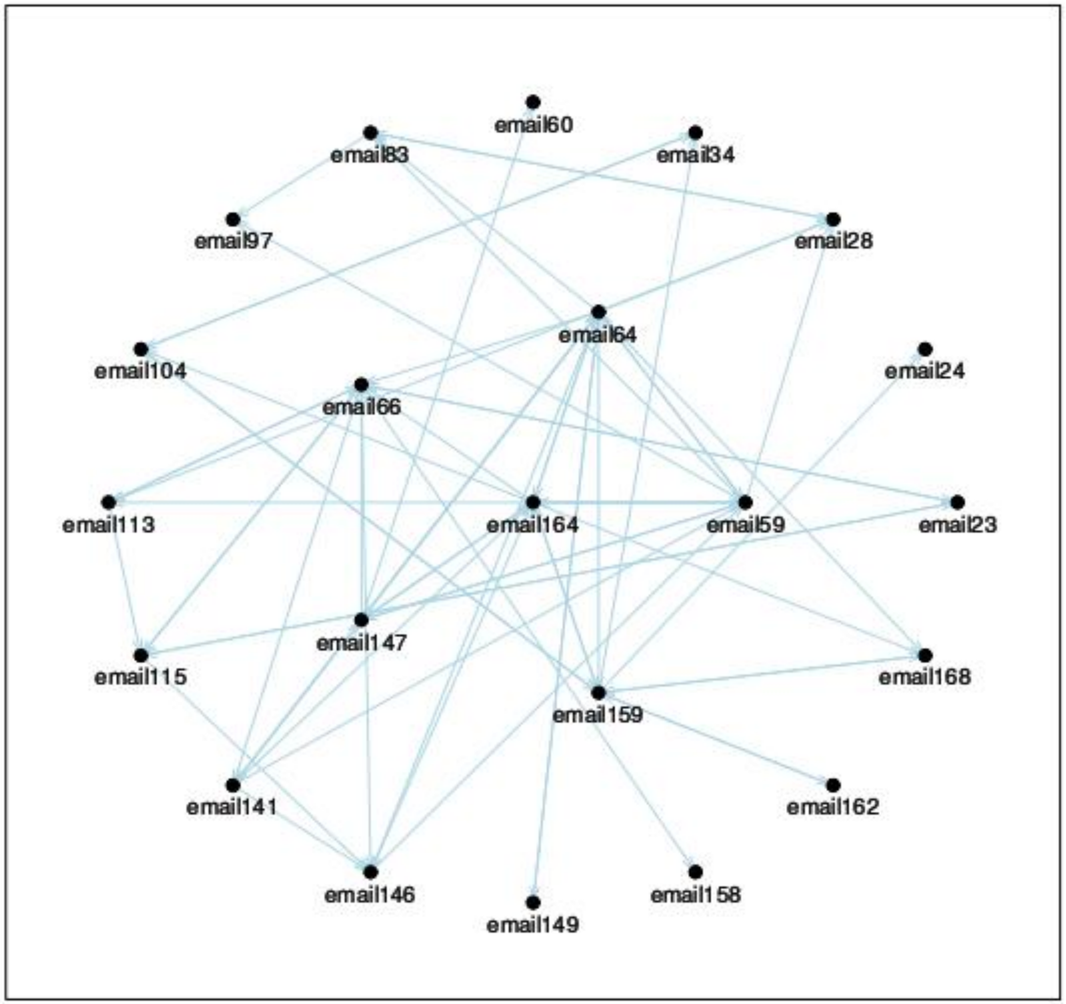
Computer Science Department, University of Kentucky,

May 1, 2009

Outline

- Privacy-Preserving Social Networks
- Gaussian Perturbation
- Greedy Perturbation
- Experimental Results
- Conclusion

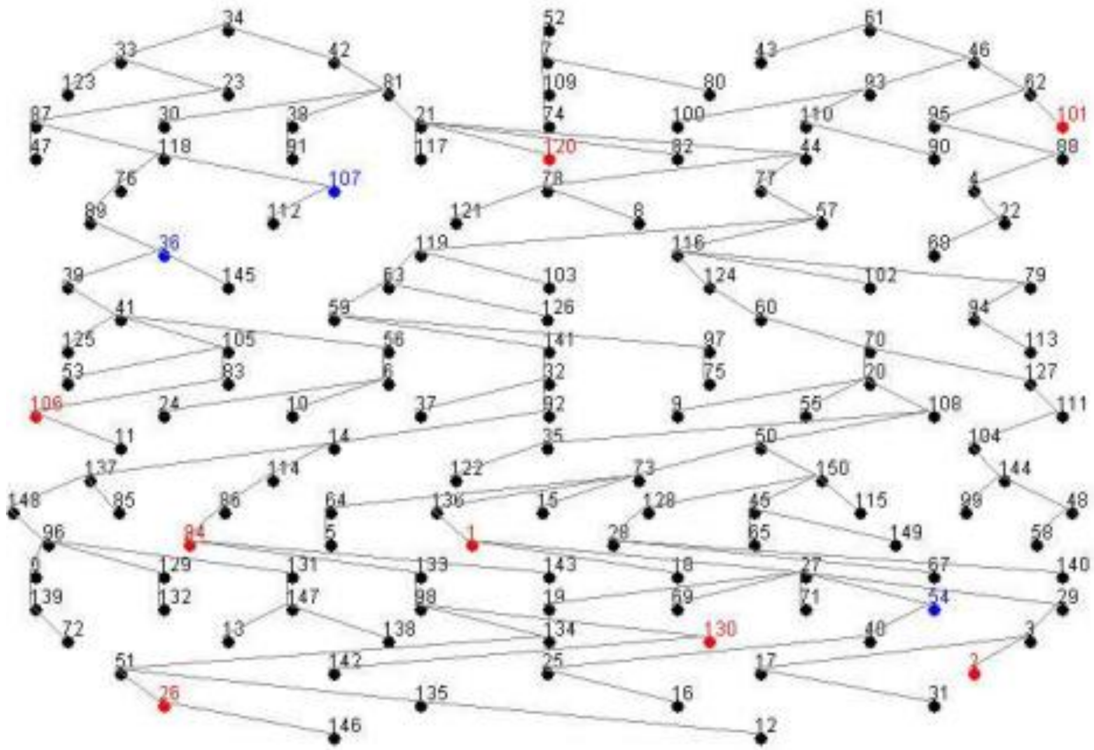
Unweighted social networks



A part social network of Enron email communication.

Source: Priebe et. al. *Scan Statistics on Enron Graphs.*

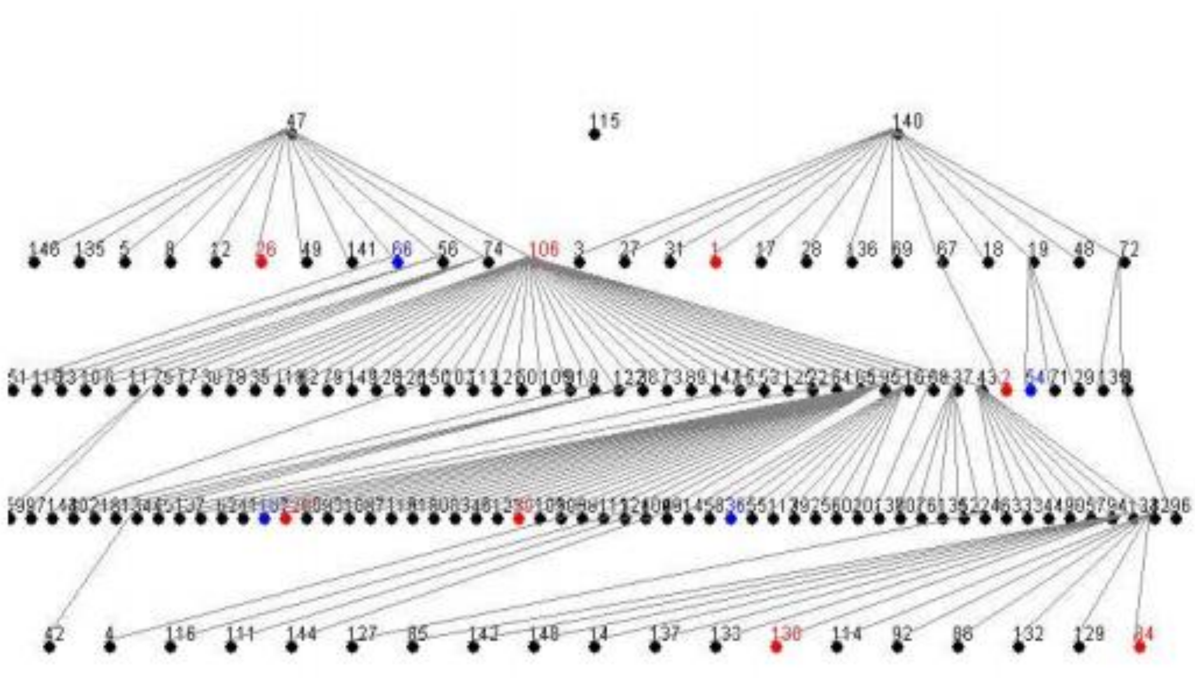
Weighted Social Networks



Source: Zhou et. al. *Towards Discovering Organizational Structure from Email Corpus.*

Weighted Social Networks

After adding weights to the social network, a new data pattern appears, such as leadership as follows.



Source: Zhou et. al. *Towards Discovering Organizational Structure from Email Corpus.*

Data Privacy and Data Utility

- Data Privacy

  The individual edge weights (essentially a local information)

Data Privacy and Data Utility

- Data Privacy

  The individual edge weights (essentially a local information)

- Data Utility

  The shortest path, i.e., a path with a minimum sum of weights (essentially a global property)

Goals

- **Our goal:**

  Preserving privacy while maintaining data utility. In this paper,

  - perturb edge weights as much as possible,
  - keep shortest paths (and lengths) approximate to the original ones as much as possible.

Challenges

Theorem: There does NOT exist one *perfect* scheme such that it can modify all weights but at the same time keep all shortest paths (and lengths). *

* Formal proposition and mathematic proof are referred to Proposition 1 in our paper.

Anonymization method:

No edge or node deletion/insertion

--> edge weight perturbation

$w_{ij} \rightarrow w^*_{ij}$

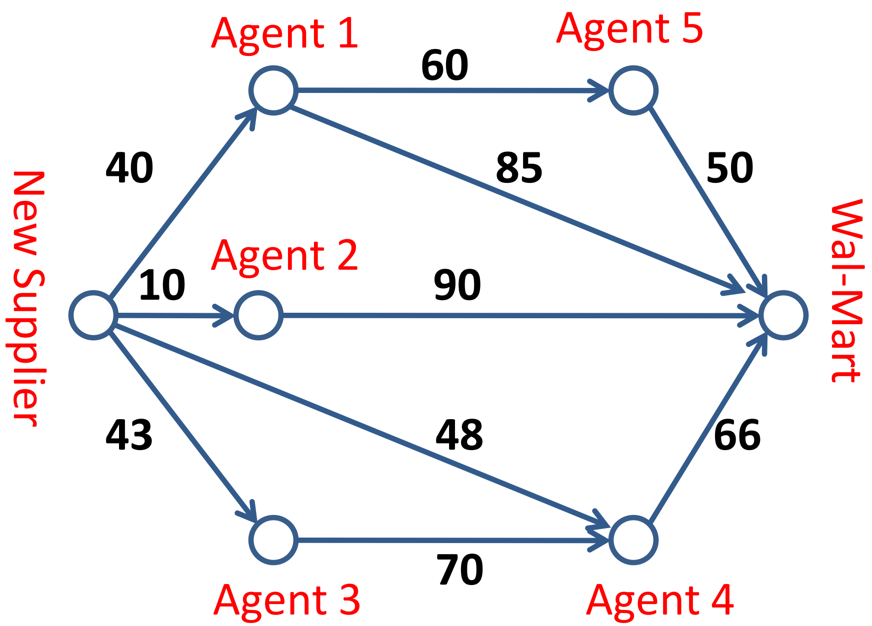
Two (utility) metrics:

- Keep the same shortest path
- Preserve the lengths of the perturbed shortest path within some bounds of the original

Outline

- Privacy-Preserving Social Networks
- Gaussian Perturbation
- Greedy Perturbation
- Experimental Results
- Conclusion

Gaussian Perturbation

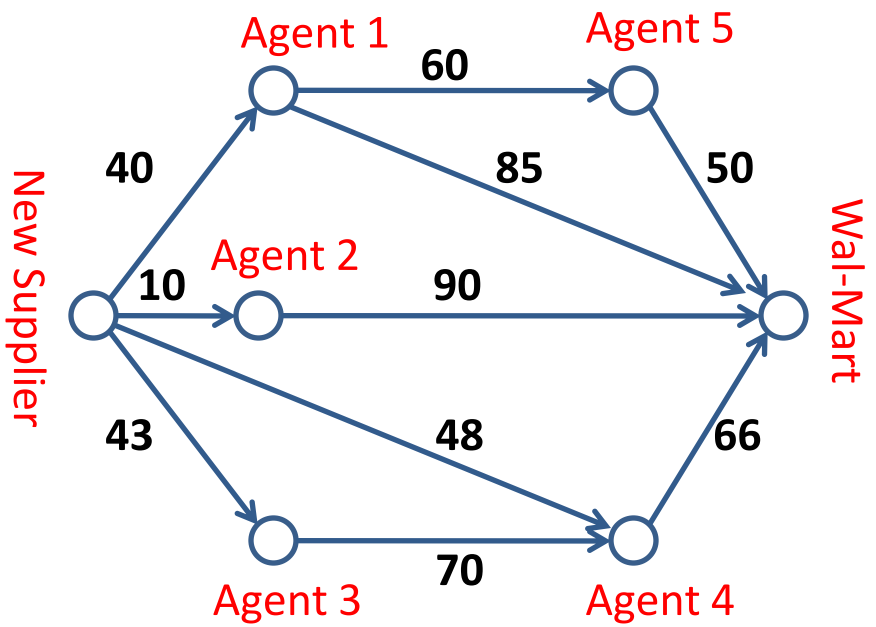


Unit=million/month

Gaussian Perturbation

$$w^*_{i,j} = w_{i,j}\,(1-x_{i,j}),$$

Here $x_{i,j}$ is a randomly generated number from the Gaussian distribution $N(0,\sigma^2)$.

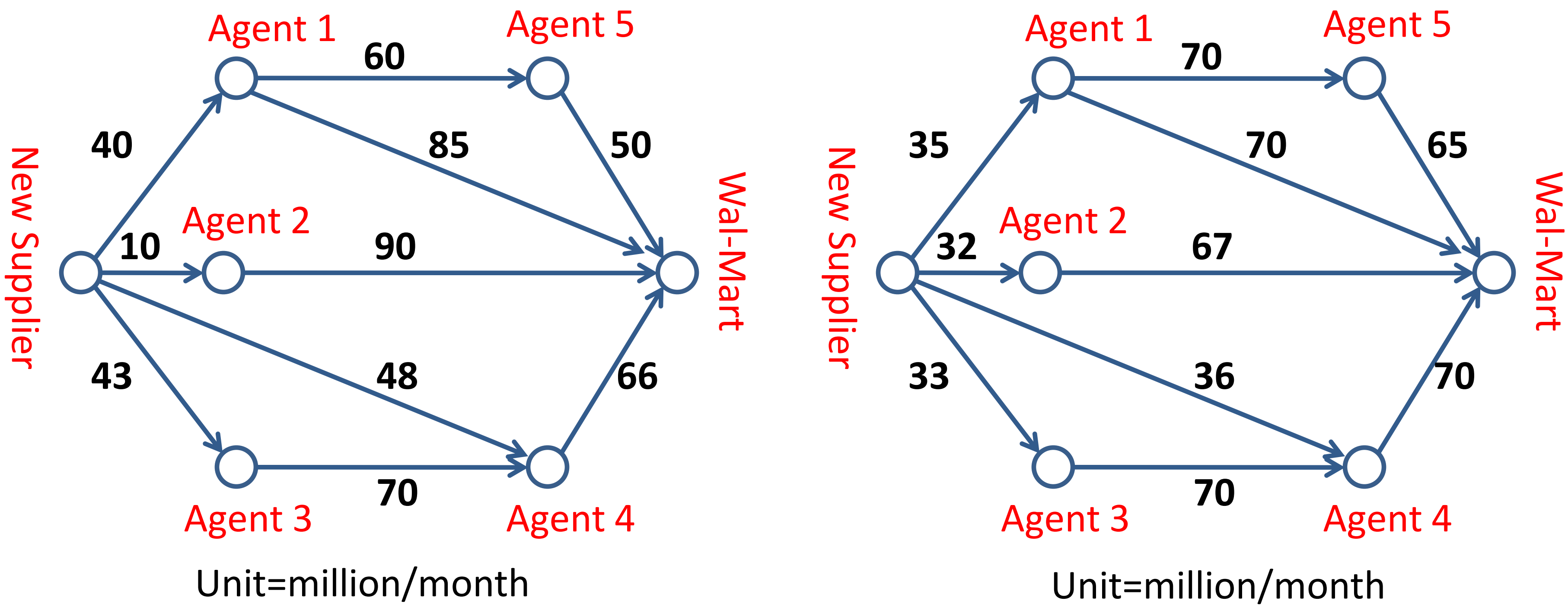


Unit=million/month

Gaussian Perturbation

$$w^*_{i,j} = w_{i,j}\,(1 - x_{i,j}),$$

Here $x_{i,j}$ is a randomly generated number from the Gaussian distribution $N(0,\sigma^2)$.



- Privacy: almost all weights are changed.
- Utility: Same shortest path between New Supplier and Walmart and length is 99.

Analysis on Gaussian perturbation

Let the length of a path be L in original networks and L^* be the length of the corresponding path in perturbed networks.

1. approximately 68% L satisfy $\left|\frac{L-L^*}{L}\right|, \leq \sigma$
2. Approximately 98% L satisfy $\left|\frac{L-L^*}{L}\right| \leq 2\sigma$
3. approximately 99.7% L satisfy $\left|\frac{L-L^*}{L}\right| \leq 3\sigma$

* Formal theorem/corollary and mathematic proofs are referred to Theorem 2 and Corollary 3 in our paper, respectively.

Analysis on Gaussian perturbation

Let $d_{i,j}$ be the length of the shortest path between node *i* and node *j*, and $d_{i,j}^{second}$ be the length of the second shortest path between same node pair.

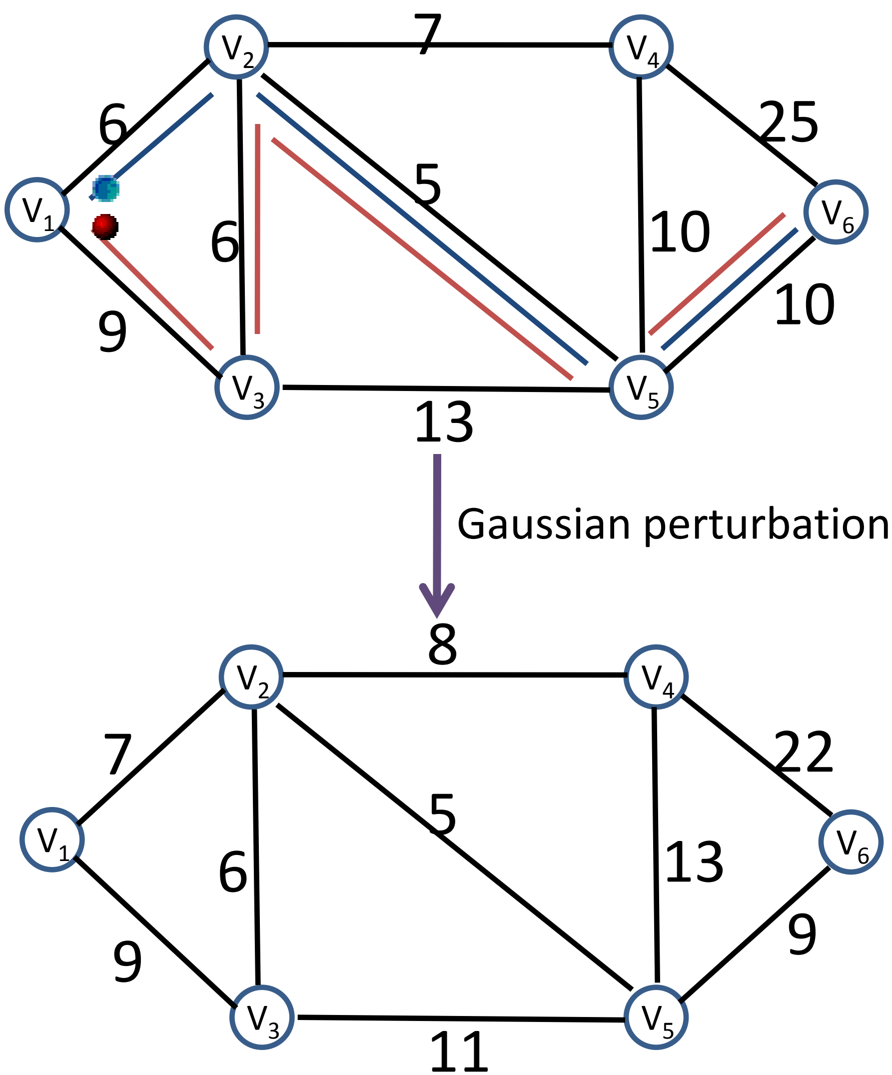
We define $\beta_{i,j} = \dfrac{d_{i,j}^{second} - d_{i,j}}{d_{i,j}}$.

If $\beta_{i,j} >= 2\sigma$, the shortest path is highly possible to be preserved after Gaussian perturbation. *

Recall approximately 98% L satisfy $\left|\frac{L - L^*}{L}\right| \leq 2\sigma$.

* Formal theorem/corollary and mathematic proofs are referred to Theorem 2 and Corollary 3 in our paper, respectively.

An example



The shortest path, length is 21

The second shortest path, length is 30

$\boldsymbol{\beta}_{1,6}$ = (30-21)/21 = 0.429 >= 2$\boldsymbol{\sigma}$. So the shortest path between $\boldsymbol{v_1}$ and $\boldsymbol{v_6}$ can be maintained no matter how you choose the random value from Gaussian distribution.

$\boldsymbol{\sigma}$ = 0.15

Gaussian Perturbation

- Gaussian Perturbation is quick and independent with global structure. But It cannot always keep the same shortest paths when perturbation get larger (i.e., σ is large).

- So we propose alternative Greedy Perturbation which can keep the exact shortest paths, and make sure that their corresponding lengths are similar to the original ones.

Outline

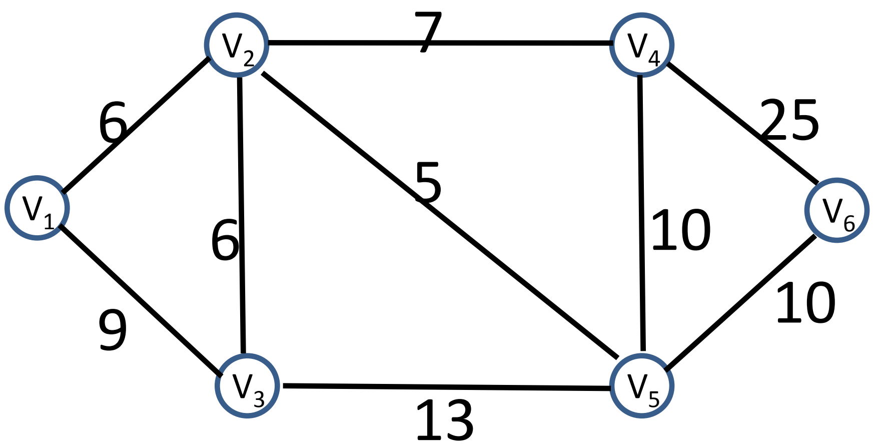
- Privacy-Preserving Social Networks
- Gaussian Perturbation
- Greedy Perturbation
- Experimental Results
- Conclusion

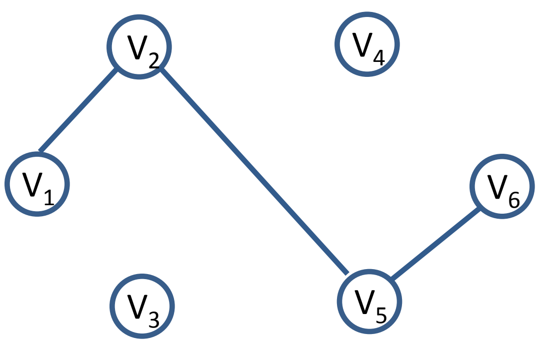
Greedy Perturbation

- Gaussian Perturbation is quick and independent with global structure. But It cannot always keep the same shortest paths when perturbation get larger

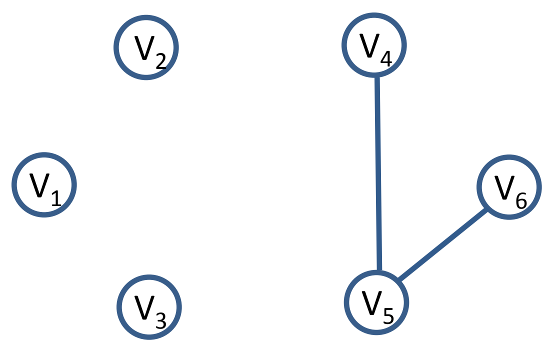
- So we propose alternative Greedy Strategy which can keep the exact shortest paths, and try to make corresponding lengths close to the original ones.

An example: Shortest Path Set ***H***

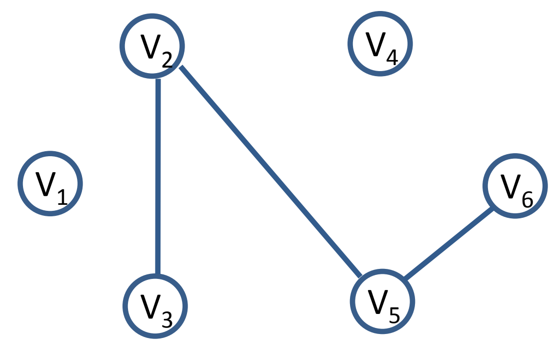




the shortest path $\boldsymbol{p_{1,6}}$

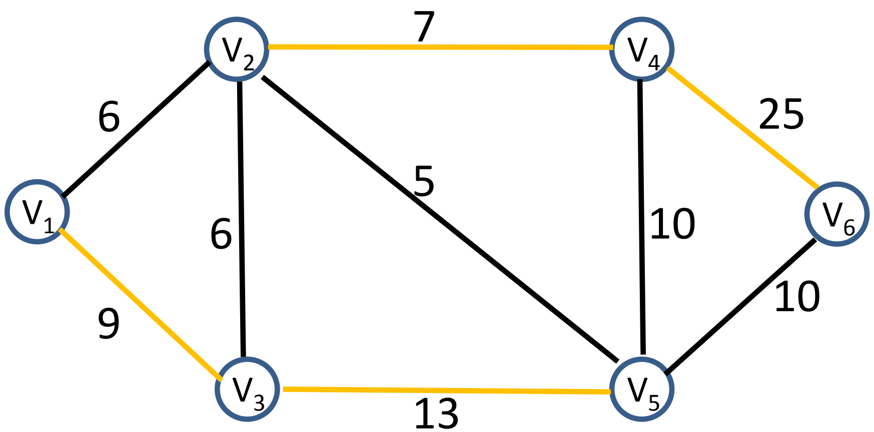


the shortest path $\boldsymbol{p_{4,6}}$

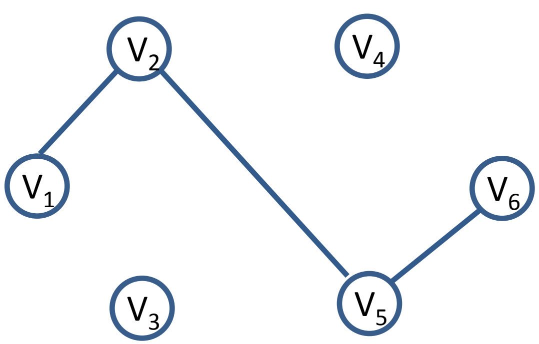


the shortest path $\boldsymbol{p_{3,6}}$

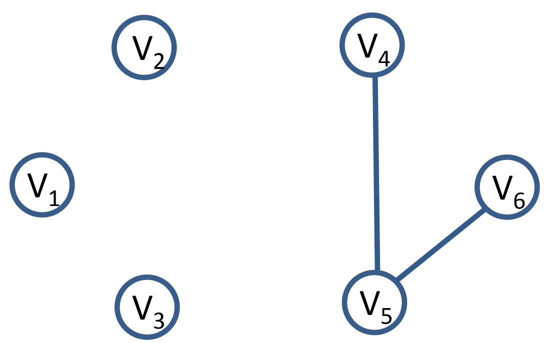
Edge Categorization



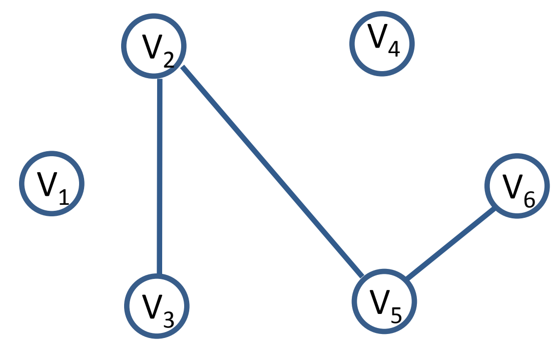
non-visited edges



the shortest path $\boldsymbol{p_{1,6}}$

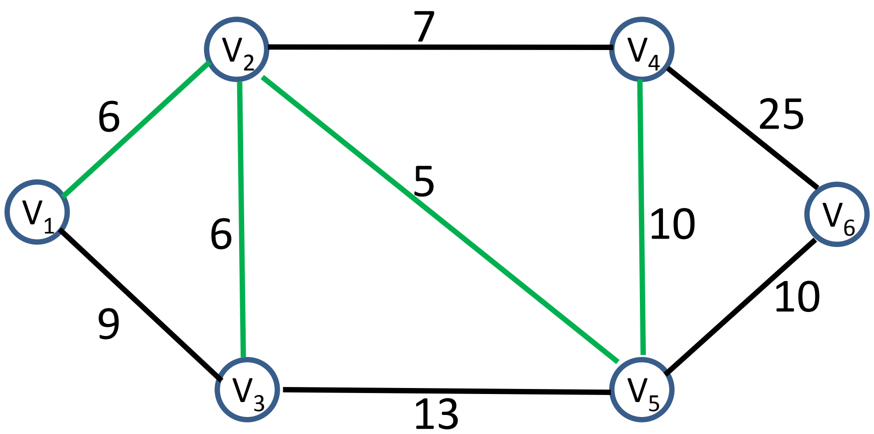


the shortest path $\boldsymbol{p_{4,6}}$

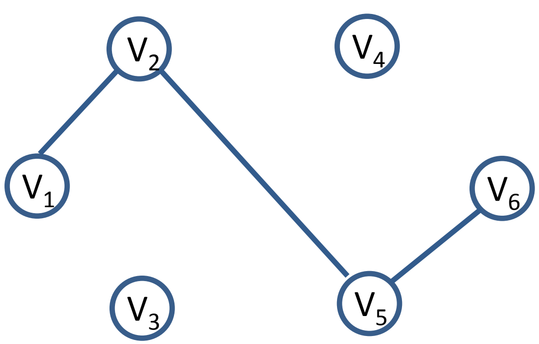


the shortest path $\boldsymbol{p_{3,6}}$

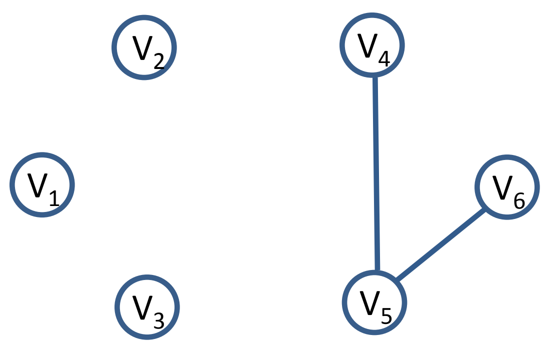
Edge Categorization



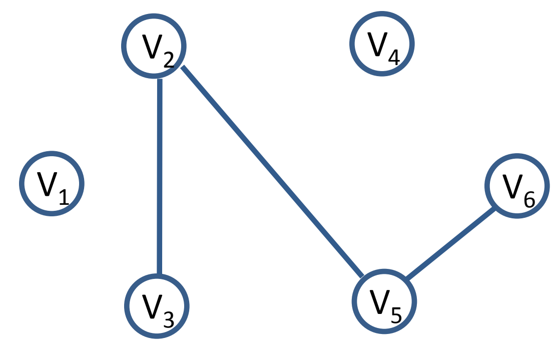
partially-visited edges



the shortest path $\boldsymbol{p_{1,6}}$

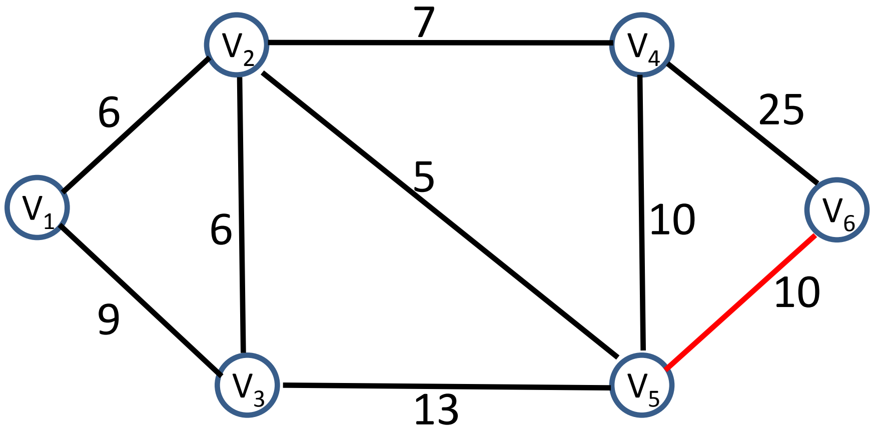


the shortest path $\boldsymbol{p_{4,6}}$

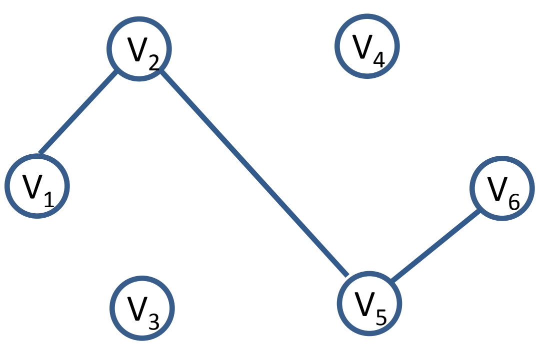


the shortest path $\boldsymbol{p_{3,6}}$

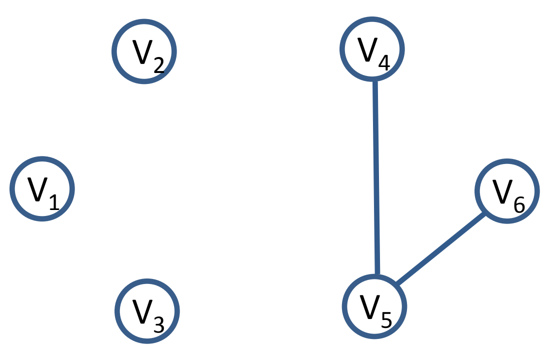
Edge Categorization



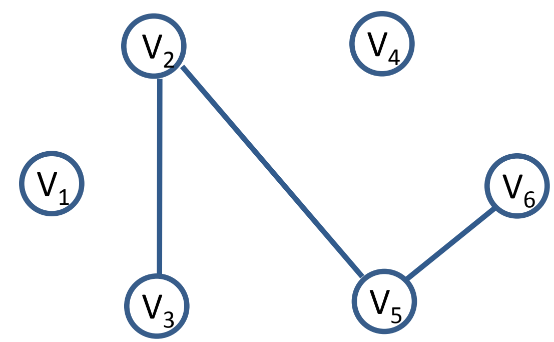
all-visited edges



the shortest path $\boldsymbol{p_{1,6}}$

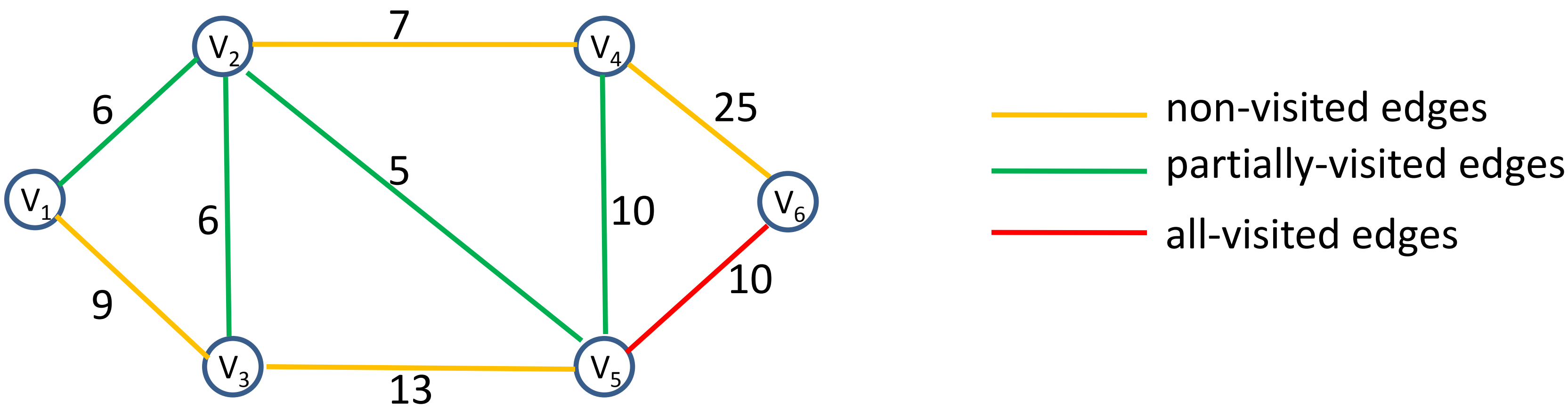


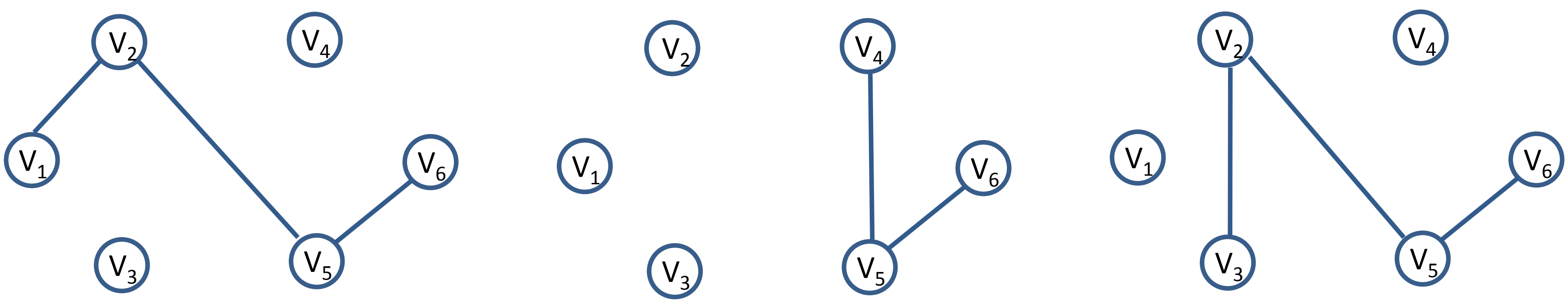
the shortest path $\boldsymbol{p_{4,6}}$



the shortest path $\boldsymbol{p_{3,6}}$

Edge Categorization

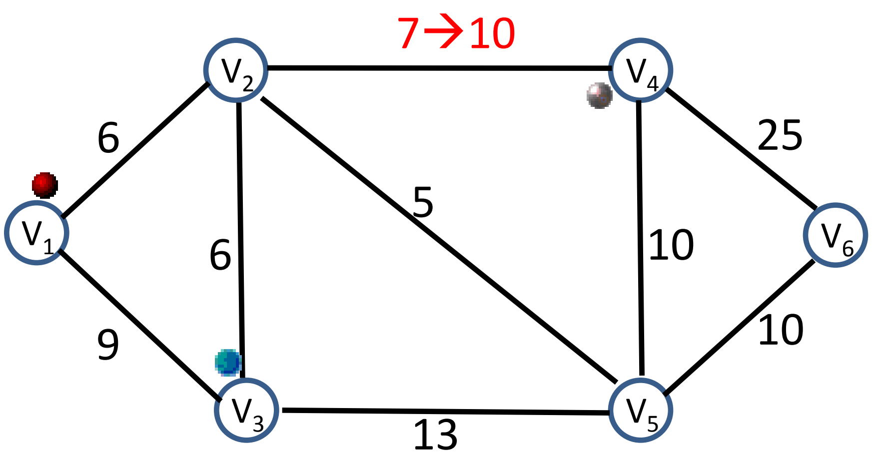




the shortest path $\boldsymbol{p_{1,6}}$ the shortest path $\boldsymbol{p_{4,6}}$ the shortest path $\boldsymbol{p_{3,6}}$

Non-Visited Edge

For a non-visited edge, increasing its weight will NOT change all shortest paths (and lengths) in *H*. *



$P_{1,6}$ *(no change)*

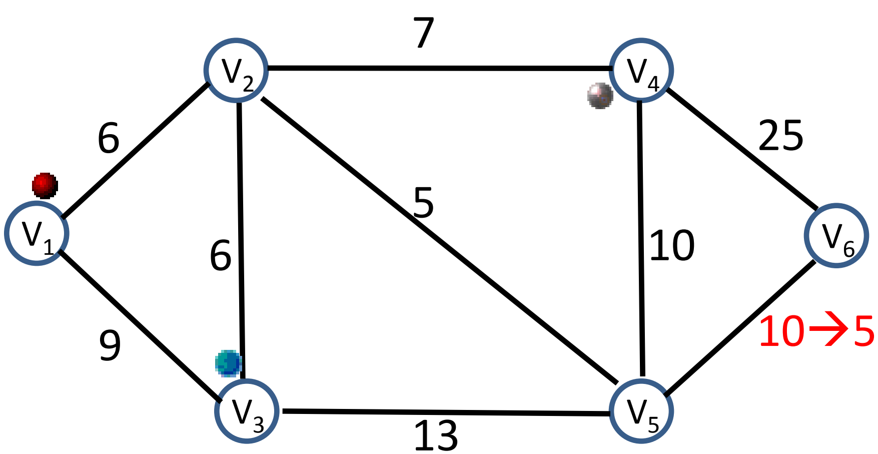
$P_{3,6}$ *(no change)*

$P_{4,6}$ *(no change)*

*Formal definition is referred to Proposition 7.

All-Visited Edge

For an all-visited edge, decreasing its weight will NOT change all shortest paths in *H*, but decrease the length of corresponding shortest paths. *



$P_{1,6}$ *(no change)*

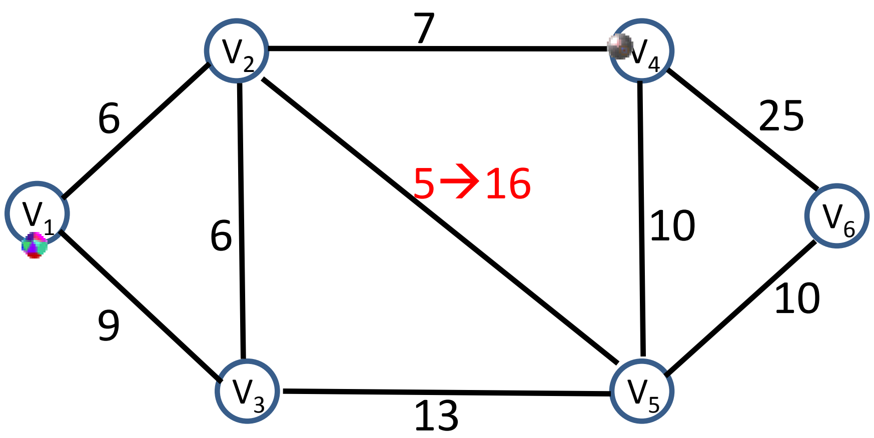
$P_{3,6}$ *(no change)*

$P_{4,6}$ *(no change)*

*Formal definition is referred to 8.

Partially-Visited Edge

For a partially-visited edge, we want to increase its weight by ***t.*** *



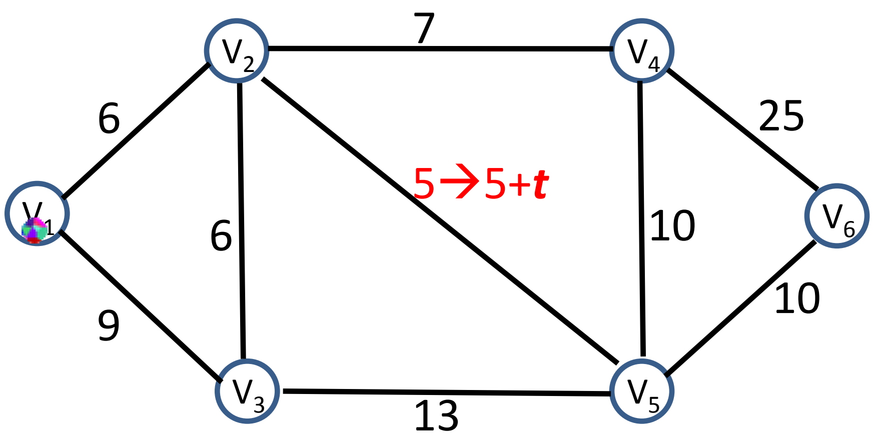
$P_{1,6}$ (changed)

$P_{4,6}$ (no change)

* How do we guarantee it (i.e., impose some constraints over the weight increasing) will be shown as Proposition 9 in our paper.

Partially-Visited Edge

For a partially-visited edge, we want to increase its weight by ***t***. *



$P_{1,6}$ *(probably change to* $P^{-}_{1.6}$ *)*

$P_{3,6}$ *(probably change to* $P^{-}_{3,6}$ *)*

$P_{4,6}$ *(no change)*

$P^{-}_{1,6}$, the shortest path between V_1 and V_6 in G^- (G deletes the edge between V_2 and V_5)

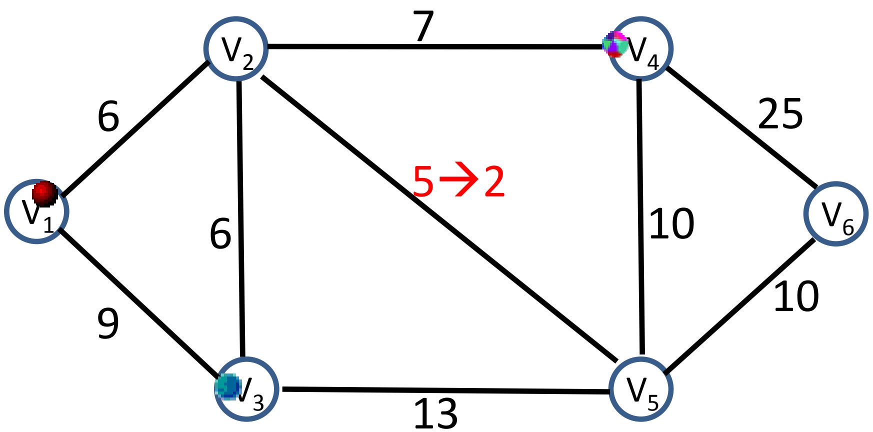
Constraints: the weight increment ***t*** should be smaller than the diff. between $d_{i,j}$ and $d^{-}_{I,j}$.

$$0 < t \quad < \quad \min\{d^{-}_{s_1,s_2} - d_{s_1,s_2} | \\ \textit{for all } p_{s_1,s_2} \textit{ such that } e_{i,j} \in p_{s_1,s_2}\}$$

* How do we guarantee it (i.e., impose some constraints over the weight increasing) will be shown as Proposition 9 in our paper.

Partially-Visited Edge

For a partially-visited edge, we want to decrease its weight by ***t***. *



$P_{1,6}$ (no change)

$P_{3,6}$ (no change)

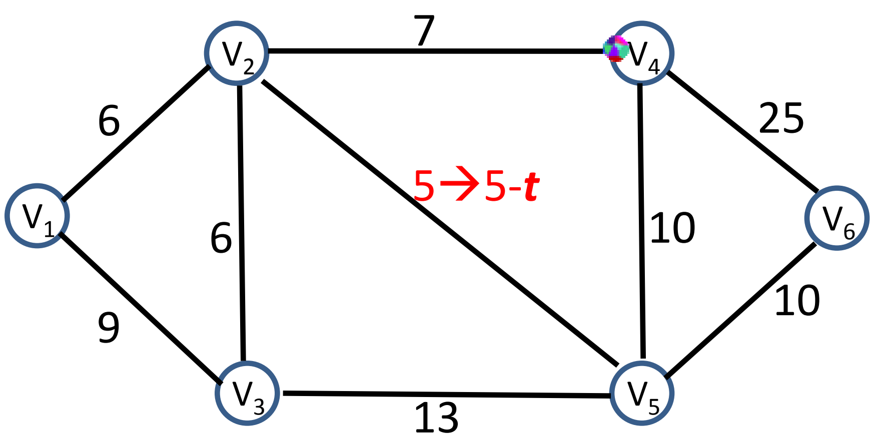
$P_{4,6}$ (changed)

---

* How do we guarantee it (i.e., impose some constraints over the weight decreasing) will be shown as Proposition 10 in our paper.

Partially-Visited Edge

For a partially-visited edge, we want to decrease its weight by ***t***. *



$P_{1,6}$ (no change)

$P_{3,6}$ (no change)

$P_{4,6}$ (probably change to $P^+_{4,6}$)

$P^+_{4,6}$, the shortest path between V_4 and V_6 through edge (V_2 → V_5)

Constraints: the weight decrement ***t*** should be larger than the diff. between $d^+_{i,j}$ and $d_{I,j}$.

$$0 < t < \min\{d_{s_1,i} + w_{i,j} + d_{j,s_2} - d_{s_1,s_2} | \text{ for all } p_{s_1,s_2} \text{ such that } e_{i,j} \notin p_{s_1,s_2}\}$$

* How do we guarantee it (i.e., impose some constraints over the weight decreasing) will be shown as Proposition 10 in our paper.

Greedy Algorithm

1. Increase non-visited edges and decrease all-visited edges.
2. Sort all partially-visited edges in descending order to a stack ***S*** by the number of shortest paths going through them.
3. For a given partially-visited edge in ***S***
   a. either increasing or decreasing its weight depends on the comparison between the real length and the current length.
   b. the modified value ***t*** is chosen as the boundary value of constraint inequalities.
   c. After modification, delete this one from ***S***.

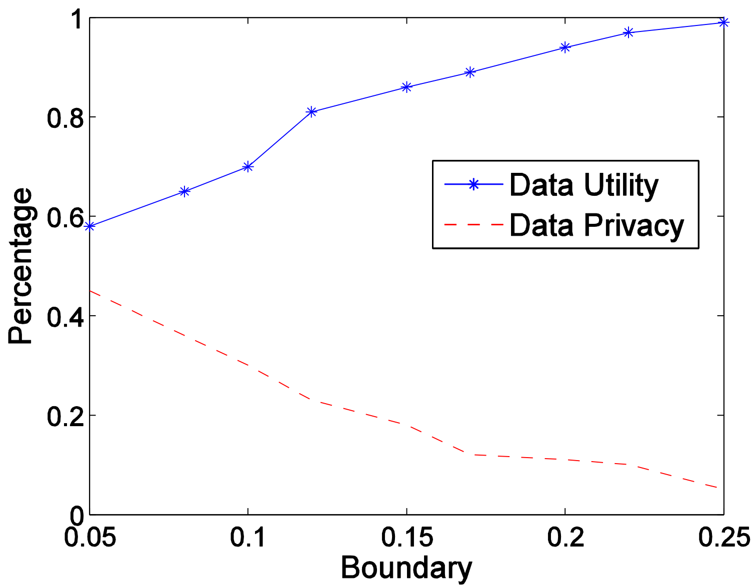
* For the detailed algorithm, please refer to Algorithm 1 in our paper.

Outline

- Privacy-Preserving Social Networks
- Gaussian Perturbation
- Greedy Perturbation
- Experimental Results
- Conclusion

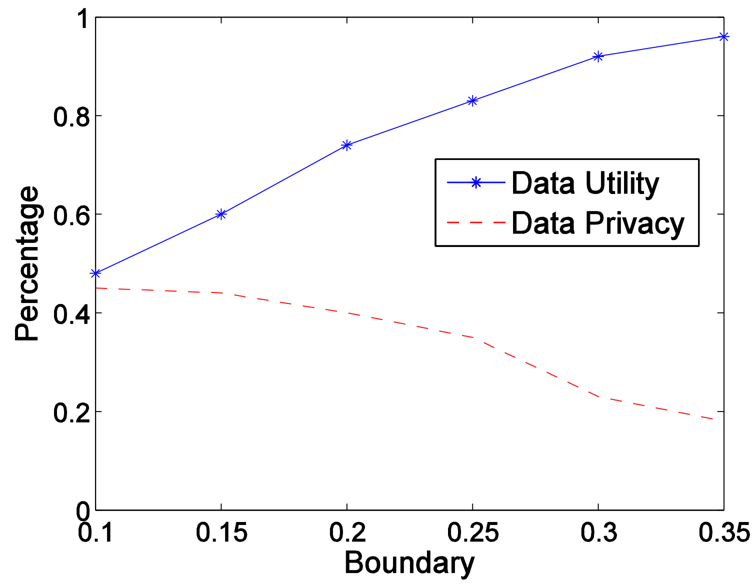
Experiments about Data Privacy and Data Utility

Gaussian Perturbation (σ=0.1)



Same shortest paths can not be guaranteed.

Greedy Perturbation (H=77%)



Same shortest paths can be guaranteed.

For more experiments based on different σ, please refer to Figure 11 and 12 in our paper.

Discussion on Experiments

	Data Utility	Data Privacy
Gaussian	Lengths of shortest paths are better preserved, cannot guarantee maintain the exact shortest path.	Low
Greedy	Length of shortest path is not well preserved compared to Gaussian. But the shortest paths are exactly maintained.	High

Conclusion

- **What do we want to do?**
  - Keep weight privacy and the shortest path utility.
- **Why do we want to do?**
  - Weights in some social cases are sensitive and confidential.
- **How do we do?**
  - Gaussian perturbation and greedy perturbation are proposed to achieve the balance between data utility and data privacy in different conditions.
- **What we do is applicable?**
  - Based on experiments, it seems that the two strategies do meet the expectation of our purpose.

Q&A

Thank you very much!

How to achieve data utility and data privacy

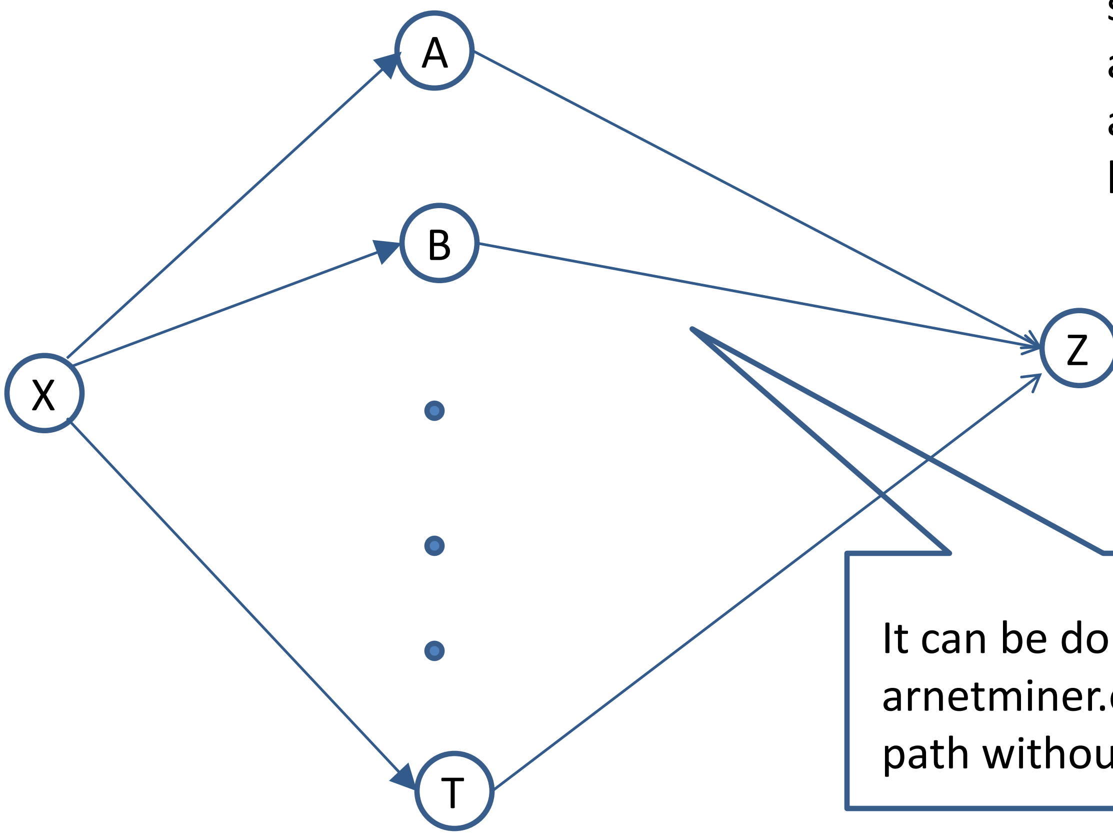
- **How to change weights as much as possible?**
  Boundary value of constraint inequalities.

- **How to guarantee the shortest paths the same as original ones?**
  If the modified weights satisfy constraints (Proposition 7—10 in our paper), it can be guaranteed.

- **How to make the length of the shortest paths as close to original ones as possible?**
  Alternating process of weight increasing and decreasing.

Why don't just hide weights

If I know the lengths of some shortest paths, I can choose an optimal Professor to write an useful recommendation letter.



It can be done via arnetminer.org. Just shortest path without weights