Personalized Social Recommendations - Accurate or Private?

Ashwin Machanavajjhala, Aleksandra Korolova, Atish Das Sarma

January 21, 2013
Outline
Making recommendations or suggestions to users in order to increase their degree of engagement is a common practice to websites.
Social Recommendations

- Making recommendations or suggestions to users in order to increase their degree of engagement is a common practice to websites.
- The phenomenal participation of users in social networks such as Facebook and LinkedIn, has given a tremendous hope for designing a new type of user experience, the *social* one.
Social Recommendations

- Making recommendations or suggestions to users in order to increase their degree of engagement is a common practice to websites.
- The phenomenal participation of users in social networks such as Facebook and LinkedIn, has given a tremendous hope for designing a new type of user experience, the social one.
- The feasibility of social recommendations has been fueled by initiatives such as Facebook’s Open Graph API and Google’s Social Graph API.
Making recommendations or suggestions to users in order to increase their degree of engagement is a common practice to websites.

The phenomenal participation of users in social networks such as Facebook and LinkedIn, has given a tremendous hope for designing a new type of user experience, the social one.

The feasibility of social recommendations has been fueled by initiatives such as Facebook’s Open Graph API and Google’s Social Graph API.

While traditional recommender systems default to generic recommendations, a social-network aware system can provide recommendations based on active friends.
Making recommendations or suggestions to users in order to increase their degree of engagement is a common practice to websites.

The phenomenal participation of users in social networks such as Facebook and LinkedIn, has given a tremendous hope for designing a new type of user experience, the social one.

The feasibility of social recommendations has been fueled by initiatives such as Facebook’s Open Graph API and Google’s Social Graph API.

While traditional recommender systems default to generic recommendations, a social-network aware system can provide recommendations based on active friends.

We will focus on recommendation algorithms based exclusively on graph-link analysis.
Improved social recommendations come at a cost - they can potentially lead to a *privacy breach* by revealing sensitive information.
Improved social recommendations come at a cost - they can potentially lead to a privacy breach by revealing sensitive information.

Example (Reveal Shopping History)

For instance, if you only have one friend, a social recommendation algorithm that recommends to you only the products that your friend buy, would reveal the entire shopping history of that friend.
Improved social recommendations come at a cost - they can potentially lead to a privacy breach by revealing sensitive information.

Example (Reveal Shopping History)

For instance, if you only have one friend, a social recommendation algorithm that recommends to you only the products that your friend buy, would reveal the entire shopping history of that friend.

Example (Lack of Trust)

A system that uses only trusted edges in friend suggestions may leak information about lack of trust along specific edges.
The paper is a first theoretical study of the privacy-utility trade-off in personalized graph link-analysis based social recommender system.
The Paper Framework

- The paper is a first theoretical study of the privacy-utility trade-off in personalized graph link-analysis based social recommender system.
- There are many different settings in which social recommendations may be used, however all these problems have a common structure - social graph.
The paper is a first theoretical study of the privacy-utility trade-off in personalized graph link-analysis based social recommender system.

There are many different settings in which social recommendations may be used, however all these problems have a common structure - social graph.

The main contributions are intuitive and precise trade-off results between privacy and utility for a clear formal model of personalized social recommendations.
The steps for formalizing the problem will be:

1. Describe what a social algorithm entails.
The steps for formalizing the problem will be:

1. Describe what a social algorithm entails.
2. State the chosen notion of privacy - *differential privacy*.
Modeling the Problem

The steps for formalizing the problem will be:

1. Describe what a social algorithm entails.
2. State the chosen notion of privacy - *differential privacy*.
3. Define the accuracy of an algorithm.
Modeling the Problem

The steps for formalizing the problem will be:

1. Describe what a social algorithm entails.
2. State the chosen notion of privacy - *differential privacy*.
3. Define the accuracy of an algorithm.
4. State the problem of designing a private and accurate social recommendation algorithm.
Let $G = (V, E)$ be the graph that describes the network of connections between people and entities.
Social Recommendation Algorithm

- Let $G = (V, E)$ be the graph that describes the network of connections between people and entities.
- Each recommendation is an edge $(i, r)$ where node $i$ is recommended to target node $r$. 
Social Recommendation Algorithm

- Let $G = (V, E)$ be the graph that describes the network of connections between people and entities.
- Each recommendation is an edge $(i, r)$ where node $i$ is recommended to target node $r$.
- We denote the utility of recommending node $i$ to node $r$ by $u_{G,r}^i$. 
Social Recommendation Algorithm

- Let $G = (V, E)$ be the graph that describes the network of connections between people and entities.
- Each recommendation is an edge $(i, r)$ where node $i$ is recommended to target node $r$.
- We denote the utility of recommending node $i$ to node $r$ by $u_{i}^{G,r}$.
- The utility is some function of the structure of $G$. 
Social Recommendation Algorithm

- Let $G = (V, E)$ be the graph that describes the network of connections between people and entities.
- Each recommendation is an edge $(i, r)$ where node $i$ is recommended to target node $r$.
- We denote the utility of recommending node $i$ to node $r$ by $u_{i,r}^G$.
- The utility is some function of the structure of $G$.
- We assume that a recommendation algorithm $R$ is a probability vector on all nodes, where $p_{i,r}^G$ denotes the probability of recommending node $i$ to node $r$ in graph $G$ by the specified algorithm $G$. 
Since privacy protections are extremely important in social networks, we will use a strong definition of privacy, *Differential Privacy*.
Since privacy protections are extremely important in social networks, we will use a strong definition of privacy, *Differential Privacy*.

**An algorithm preserves privacy of an entity if the algorithm’s output is not sensitive to the presence or absence of the entity’s information in the input data set.**
Privacy Definition - *Differential Privacy*

- Since privacy protections are extremely important in social networks, we will use a strong definition of privacy, *Differential Privacy*.

- **An algorithm preserves privacy of an entity if the algorithm’s output is not sensitive to the presence or absence of the entity’s information in the input data set.**

- **In our setting of graph link-analysis based social recommendations, we wish to maintain the presence (or absence) of an edge in the graph private.**

**Definition**

A recommendation algorithm $R$ satisfies $\epsilon$-differential privacy if for any pair of graphs $G$ and $G'$ that differ in one edge and every set of possible recommendation $S$,

$$Pr[R(G) \in S] \leq \exp(\epsilon) \times Pr[R(G') \in S]$$
Accuracy of an Algorithm

- For simplicity, we focus on the problem of making recommendations for a fixed target node $r$. 
For simplicity, we focus on the problem of making recommendations for a fixed target node $r$.

Therefore the algorithm takes as input only one utility vector $u$ and returns one probability vector $p$. 
For simplicity, we focus on the problem of making recommendations for a fixed target node $r$.

Therefore the algorithm takes as input only one utility vector $u$ and returns one probability vector $p$.

**Definition 2 - Accuracy**

The accuracy of an algorithm $R$ is defined as $\min_{\bar{u}} \frac{\sum u_ip_i}{u_{\max}}$, where $u_{\max} = \max_i (u_i)$. 
In other words, an algorithm is \((1 - \delta)\)-accurate if (1) the output \(p_i\) are such that \(\frac{\sum u_i p_i}{u_{\text{max}}} \geq (1 - \delta)\), and (2) there exists an input utility vector \(\vec{u}\) such that the output \(p_i\) satisfies \(\frac{\sum u_i p_i}{u_{\text{max}}} = (1 - \delta)\).
In other words, an algorithm is \((1 - \delta)\)-accurate if (1) the output \(p_i\) are such that \(\sum \frac{u_i p_i}{u_{\text{max}}} \geq (1 - \delta)\), and (2) there exists an input utility vector \(\vec{u}\) such that the output \(p_i\) satisfies \(\sum \frac{u_i p_i}{u_{\text{max}}} = (1 - \delta)\).

We follow the paradigm of worst-case performance analysis from the algorithm literature.
Problem Statement

Definition 3 - *Private Social Recommendations*

*Design a social recommendation algorithm* $R$ *with maximum possible accuracy under the constraint that* $R$ *satisfies $\epsilon$-differential privacy.*
Instead of assuming a specific graph link-based recommendation algorithm, more ambitiously, we aim to determine accuracy bounds for a general class of recommendation algorithms.
Instead of assuming a specific graph link-based recommendation algorithm, more ambitiously, we aim to determine accuracy bounds for a general class of recommendation algorithms.

We first define properties that one can expect most reasonable utility functions and recommendation algorithms to satisfy.
Instead of assuming a specific graph link-based recommendation algorithm, more ambitiously, we aim to determine accuracy bounds for a general class of recommendation algorithms.

We first define properties that one can expect most reasonable utility functions and recommendation algorithms to satisfy.

We then present a general bound that applies to all algorithms and utility functions satisfying those properties.
Properties of Utility Functions and Algorithms

A meaningful utility function in the context of recommendations on social network should be satisfy two axioms:
A meaningful utility function in the context of recommendations on social network should be satisfy two axioms:

**Axiom 1 (Exchangeability)**

Let $G$ be a graph and let $h$ be an isomorphism on the nodes giving graph $G_h$, s.t. for target node $r$, $h(r) = r$. Then $\forall i : u_i^{G,r} = u_{h(i)}^{G_h,r}$.

- The utility of a node $i$ should not depend on the node’s identity.
A meaningful utility function in the context of recommendations on social network should be satisfy two axioms:

**Axiom 1 (Exchangeability)**

Let $G$ be a graph and let $h$ be an isomorphism on the nodes giving graph $G_h$, s.t. for target node $r$, $h(r) = r$. Then $orall i : u^G_{i,r} = u^{G_h}_{h(i),r}$.

- The utility of a node $i$ should not depend on the node’s identity.
- The utility for target node $r$ only depends on the structural properties of the graph, and so, nodes isomorphic from the perspective of $r$ should have the same utility.
Axiom 2 (Concentration)

There exists $S \subset V(G)$, such that $|S| = \beta$, and

$$\sum_{i \in S} u_i \geq \Omega(1) \sum_{i \in V(G)} u_i$$

- This says there are some $\beta$ nodes that together have at least a constant fraction of the total utility.
Properties of Utility Functions and Algorithms

Axiom 2 (Concentration)

There exists $S \subset V(G)$, such that $|S| = \beta$, and

$$\sum_{i \in S} u_i \geq \Omega(1) \sum_{i \in V(G)} u_i$$

- This says there are some $\beta$ nodes that together have at least a constant fraction of the total utility.
- In large graphs there are usually a small number of nodes that are very good recommendations for $r$ and a long tail of those that are not.
Properties of Utility Functions and Algorithms

We now define a property of a recommendation algorithm:

**Definition 4 (Monotonicity)**

An algorithm is said to be monotonic if \( \forall i, j, u_i > u_j \) implies that \( p_i > p_j \).
We now define a property of a recommendation algorithm:

**Definition 4 (Monotonicity)**

An algorithm is said to be monotonic if \( \forall i, j, u_i > u_j \) implies that \( p_i > p_j \).

- A very natural notion for recommendation algorithm to satisfy - the algorithm recommends a higher utility node with a higher probability than a lower utility node.
Properties of Utility Functions and Algorithms

We now define a property of a recommendation algorithm:

**Definition 4 (Monotonicity)**

An algorithm is said to be monotonic if $\forall i, j, u_i > u_j$ implies that $p_i > p_j$.

- A very natural notion for recommendation algorithm to satisfy - the algorithm recommends a higher utility node with a higher probability than a lower utility node.

**Example (Number of common neighbors utility function)**

Given a target node $r$ and graph $G$, the number of common neighbors utility function assigns a utility $u_i^{G,r} = C(i, r)$, where $C(i, r)$ is the number of common neighbors between $i$ and $r$. 
General Lower Bound

- Lower bound on the privacy parameter $\epsilon$ for any differentially private recommendation algorithm that (a) achieves a constant accuracy and (b) is based on any utility functions that satisfies the former axioms.
General Lower Bound

- Lower bound on the privacy parameter $\epsilon$ for any differentially private recommendation algorithm that (a) achieves a constant accuracy and (b) is based on any utility functions that satisfies the former axioms.

- Proof technique for the lower bound using the number of common neighbors utility metric.
Let $r$ be the target node for recommendation.
General Lower Bound

1. Let $r$ be the target node for recommendation.
2. The nodes in any graph can be split into two groups - $V^r_{hi}$, nodes which have a high utility for the target node $r$ and $V^r_{lo}$, nodes that have a low utility.
General Lower Bound

2 The nodes in any graph can be split into two groups - $V^r_{hi}$, nodes which have a high utility for the target node $r$ and $V^r_{lo}$, nodes that have a low utility.

3 Since the recommendation algorithm has to achieve a constant accuracy, it has to recommend one of the high utility nodes with constant probability.
3. Since the recommendation algorithm has to achieve a constant accuracy, it has to recommend one of the high utility nodes with constant probability.

4. By the concentration axiom, there are only a few nodes in $V_{hi}$, but there are many nodes in $V_{lo}$. 
By the concentration axiom, there are only a few nodes in $V_{hi}^r$, but there are many nodes in $V_{lo}^r$.

Hence, there exists a node $i$ in the high utility group and a node $l$ in the low utility group such that $\Gamma = \frac{p_i}{p_l}$ is very large.
5 Hence, there exists a node $i$ in the high utility group and a node $l$ in the low utility group such that $\Gamma = \frac{p_i}{p_l}$ is very large.

6 We show that we can carefully modify the graph $G$ by adding/or deleting a small number ($t$) of edges in such a way that the node $l$ becomes the node with highest utility in $G'$ (using the exchangeability axiom).
We show that we can carefully modify the graph $G$ by adding/or deleting a small number ($t$) of edges in such a way that the node $l$ becomes the node with highest utility in $G'$ (using the exchangeability axiom).
It now follows from differential privacy that $\epsilon \geq \frac{1}{t} \log \Gamma$. 
General Lower Bound

- It now follows from differential privacy that $\epsilon \geq \frac{1}{t} \log \Gamma$.
- After generalizing it further, we will get the following lemma states the main trade-off relationship between the accuracy parameter $1 - \delta$ and the privacy parameter $\epsilon$ of a recommendation algorithm: $\epsilon \geq \frac{1}{t} (\ln\left(\frac{c-\delta}{\delta}\right) + \ln\left(\frac{n-k}{k+1}\right))$. 
It now follows from differential privacy that $\epsilon \geq \frac{1}{t} \log \Gamma$.

After generalizing it further, we will get the following lemma states the main trade-off relationship between the accuracy parameter $1 - \delta$ and the privacy parameter $\epsilon$ of a recommendation algorithm: $\epsilon \geq \frac{1}{t} \left( \ln \left( \frac{c-\delta}{\delta} \right) + \ln \left( \frac{n-k}{k+1} \right) \right)$.

The lemma gives us a lower bound on the privacy guarantee $\epsilon$ in terms of the accuracy parameter $1 - \delta$. 
It now follows from differential privacy that $\epsilon \geq \frac{1}{t} \log \Gamma$.  

After generalizing it further, we will get the following lemma states the main trade-off relationship between the accuracy parameter $1 - \delta$ and the privacy parameter $\epsilon$ of a recommendation algorithm:

$$\epsilon \geq \frac{1}{t} \left( \ln \left( \frac{c-\delta}{\delta} \right) + \ln \left( \frac{n-k}{k+1} \right) \right).$$

The lemma gives us a lower bound on the privacy guarantee $\epsilon$ in terms of the accuracy parameter $1 - \delta$.

Using the concentration axiom with parameter $\beta$ they prove: For $(1 - \delta) = \Omega(1)$ and $\beta = o\left(\frac{n}{\log n}\right)$,

$$\epsilon \geq \frac{\log n - o(\log n)}{t}.$$
Theorem 1

For a graph with maximum degree $d_{\text{max}} = \alpha \log n$ a differentially private algorithm can guarantee constant accuracy only if $\epsilon \geq \frac{1}{\alpha} \left( \frac{1}{4} - o(1) \right)$. 

Example (A graph with maximum degree $\log n$)

As an example, the theorem implies that for a graph with maximum degree $\log n$, there is no 0.24-differentially private algorithm that achieves any constant accuracy.
Theorem 1

For a graph with maximum degree $d_{\text{max}} = \alpha \log n$ a differentially private algorithm can guarantee constant accuracy only if $\epsilon \geq \frac{1}{\alpha} \left( \frac{1}{4} - o(1) \right)$.

Example (A graph with maximum degree $\log n$)

As an example, the theorem implies that for a graph with maximum degree $\log n$, there is no 0.24-differentially private algorithm that achieves any constant accuracy.
Theorem 1

For a graph with maximum degree $d_{max} = \alpha \log n$ a differentially private algorithm can guarantee constant accuracy only if $\epsilon \geq \frac{1}{\alpha} \left( \frac{1}{4} - o(1) \right)$.

Example (A graph with maximum degree $\log n$)

As an example, the theorem implies that for a graph with maximum degree $\log n$, there is no $0.24$-differentially private algorithm that achieves any constant accuracy.

- The model can be extended.
Specific Utility lower Bounds

- We’ll prove a stronger lower bounds for particular utility functions using tighter upper bounds on $t$. 
Consider a graph and a target node $r$. 
Consider a graph and a target node \( r \).

We can make any node \( x \) have the highest utility by adding edges from it to all \( r \)’s neighbors.
Consider a graph and a target node \( r \).

We can make any node \( x \) have the highest utility by adding edges from it to all \( r \)'s neighbors.

If \( d_r \) is \( r \)'s degree, it suffices to add \( t = d_r + O(1) \) edges to make a node the highest utility node.
Consider a graph and a target node $r$.

We can make any node $x$ have the highest utility by adding edges from it to all $r$’s neighbors.

If $d_r$ is $r$’s degree, it suffices to add $t = d_r + O(1)$ edges to make a node the highest utility node.

**Theorem 2**

Let $U$ be a utility function that depends only on and is monotonically increasing with $C(x, y)$, the number of common neighbors between $x$ and $y$. A recommendation algorithm based on $U$ that guarantees any constant accuracy for target node $r$ has a lower bound on privacy given by $\epsilon \geq \frac{1-o(1)}{\alpha}$ where $d_r = \alpha \log n$. 
This is a very strong lower bound.
Privacy Bound for Common Neighbors

- This is a very strong lower bound.
- Since significant fraction of nodes in real-world graphs have small $d_r$, we can expect no algorithm based on common neighbors utility to be both accurate and satisfy differential with reasonable $\epsilon$. 

Example ($\text{Maximum Degree} - \log n$)

To understand the consequence of this theorem, consider an example of a graph on $n$ nodes with maximum degree $\log n$. Any algorithm that makes recommendations based on the common neighbors utility function achieves a constant accuracy is at best 1-differentially private.
Privacy Bound for Common Neighbors

- This is a very strong lower bound.
- Since significant fraction of nodes in real-world graphs have small $d_r$, we can expect no algorithm based on common neighbors utility to be both accurate and satisfy differential with reasonable $\epsilon$.
- Moreover, this is contrary to the commonly held belief that one can eliminate privacy risk by connecting to a few high degree nodes.
Privacy Bound for Common Neighbors

- This is a very strong lower bound.
- Since significant fraction of nodes in real-world graphs have small $d_r$, we can expect no algorithm based on common neighbors utility to be both accurate and satisfy differential privacy with reasonable $\epsilon$.
- Moreover, this is contrary to the commonly held belief that one can eliminate privacy risk by connecting to a few high degree nodes.

**Example (Maximum Degree - $\log n$)**

To understand the consequence of this theorem, consider an example of a graph on $n$ nodes with maximum degree $\log n$. Any algorithm that makes recommendations based on the common neighbors utility function achieves a constant accuracy is at best 1.0-differentially private.
Privacy-Preserving Algorithms

Exponential Mechanism

The exponential mechanism creates a smooth probability distribution from the utility vector and samples from it.
Privacy-Preserving Algorithms

Exponential Mechanism

The exponential mechanism creates a smooth probability distribution from the utility vector and samples from it.

Laplace Mechanism

The Laplace mechanism first adds random noise drawn from a Laplace distribution and like the optimal mechanism, picks the node with the maximum noise-infused utility.

- Algorithms $A_L(\epsilon)$ and $A_E(\epsilon)$ guarantee $\epsilon$ differential privacy.
Privacy-Preserving Algorithms

Exponential Mechanism

The exponential mechanism creates a smooth probability distribution from the utility vector and samples from it.

Laplace Mechanism

The Laplace mechanism first adds random noise drawn from a Laplace distribution and like the optimal mechanism, picks the node with the maximum noise-infused utility.

- Algorithms $A_L(\epsilon)$ and $A_E(\epsilon)$ guarantee $\epsilon$ differential privacy.
- $A_L(\epsilon)$ and $A_E(\epsilon)$ achieve very similar accuracies.
Privacy-Preserving Algorithms

Exponential Mechanism

The exponential mechanism creates a smooth probability distribution from the utility vector and samples from it.

Laplace Mechanism

The Laplace mechanism first adds random noise drawn from a Laplace distribution and like the optimal mechanism, picks the node with the maximum noise-infused utility.

- Algorithms $A_L(\epsilon)$ and $A_E(\epsilon)$ guarantee $\epsilon$ differential privacy.
- $A_L(\epsilon)$ and $A_E(\epsilon)$ achieve very similar accuracies.
- Both algorithms assume the knowledge of the entire utility vector.
Present experimental results on two real-world graphs and for two particular utility functions.

Compute accuracies achieved by the Laplace and Exponential mechanisms and compare them with the theoretical upper bound on accuracy that any $\epsilon$-differentially private algorithm can hope to achieve.

We use two publicly available networks - Wikipedia vote network ($G_{WV}$) and Twitter connections network ($G_T$).

We use two particular utility functions: the number of common neighbors and weighted paths (practical use by many companies).
Figure 1: Accuracy of algorithms using \# of common neighbors utility function for two privacy settings. X-axis is the accuracy $(1-\delta)$ and y-axis is the % of nodes receiving recommendations with accuracy $\leq 1-\delta$. 
(a) Accuracy on Wiki vote network using \# of weighted paths as the utility function, for \( \epsilon = 1 \).

(b) Accuracy on Twitter network using \# of weighted paths as the utility function, for \( \epsilon = 1 \).
Results

- **Exponential vs Laplace mechanism:** All experiments verified that Laplace mechanism achieves nearly identical accuracy as the Exponential mechanism.

- For a large fraction of nodes, the accuracy achieved by Laplace and Exponential mechanisms is close to the best possible accuracy suggested by the theoretical bound.

- For most nodes, our bounds suggest that there is an inevitable harsh trade-off between privacy and accuracy when making social recommendations, yielding poor accuracy under reasonable privacy parameter $\epsilon$. 
Future Work and Extensions

- Examine other utility functions.
Future Work and Extensions

- Examine other utility functions.
- Most works on making recommendations deal with static data.
Future Work and Extensions

- Examine other utility functions.
- Most works on making recommendations deal with static data.
- What happens when a certain edges are sensitive.
Future Work and Extensions

- Examine other utility functions.
- Most works on making recommendations deal with static data.
- What happens when a certain edges are sensitive.
- Examine weaker privacy notion than the differential.