# Networked Bandits with Disjoint Linear Payoffs

Meng Fang
Centre for Quantum Comp. & Intelligent Syst.
University of Technology, Sydney
235 Jones Street, Ultimo, NSW 2007, Australia
Meng.Fang@student.uts.edu.au

Dacheng Tao
Centre for Quantum Comp. & Intelligent Syst.
University of Technology, Sydney
235 Jones Street, Ultimo, NSW 2007, Australia
Dacheng.Tao@uts.edu.au

## ABSTRACT

In this paper, we study 'networked bandits', a new bandit problem where a set of interrelated arms varies over time and, given the contextual information that selects one arm, invokes other correlated arms. This problem remains under-investigated, in spite of its applicability to many practical problems. For instance, in social networks, an arm can obtain payoffs from both the selected user and its relations since they often share the content through the network. We examine whether it is possible to obtain multiple payoffs from several correlated arms based on the relationships. In particular, we formalize the networked bandit problem and propose an algorithm that considers not only the selected arm, but also the relationships between arms. Our algorithm is 'optimism in face of uncertainty' style, in that it decides an arm depending on integrated confidence sets constructed from historical data. We analyze the performance in simulation experiments and on two real-world offline datasets. The experimental results demonstrate our algorithm's effectiveness in the networked bandit setting.

## Categories and Subject Descriptors

H.3.5 [**Information Systems**]: Social and Information Networks; 1.2.6 [**Computing Methodologies**]: [Learning]

## General Terms

Algorithm, Theory

## Keywords

Networked bandits; social network; exploration/exploitation dilemma

## 1. INTRODUCTION

A multi-armed bandit problem (or bandit problem) is a sequential decision problem defined by a set of actions (or arms). The term 'bandit' originates from the colloquial term for a casino slot machine ('a one-armed bandit'), in which a player (or a forecaster) faces a finite number of slot machines (or arms). The player sequentially allocates coins (one at a time) to different machines and earns money (or payoff ) depending on the machine selected. The goal is to earn as high a payoff as possible.

Robbins formalized this problem in 1952 [20]; in the multi-armed bandit problem, $K$ arms exist that are associated with unknown payoff distributions, and a forecaster can select an arm sequentially. In each round of play, a forecaster selects one arm and then receives the payoff from the selected arm. The forecaster's aim is to maximize the total cumulative payoff, i.e., the sum of the payoffs of the chosen arms in total. Since the forecaster does not know the process generating the payoffs but has historical payoffinformation, the bandit problem highlights the fundamental difficultly of decision making in the face of uncertainty: balancing the decision of whether to exploit past choices or make new choices with the hope of discovering a better one.

The bandit problem has been studied for many years, with works primarily focusing on the theory and designing different algorithms based on different settings, such as the stochastic setting, adversarial setting, and contextual setting [9]. In real-world applications, the multi-armed bandit problem is an effective way of solving situations where one encounters an exploration-exploitation dilemma. It has historically been used to decide which clinical trial is better when multiple treatments are available for a given disease and there is a need to decide which treatment to use on the next patient.

Modern technologies have created many opportunities for use of the bandit problem, and it has a wide range of applications including advertising, recommendation systems, online systems, and games. For example, an advertising task may be the choice of which advertisement to display to the next visitor to a web page, where the payoff is associated with the visitor's actions. More recently, the bandit algorithm has been used in personalized recommendation tasks [17], where a user visits a website and the system collects the user's information. The system selectively provides content from a content pool through the user's current and past behaviors analyzing to best satisfy the user's needs, and the payoff is based on user-click feedback.

All the above bandit problems have the major underlying assumption that all the arms are independent, which is inappropriate for web-based social network applications. In a network, including social networks, the users are connected by relationships [2, 22]. Contextual information can


*KDD'14,* August 24–27, 2014, New York, NY, USA.

http://dx.doi.org/10.1145/2623330.2623672.

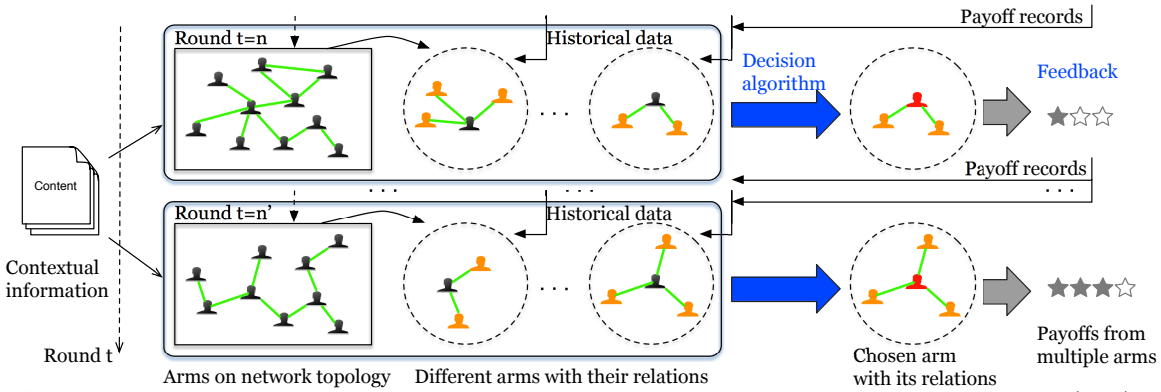



Figure 1: An overview of networked bandits at different rounds. The network is changing over time. An arm (user) can invoke other arms (relations) and have different relations at different rounds. Given the contextual information an arm is chosen by the decision algorithm for getting multiple payoffs (feedback). The algorithm improves the selection strategy after collecting new payoffinformation.

be obtained from other users and can be spread via these relationships. Content promoted to one user provides feedback, not only from that user, but also from his/her relations. For example, a user of Twitter or Facebook can read a tweet/message and can re-post someone else's tweet/message, allowing the user to quickly share it with his/her followers. Impact can be assessed by counting the number of 'favorites/likes' from different users' pages. Therefore, careful selection of a user for tweet/message posting can maximize the number of 'favorites/likes'.

Our study is motivated by the observation that even when a user is randomly selected for promotion, other users close to the selected user in the network will be influenced [18, 22]. Specifically, as shown in Figure 1, in a social network, if we promote a content to a user, the user may share it with others and the payoffs can be collected from the user and its relations. The goal is to gain higher payoffs. The process is similar to share-then-like, which occurs daily in social networks and needs to be considered for optimized recommendation and advertising tasks, the important point being that the context is extended from the selected user to all other invoked users. There are several challenges to realizing this problem. First, only partial information is available about the chosen users when content is posted, and the information of other users is unknown. Therefore, there is a dilemma of whether the system should select a user with the best payoff history or a new user in order to explore more possibilities. Second, the content may frequently change and few overlapping historical records may exist. Furthermore, relationships exist between users and these relationships may change over time. These challenges inspire us to formalize the networked bandit problem.

The above problem can be considered a balance of the trade-off between exploration (discovering a new user) and exploitation (using the current best user) when network topology is known.

We formalize a well-defined but simple setting for the networked bandit problem, in which there exist $K$ arms connected by network topology $G$. We propose an approach in which a learning algorithm optimally selects an arm at each round based on contextual information and the network topology information of arms. The networked bandit problem can be considered an extension of the contextual multi-armed bandit problem, the difference being that in our problem the arm can be connected to other arms and the payoffs come from the multiple arms.

Our contribution is three-fold: first, we formalize a new networked bandit problem motivated by real network applications; second, we provide an algorithm based on confidence sets to solve it along with theoretical analysis; and third, we design a set of experiments to test and evaluate the algorithm. To the best of our knowledge, we define and solve this problem for the first time and answer the fundamental question of how to define regret when payoffs come from interrelated multiple arms. We design an effective strategy to select arms in order to increase payoffs over time, known as NetBandits, which provides a solution to this problem. Our approach is an 'optimism in the face of uncertainty'-style algorithm that considers the integrated confidence sets and we prove a regret bound for it. Finally, we analyze empirically the performance, which shows that our algorithm is effective in the networked bandit setting.

## 2. RELATED WORK

The traditional multi-armed bandit problem does not assume that side information is observed. The forecaster's goal is to maximize the sum of payoffs over time based on the historical payoff information. There are two basic settings. In the first, the stochastic setting, the payoffs are i.i.d. drawn from an unknown distribution. The upper confidence bound (UCB) strategy has been used to explore the exploration-exploitation trade-off [4, 6, 16], in which an upper bound estimate is constructed on the mean of each arm at a fixed confidence level, and then the arm with the best estimate is selected. In the second, the adversarial setting, the i.i.d. assumption does not exist. Auer et al. [7] proposed the EXP3 algorithm for the adversarial setting, which was later improved by Bubeck and Audibert [3].

The contextual multi-armed bandit problem is a natural extension of the original bandit problem. Our setting addresses bandit problem with contextual information. Compared to the traditional $K$-armed bandit problem, the forecaster may use action features to infer the payoff in the contextual setting. This problem largely considers the linear model assumption about payoff of action [1, 5, 12, 14, 21]. Auer [5] proposed the LinRel algorithm, a UCB-style algorithm that has a regret of $\tilde{O}(\sqrt{Td})$. Dani et al. [14] stud-

ied the LinRel and provided an $\tilde{O}(d\sqrt{T})$ regret bound and proved this upper bound is tight. Chu et al. [12] provided the LinUCB and SupLinUCB algorithms, and proved an $O(\sqrt{Td\log^3(KT\log(T)/\delta)})$ regret bound for SupLinUCB that holds with probability $1-\delta$. Abbasi-Yadkori et al. [1] proposed an algorithm that modified the UCB-style algorithm based on the confidence sets, and showed a regret of $O(d\log(1/\delta)/\Delta)$.

Recently, the bandit problem has been used in real-life problems, such as recommendation systems and advertising. Li et al. [17] first introduced the bandit problem to recommendation systems by considering a personalized recommendation as a feature-based exploration-exploitation problem. This problem was formalized as a contextual bandit problem with disjoint linear payoffs and by focusing on the article-selection strategy based on user-click feedback, maximizing the total number of clicks. The features of the users and articles were defined as contextual information, and the expected payoff of an arm was assumed to be a linear function of its contextual features, including the user and article information. Finally, the LinUCB algorithm was proposed to solve this problem and attained a good empirical regret. They further extended the algorithm as SupLinUCB and provided the theoretical analysis [12].

There are limited studies that consider the networked bandit problem or that combine bandit problem and network. Buccapatnam et al. [10] considered the bandit problem in social networks, and assumed that the forecaster can take advantage of side observations of neighbors, except for the selected user (arm). The side observations were used to update the sample mean of other related users and the payoff of the selected arm was collected each time, the goal once again being to maximize the total cumulative payoff of selected arms. Bnaya et al. [8] considered a bandit view for network exploration and proposed VUCB1 to handle the dynamic changes in arms when crawling the network. More recently, Cesa-Bianchi et al. [11] considered the recommendation problem by taking advantage of the relationships between users in the network. They proposed GOB.Lin, which models the similarity between users and used this similarity to help predict the behavior of other users.

Our work belongs to the contextual bandit setting. However, in contrast to these previous studies we assume that the arms (actions) are correlated in the network. The selected arm can invoke other related arms and the forecaster obtains multiple payoffs from these arms. It is a more general setting in networked bandit problem.

## 3. NETWORKED BANDITS

We consider a network $G$. Let $\mathcal{V}$ indicate the nodes in the network and $\mathcal{E}$ indicate the edges of the network. We can then use $G=(\mathcal{V},\mathcal{E})$ to represent the networked bandits where $v\in\mathcal{V}$ can be considered as an arm and $e\in\mathcal{E}$ indicates the relationship between arms; nodes here are correlated. Thus, given the network $G$ and a node $v$, it is possible to obtain the information for the node $v$ and its relations $N(v)$.

In our formulation, we consider a sequential decision problem with contextual information. At round $t$, except for contextual information $x_t$, we have a network topology of arms denoted by $G_t$. Given $v$ we let $N_t(v)$ be its relations and $N_t(v)$ may change over time. If $v$ is selected then $N_t(v)$ will also be invoked. We define this setting as networked bandits. Formally, a networked bandit algorithm $\mathcal{A}$ proceeds as follows: at each round $t$, the algorithm observes a set of arms $\mathcal{K}_t=\{1,2,\cdots,k\}_t$, contextual information $x_t$, and the network topology $G_t$ of arms associated with the relationships of arms. The set of relations of arm $a$ is denoted by $N_t(a)$. If we also consider the information of arms, we can redefine the context as a set $\mathcal{C}_t=\{x_{1,t},\cdots,x_{k,t}\}$ by adding arms' information. When the algorithm selects an arm $a_t$, the $a_t$ will invoke other related arms $N_t(a_t)$. $N_t(a_t)$ can be observed based on the network topology of arms. Before the decision algorithm selects the arm, it observes $G_t$, $\mathcal{C}_t$, and $\mathcal{K}_t$. Based on historical payoff records, the algorithm selects an arm $a_t$ and receives a set of payoffs $\{y_{a_t}\}\cup\{y_a|a\in N_t(a_t)\}$. The algorithm will improve the selection strategy after collecting new payoff information. It then proceeds to the next round $t+1$. Note that traditional contextual bandit problems usually assume that the arms are independent. However, in our problem, we assume that the correlation exists between the chosen arm and its relations. After a total $T$ rounds the cumulative payoff is defined as $\sum_{t=1}^{T} g_{a_t,t}$, where $g_{a_t,t}=\sum_{a\in N_t(a_t)} y_a + y_{a_t}$ and $y_a$ is the payoff from arm $a$. For simplicity, we use $N_t(a)$ to indicate both $a$ and its relations. We rewrite $g_{a_t,t}$ as $g_{a_t,t}=\sum_{a\in N_t(a_t)} y_a$.

For this networked bandit problem, the algorithm $\mathcal{A}$ selects an arm $a_t$ at each round $t=1,2,\cdots$ and receives the associate payoff $g_{a_t,t}$. After $n$ selections $a_1,a_2,\cdots,a_n$ we define the regret as follows:

$$\mathcal{R}_n=\max_{a=1,\cdots,k}\sum_{t=1}^{n} g_{a,t}-\sum_{t=1}^{n} g_{a_t,t}. \tag{1}$$

The regret can now be used to compare the best decision with the algorithm $\mathcal{A}$. In this problem, $\mathcal{R}_n$ is a random variable; therefore, the goal is to calculate the expectation of $\mathcal{R}_n$ with high probability, and it is not easy to obtain expectation directly since its search space is large. Normally we try to bound the pseudo-regret, i.e.,

$$\overline{\mathcal{R}}_n=\max_{a=1,\cdots,k}\mathbb{E}\sum_{t=1}^{n} g_{a,t}-\mathbb{E}\sum_{t=1}^{n} g_{a_t,t}, \tag{2}$$

where the pseudo-regret competes against the optimal action in the expectation.

There are two important issues in the networked bandit problem: arms and their network topology. In the context of a social network, the users in the pool may be viewed as arms, the provided message or article as context, and the user's information as additional contextual information. The new context vector then summarizes information of both user and context. A payoff of 1 is incurred when a provided message is 'favorited' or 'liked'; otherwise, the payoff is 0. The network topology of a social network naturally constructs the relationships between users. When a message is posted to a user, the message can be seen by relations (followers). The payoff can be collected from the user's page (selected arm). Furthermore, any 'like', 'share', or 'comment' action by a follower will allow the message to be reposted on the follower's page and to be seen by the follower's friends. The payoff can then be collected from the followers' pages (invoked arms). In the special case that the follower does not repost the message, the payoff can be considered as 0 or the arm is not invoked. For simplicity, we only consider the selected arm and its relations. With these definitions of payoff, arm, and invoked arms, the collected payoff after selecting an arm involves the selected user and

his/her relations. Thus, the payoff at round $t$ is defined as $g_{a_t,t} = \sum_{a \in N_t(a_t)} y_a$.

It is assumed that algorithm $\mathcal{A}$ can observe the network topology prior to make a decision. This is intuitive, since network structure information between users can easily be collected or the network structure information can be obtained in advance. In practice, given an arm, we only need concern itself with the invoked arms, and therefore knowledge of full network topology is unnecessary. The invoked arms depend on how we define $N_t(a)$. The worst case scenario is that the whole network needs to be searched to find the invoked arms and feedback; however, we do not concern how to constrict $N_t(a)$ using such a network propagation model since, as stated above, we only focus on the selection strategy and we simplify the problem by only observing the invoked arms.

# 4. ALGORITHM

In this work, we propose an algorithm to solve the networked bandit problem and show that an integrated confidence bound can efficiently be computed in a closed form when the payoff model of an arm is linear. As with previous contextual bandit work [17], we assume that the expected payoff of an arm $a$ is linear in context $x_t$ and coefficient $w_a$. At round $t$, for arm $a$ given context $x_{a,t}$, we assume that the expected payoff of the arm $a$ is a linear function:

$$\mathbb{E}\left[y_{a,t}|x_{a,t}\right] = x_{a,t}^{\top} w_a + \epsilon_a, \tag{3}$$

where different arms have different $w_a$ and $\epsilon_a$ is conditionally $R$-sub-Gaussian when $R \leq 0$ is a fixed constant. Formally, this means that $\forall\lambda$ and we have

$$\mathbb{E}\left[e^{\lambda \epsilon_{a,t}}|x_{a,1:t}, \epsilon_{a,1:t-1}\right] \leq \exp\left(\frac{\lambda^2 R^2}{2}\right), \tag{4}$$

where $x_{a,1:t}$ denotes the sequence $x_{a,1}, x_{a,2}, \cdots, x_{a,t}$ and, similarly $\epsilon_{a,1:t-1}$ denotes the sequence $\epsilon_{a,1}, \cdots, \epsilon_{a,t-1}$. The arms therefore have disjoint linear payoffs. The decision of the algorithm lies on $w$ with distribution $\epsilon$. Based on our R-sub-Gaussian assumption of the noise, we can obtain meaningful upper bound on the regret. According to this sub-Gaussian condition, we know that $\mathbb{E}\left[\epsilon_{a,t}|x_{a,1:t}, \epsilon_{a,1:t-1}\right] = 0$ and $\mathbb{VAR}[\epsilon_{a,t}|x_{a,1:t}, \epsilon_{a,1:t-1}] \leq R^2$. The conditions therefore show that $\epsilon_{a,t}$ is bounded by a zero-mean noise lying in an interval of length of at most $2R$.

As the networked bandit problem, the algorithm faces a set of uncertainties of arms which involve $N_t(a)$. We design a new algorithm which is the 'optimism in the face of uncertainty' principle, by maintaining confidence of parameter $w$ for each arm. The basic idea is to construct the confidence sets for parameters of each disjoint payoff function and then provide an integrated upper bound.

We use technology from the 'self-normalized bound for vector-valued martingales' [19] and confidence sets [1]. For each arm $\hat{w}_a$ is defined as the $L^2$-regularized least-squares estimate of $w_a^*$ with regularization parameter $\lambda > 0$:

$$\hat{w}_a = (X_a^{\top} X_a + \lambda)^{-1} X_a^{\top} Y_a, \tag{5}$$

where $X_a$ is the matrix whose rows are $x_1, \cdots, x_{n_a(t)}$ corresponding to historical contexts of an arm $a$ and $Y_a \in \mathbb{R}^{n_a(t)}$ is the corresponding historical payoff vector. For a positive definite self-adjoint operator $V$, we define $\|x\|_V = \sqrt{\langle x, Vx\rangle}$ as the weighted norm of vector $x$. It can be proved that $\hat{w}$ lies with high probability in an ellipsoid centered at $w^*$ as follows:

THEOREM 1. [1, 19] *According to the 'self-normalized bound for vector-valued martingales', let $V = \lambda I, \lambda > 0$, and $\overline{V}_t = V + \sum_{n=1}^{t-1} x_n \otimes x_n$ be the regularized design matrix underlying the covariates. Define $y_t = x_t^{\top} w^* + \epsilon_t$ and assume that $\|w^*\|_2 \leq S$. Then, for any $0 < \delta < 1$, with probability at least $1-\delta$, for all $t \geq 1$ we can bound $w^*$ in such a confidence set:*

$$C_t = \Bigg\{ w^* \in \mathbb{R}^d : \|\hat{w}_t - w^*\|_{\overline{V}_t} \leq R\sqrt{2\log\left(\frac{|\overline{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S \Bigg\}. \tag{6}$$

*In addition, if $\|x_t\| \leq L$ then with probability at least $1-\delta$, for all $t \geq 1$, we can bound $w^*$ in a new confidence set:*

$$C_t' = \Bigg\{ w^* \in \mathbb{R}^d : \|\hat{w}_t - w^*\|_{\overline{V}_t} \leq R\sqrt{d\log\left(\frac{1 + tL^2/\lambda}{\delta}\right)} + \lambda^{1/2} S \Bigg\}. \tag{7}$$

The above bound provides the confidence region at time $t$. It shows that with good choice of the right parts of the equation, $w^*$ always remains inside this ellipsoid for all times $t$ with probability $1-\delta$. Next, we show the bound of the arm with a single linear payoff.

THEOREM 2. *Let $(x_1, y_1), \cdots, (x_{t-1}, y_{t-1})$, $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$ satisfy the linear model assumption. Furthermore, we have the same assumption as Theorem 1. Then, for any $0 < \delta < 1$, with probability at least $1-\delta$, for all $t \geq 1$ we can have:*

$$\|x^{\top}\hat{w} - x^{\top}w^*\| \leq \|x\|_{\overline{V}_t^{-1}} \left(R\sqrt{2\log\left(\frac{|\overline{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S\right). \tag{8}$$

*In addition, if $\|x_t\| \leq L$ then for all $t \geq 1$, with probability $1-\delta$ we can have:*

$$\|x^{\top}\hat{w} - x^{\top}w^*\| \leq \|x\|_{\overline{V}_t^{-1}} \left(R\sqrt{d\log\left(\frac{1 + tL^2/\lambda}{\delta}\right)} + \lambda^{1/2} S\right). \tag{9}$$

PROOF.

$$\|x^{\top}\hat{w} - x^{\top}w^*\| = \|x^{\top}(\hat{w} - w^*)\| \leq \|x\|\|\hat{w} - w^*\| = \|x\|_{\overline{V}_t^{-1}}\|\hat{w} - w^*\|_{\overline{V}_t}.$$

According to (6), with probability at least $1-\delta$, for all $t \geq 1$, we have:

$$\|x^{\top}\hat{w} - x^{\top}w^*\| \leq \|x\|_{\overline{V}_t^{-1}} \left(R\sqrt{2\log\left(\frac{|\overline{V}_t|^{1/2}|\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S\right).$$

According to (7), with probability at least $1-\delta$, for all $t \geq 1$ we have:

$$\|x^\top \hat{w} - x^\top w^*\| \leq \|x\|_{\overline{V}_t^{-1}} \left( R\sqrt{d \log\left(\frac{1+tL^2/\lambda}{\delta}\right)} + \lambda^{1/2} S \right).$$

□

LEMMA 1. *Given an arm $a \in \mathcal{K}_t$ with the context feature $x$, let $(x_{a,1}, y_{a,1}), (x_{a,2}, y_{a,2}), \cdots, (x_{a,n_a(t-1)}, y_{a,n_a(t-1)})$ be history records of arm $a$ before $t$ and $x_a \in X_a$ and $y_a \in Y_a$, and let $\hat{w}_a = (X_a^\top X_a + \lambda I)^{-1} X_a^\top Y_a$. We have*

$$x^\top w_a^* \leq x^\top \hat{w}_a + \|x\|_{\overline{V}_t^{-1}} \left( R\sqrt{2 \log\left(\frac{|\overline{V}_t|^{1/2} |\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S \right). \quad (10)$$

As shown in (10), we have a possible upper bound of $x^\top w_a^*$, which has two parts. The first term can be deemed as empirical expected estimation of payoff of the arm, and the second term can be considered as a penalty. This penalty is typically a high probability upper confidence bound on the payoff of the arm.

Thus, given an arm $a$ and its relations $N_t(a)$, we face the exploitation-exploration problem. We use the integrated confidence bound on the payoffs of these invoked arms.

LEMMA 2. *In the networked bandits, given an arm $a \in \mathcal{K}_t$ and the network relationship $N_t(a)$, we obtain:*

$$\sum_{a \in N_t(a)} x_a^\top w_a^* \leq \sum_{a \in N_t(a)} x_a^\top \hat{w}_a + \sum_{a \in N_t(a)} \|x_a\|_{\overline{V}_t^{-1}} \left( R\sqrt{2 \log\left(\frac{|\overline{V}_t|^{1/2} |\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S \right). \quad (11)$$

We believe that the confidence bound can be successfully applied to this situation with the exploitation-exploration trade-off. We use the confidence bound generated by the confidence sets of parameters, defined by:

$$B_{a,t} = \nu_{a,t} + \xi_a(t), \quad (12)$$

$\nu_{a,t} = \sum_{a \in N_t(a)} x^\top \hat{w}_{a,t}$ indicates the expected value, and $\xi_a(t)$ is the last term of (11) and indicates the penalty of the estimation. Figure 2 shows the upper bound of arms from our illustrative example. Each arm has the empirical payoff and a potential value. Thus, in each round, our algorithm selects an arm based on the estimation from the confidence bound, such that the predicted payoff is maximized. Our algorithm is shown in Algorithm 1.

## 5. REGRET ANALYSIS

We next provide a bound on the regret of our algorithm when run through the confidence sets constructed in Theorem 1. We assume that the expected estimation of payoffis bounded. We can view this as a bound on parameters and the bound on the arms set. We state a bound on the regret of the algorithm as follows:

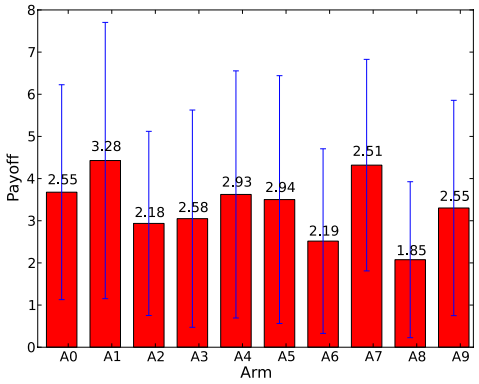



Figure 2: An example of the upper bound $B$ in 10-arm networked bandits when $t = 120$. Bar denotes the payo ff estimation and vertical line denotes the penalty of the estimation.

**Algorithm 1** NetBandits

**Input:** $\mathcal{K}_t$, $G_t$, $\mathcal{C}_t$, $t = 1, \cdots, T$

1: **for** round $t = 1, 2, \cdots, T$ **do**
2: For each arm we can observe the features $x_{a,t}, a \in \mathcal{K}_t$, and the invoked arms $N_t(a)$ based on $G_t$
3: **for** each $a \in \mathcal{K}_t$ **do**
4: Compute $\hat{w}_a$ according to (5)
5: Compute the quality

$$B_{a,t} = \sum_{a \in N_t(a)} x_{a,t}^\top \hat{w}_a + \sum_{a \in N_t(a)} \|x_{a,t}\|_{\overline{V}_t^{-1}} \left( R\sqrt{2 \log\left(\frac{|\overline{V}_t|^{1/2} |\lambda I|^{-1/2}}{\delta}\right)} + \lambda^{1/2} S \right)$$

6: **end for**
7: Choose arm $a_t = \arg\max_{a \in \mathcal{K}_t} B_{a,t}$
8: Observe the multiple payoffs $\{y_{a,t} | a \in N_t(a_t)\}$
9: **for** each node $a \in N_t(a_t)$ **do**
10: Update $X_a, Y_a$
11: **end for**
12: **end for**

THEOREM 3. *On the networked bandits, assume that each arm's payoff function satisfies the linear model, and assume that the contextual vector is $x_{a,t}$ for each arm $a \in \mathcal{K}_t, |\mathcal{K}_t| \leq K$ and $t = 1, \cdots, T$. Then, for any $0 < \delta < 1$, with probability at least $1 - \delta$, the cumulative regret satisfies*

$$\overline{\mathcal{R}}_T \leq 2K\sqrt{2\beta_T(\delta) T \log|I + XX^\top/\lambda|}, \quad (13)$$

*where*

$$\beta_T(\delta) = \left( R\sqrt{2 \log\left(\frac{|I + XX^\top/\lambda|^{1/2}}{\delta}\right)} + \lambda^{1/2} S \right)^2.$$

PROOF. Considering the instantaneous regret at round $t$, we select an optimal arm according to our algorithm. Thus, we have optimistic $(a_t, \tilde{w}_a)$ and $\hat{w}_a$ for the $N(a_t)$. For round

$t$, we rely on [1] to have:

$$\begin{aligned}
r_t &= \sum_{a\in N(a_t)} x_{a,t}^\top w_a^* - \sum_{a\in N(a^*)} x_{a,t}^\top w_a^* \\
&\le \sum_{a\in N(a_t)} x_{a,t}^\top w_a^* - \sum_{a\in N(a_t)} x_{a,t}^\top \tilde{w}_a \\
&= \sum_{a\in N(a_t)} x_{a,t}^\top w_a^* - \sum_{a\in N(a_t)} x_{a,t}^\top \hat{w}_a \\
&+ \sum_{a\in N(a_t)} x_{a,t}^\top \hat{w}_a - \sum_{a\in N(a_t)} x_{a,t}^\top \tilde{w}_a \\
&= \sum_{a\in N(a_t)} x_{a,t}^\top (w_a^* - \hat{w}_a) + \sum_{a\in N(a_t)} x_{a,t}^\top (\hat{w}_a - \tilde{w}_a) \\
&= \sum_{a\in N(a_t)} \|x_t\|_{\overline{V}_t^{-1}} \|w^* - \hat{w}_t\|_{\overline{V}_t^{-1}} \\
&+ \sum_{a\in N(a_t)} \|x_t\|_{\overline{V}_t^{-1}} \|\hat{w}_t - \tilde{w}_t\|_{\overline{V}_t^{-1}} \\
&\le \sum_{a\in N(a_t)} 2\sqrt{\beta_t(\delta)} \|x_t\|_{\overline{V}_t^{-1}}. \qquad (14)
\end{aligned}$$

For each arm $a \in N_t(a)$, we define

$$r_{a,t} = x_{a,t}^\top w_a^* - x_{a,t}^\top \hat{w}_a + x_{a,t}^\top \hat{w}_a - x_{a,t}^\top \tilde{w}_a, \qquad (15)$$

and we have

$$r_{a,t} \le 2\sqrt{\beta_t(\delta)} \|x_{a,t}\|_{\overline{V}_t^{-1}}. \qquad (16)$$

We then rewrite the instantaneous regret (14) as

$$r_t \le \sum_{a\in N(a_t)} r_{a,t}. \qquad (17)$$

Regarding to the fact that $r_{a,t} \le 2$, we have

$$\begin{aligned}
r_{a,t} &\le 2\min\left(\sqrt{\beta_t(\delta)} \|x_{a,t}\|^2_{\overline{V}_t^{-1}}, 1\right) \\
&\le 2\sqrt{\beta_t(\delta)} \min\left(\|x_{a,t}\|^2_{\overline{V}_t^{-1}}, 1\right). \qquad (18)
\end{aligned}$$

Given arms $N_t(a)$, we define $a^{'} = \arg\max_a \|x_{a,t}\|^2_{\overline{V}_t^{-1}}$. Then we have

$$\begin{aligned}
r_t &\le |N_t(a)| 2\sqrt{\beta_t(\delta)} \|x_{a',t}\|_{\overline{V}_t^{-1}} \\
&\le |N_t(a)| 2\sqrt{\beta_t(\delta)} \min\left(\|x_{a',t}\|^2_{\overline{V}_t^{-1}}, 1\right). \qquad (19)
\end{aligned}$$

Thus, with probability at least $1-\delta$, for any $T \ge 1$,

$$\begin{aligned}
\overline{\mathcal{R}}_T &= \sum_{t=1}^T r_t \le \sum_{t=1}^T |N_t(a)| 2\sqrt{\beta_t(\delta)} \left(\|x_{a',t}\|_{\overline{V}_t^{-1}} \wedge 1\right) \\
&\le 2K \sum_{t=1}^T \sqrt{\beta_t(\delta)} \left(\|x_{a',t}\|_{\overline{V}_t^{-1}} \wedge 1\right) \\
&\le 2K \sqrt{T \sum_{t=1}^T \left(\sqrt{\beta_t(\delta)} \left(\|x_{a',t}\|_{\overline{V}_t^{-1}} \wedge 1\right)\right)^2} \\
&\le 2K \sqrt{T\beta_T(\delta) \sum_{t=1}^T \left(\|x_{a,t}\|^2_{\overline{V}_t^{-1}} \wedge 1\right)}. \qquad (20)
\end{aligned}$$

According to $\log(1+z) \le z$, we have:

$$\log|I + XV^{-1}X| \le \sum_{k=1}^{t-1} \|x_k\|^2_{\overline{V}_k^{-1}}. \qquad (21)$$

Then according to $z \le 2\log(1+z), z \in [0,1]$, we have

$$\sum_{k=1}^{t-1} \left(\|x_k\|^2_{\overline{V}_k^{-1}} \wedge 1\right) \le 2\log|I + XV^{-1}X|. \qquad (22)$$

We choose $V = \lambda I$, then we rewrite $\overline{\mathcal{R}}_T$ as

$$\overline{\mathcal{R}}_T \le 2K\sqrt{2T\beta_T(\delta)\log|I + XX^\top/\lambda|}. \qquad (23)$$

□

LEMMA 3. *Assume that $x \in \mathbb{R}^d$ and $V = \lambda I$. Then, for any $0 < \delta < 1$, with probability at least $1-\delta$, the bound is*

$$\begin{aligned}
\overline{\mathcal{R}}_T &\le 2K\sqrt{2Td\log\left(\lambda + \frac{(T-1)L}{d}\right)} \\
&\cdot \left(\lambda^{1/2}S + R\sqrt{2\log\left(\frac{1}{\delta}\right) + d\log\left(1 + \frac{(T-1)L}{\lambda d}\right)}\right). \qquad (24)
\end{aligned}$$

We are mainly interested in the interrelated arms. Our regret bound depends on the number of invoked arms $|N_t(a)|$ or loose $K$. Figure 3 shows experimentation of our bound applied to the networked bandit problem. Our algorithm keeps the regret as low as possible, and can reach $\overline{\mathcal{R}}_t/t \to 0$ with high probability when $t$ is large enough.

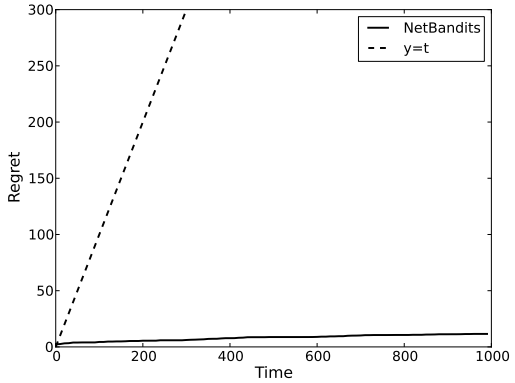



Figure 3: An example of the regret value in 10-arm networked bandits. The experiments are repeated 100 times and the average regrets are shown. $y = x$ is provided for comparison.

# 6. PRACTICAL ISSUES

In real-world applications, according to different assumptions about the network topology of arms, we can consider special cases of the networked bandit problem. We focus on $N_t(a)$. In our algorithm, we make the very loose assumption that $N_t(a)$ varies over time. However, the network topology is sometimes stable over a fixed duration. For example, a school social network is stable for the duration of a semester. This means that $N_t(a) = N_0(a)$, which is a special case of network bandits. In other cases, for example when inquiring users in the same company, we only need to consider their colleagues or a group of interest.

## 6.1 Dynamic network

In the networked bandit problem the network topology of arms is usually dynamic over time, which means for each

round $t$ we have different $N_t(a)$. Although we assume that $N_t(a)$ will be active after the forecaster selects $a$, we omit how to generate $N_t(a)$ and how arm $a$ invokes $N_t(a)$, which are not our primary concerns. Instead, we simplify our problem using the simple setting of selecting an arm and then receiving payoffs from invoked arms. We assume that we can observe invoked arms $N_t(a)$. In practice, we can directly obtain $N_t(a)$ by predefining the arms, for example as neighbors or groups, or by observing feedback and collecting arms which provide feedback. We focus on how to select arms in order to maximize the total payoff, and therefore we concern the arms of $N_t(a)$ and the forecaster can obtain the invoked bandits over the course of collecting the payoffs from the network. In particular, at each round the algorithm observes the network topology of arms; it then decides which arm to select using the knowledge of the network topology and historical payoffinformation.

In Algorithm 2, we provide a pseudo-code for the selection at each round in a dynamic network.

**Algorithm 2** Selection at round $t$ in dynamic network

1: For each arm we have $\hat{w}_a$ and observer the context $x_{a,t}$
2: For each arm we collect $N_t(a)$
3: **for** each $a \in \mathcal{K}_t$ **do**
4: Compute $B_{a,t}$
5: **end for**
6: Select arm $a_t = \arg\max_{a \in \mathcal{K}_t} B_{a,t}$
7: Observe the payoffs $\{y_{a,t} | a \in N_t(a_t)\}$ from the network
8: For each arm $a \in N_t(a_t)$ update $X_a$, $Y_a$ and $\hat{w}_a$

## 6.2 Static network

We make the simple assumption that the network topology is fixed. In other words, the relationships between arms do not change. For all $t$, we have $G_t = G_0$, $\mathcal{K}_t = \mathcal{K}_0$ and $N_t(a) = N_0(a)$. For example, DBLP, Last.FM, and many offline social network datasets are of fixed duration. This is a degenerate version of our problem and can be solved using our algorithm.

In Algorithm 3, we provide a pseudo-code for the selection at each round in a static network.

**Algorithm 3** Selection at round $t$ in static network

1: For each arm we have $\hat{w}_a$ and $N_0(a)$, and observer the context $x_{a,t}$
2: **for** $a \in \mathcal{K}_0$ **do**
3: Compute $B_{a,t}$
4: **end for**
5: Select arm $a_t = \arg\max_{a \in \mathcal{K}_0} B_{a,t}$
6: Observe the payoffs $\{y_{a,t} | a \in N_0(a_t)\}$ from the network
7: For each arm $a \in N_0(a_t)$ update $X_a$, $Y_a$ and $\hat{w}_a$

## 6.3 Neighborhood or group

In networks, especially in social networks, a simple yet common assumption is that the node largely influences its neighborhoods or community [13, 15]. Moreover, some applications only focus on people who have the same interest or are in a group. This makes it possible to assume that the selected arm only invokes its neighbors or a group; that is, $N_t(a) = Neig_t(a)$, where $Neig_t(a)$ indicates the neighbors of $a$. We can only collect the payoffs of neighbors of an arm, and therefore $N_t(a)$ is appropriate. Although there are also two cases in this situation - static and dynamic - here we focus only on the neighborhood.

In Algorithm 4, we provide a pseudo-code for the selection at each round with specific $N_t(a)$.

**Algorithm 4** Selection at round $t$ with neighborhood

1: For each arm we have $\hat{w}_a$ and observer the context $x_{a,t}$
2: For each arm we collect $Neig_t(a)$
3: **for** $a \in \mathcal{K}_t$ **do**
4: Compute $B_{a,t}$
5: **end for**
6: Select arm $a_t = \arg\max_{a \in \mathcal{K}_t} B_{a,t}$
7: Observe the payoffs $\{y_{a,t} | a \in Neig_t(a_t)\}$ from the network
8: For each arm $a \in Neig_t(a_t)$ update $X_a$, $Y_a$ and $\hat{w}_a$

# 7. EXPERIMENTS

## 7.1 Illustrative Example

We first illustrate our model by a synthetic example (Figure 4), which contains 10 arms (A0-A9) randomly connected at each round. At different rounds, the networks are different. At rounds $t = 11$ and $t = 20$, the upper bound $B$ (the second row, blue) is large; however, the expected estimation is small (the third row, red) because the variance is large. Our algorithm selects the arm with maximal upper bound. We also show the real payoffs of all the arms, which are not known to the algorithm. At an early stage, the selection is poor compared to the real payoff (the fourth row, green). At round $t = 20$, our algorithm chooses A0 however, the best is A1, illustrating that the expected estimation is small and the algorithm can try another arm that has potential, but with uncertainty. Later, selection becomes efficient and at $t = 120$ the algorithm chooses A1 since more information has been learned and the upper bound becomes more stable with lower penalty. This is close to the real situation and provides a good estimation.

## 7.2 Baselines and Performance Metric

In this section we evaluate the proposed NetBandits strategy on four synthetic datasets and two public real-world datasets. We perform two types of experiments: simulation experiments and offline evaluation of two real applications.

We compare our proposed method against two baselines: a state-of-art algorithm for the contextual bandit problem, referred to as TraBandits, and the random strategy. Since there is no existing method for the networked bandit problem, these methods are little altered for networked bandits. The details are follows:

- **TraBandits:** a state-of-art method for contextual bandits with linear payoff models [17]. The algorithm is a UCB style method with the linear payoff assumption that always selects the arm using highest UCB at each round.
- **Random:** a simple strategy that just randomly selects an arm.

We use two methods to assess the performance of our algorithm. We first analyze the average payoff at each round,

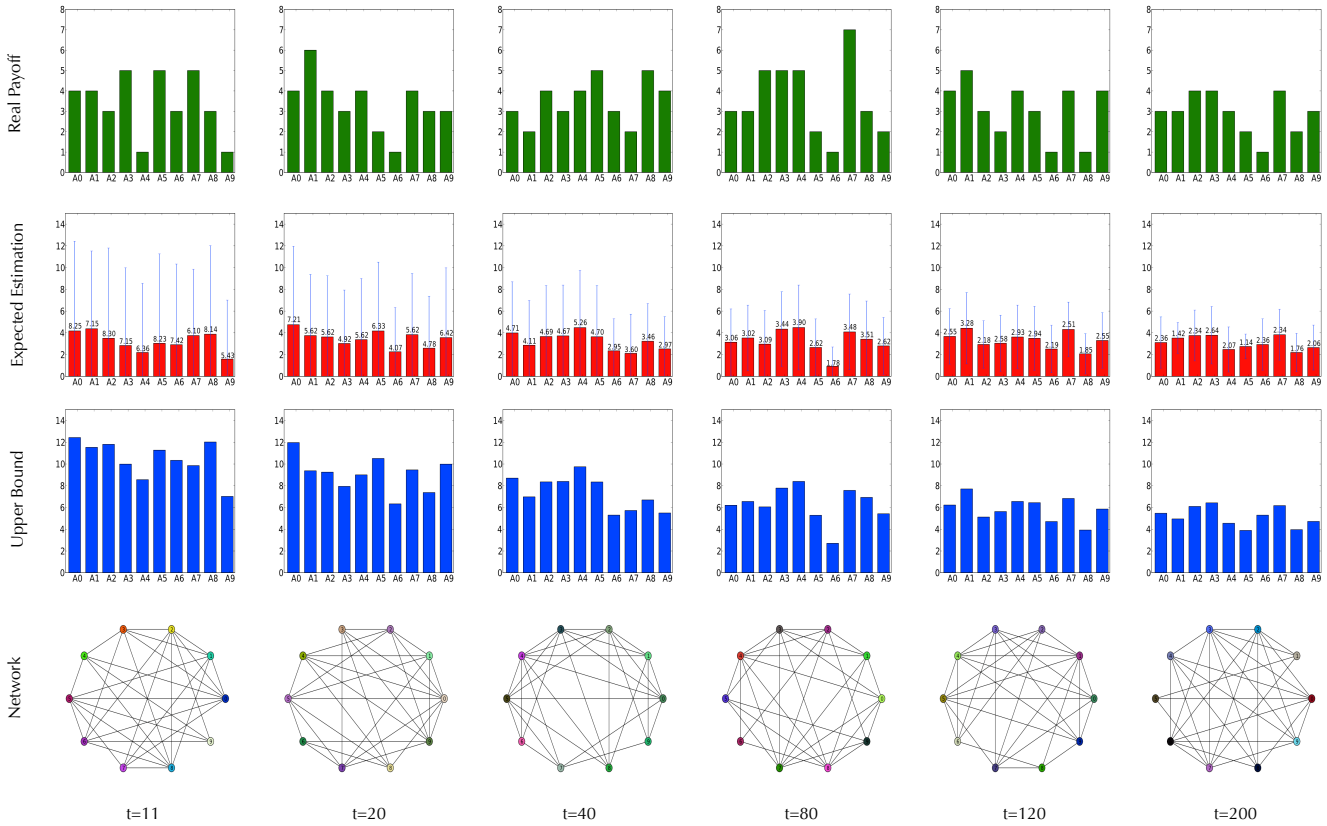



Figure 4: Illustrative synthetic example of exploration-exploitation trade-off. Bottom, arms with networked topology. Second row: the upper bound $B$ for each arm computed using NetBandits. Third row: the expected estimation $\nu$, where bar denotes the estimation and vertical line denotes the penalty of estimation. Fourth row: the real payoff of each arm.

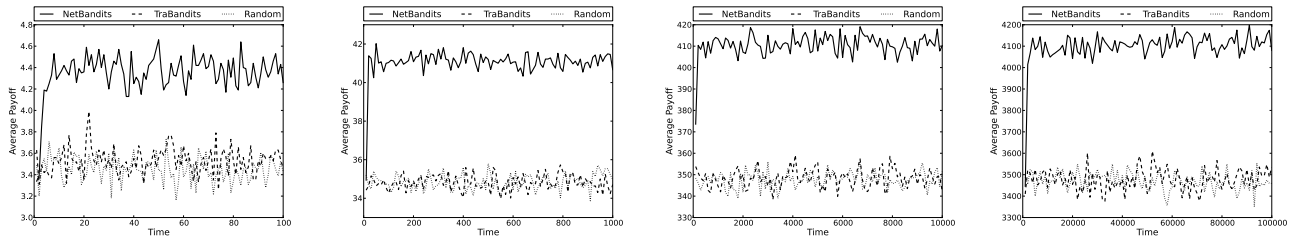
and then we analyze the cumulative payoff at each round, which ignores the performance of the algorithm at each fixed round but gives an overall view of the lifetime performance of the algorithm.

## 7.3 Simulation Experiments

We test our algorithm on a series of synthetic datasets. In contrast to previous work, we need to construct the network topology, which can be either static or dynamic. Static network is a special case of dynamic situation and static network is generated in advance and remains unchanged. We therefore construct the networked bandits based on the dynamic network as follows: we first construct a fixed number of nodes $k$, which are considered as arms. We then randomly create edges between them, which are used to generate the relations for each arm. Neighborhood is considered as relationship. For every node, we assign different norm random vector $u_i, u_i \in \mathbb{R}^{10}$ and we use the following stochastic model to generate its payoffs: $y_a(x) = x^\top u_a + \epsilon_a$, where $\epsilon_a$ is uniformly distributed in a bounded interval centered around zero and $u_a$ and $\epsilon_a$ are not known to the decision algorithm. For contextual information, at each round $t$ we randomly create a set of context vectors $\{x_{1,t}, \cdots, x_{k,t}\}, x_{k,t} \in \mathbb{R}^{10}$. The network topology does not have strict assumptions and is created simply: in the dynamic situation, we generate the network topology at each round (relationships between the nodes change at each round). We randomly create $k^2/3$ edges between the nodes, and therefore for most nodes the relations will be no greater than $k/3$.

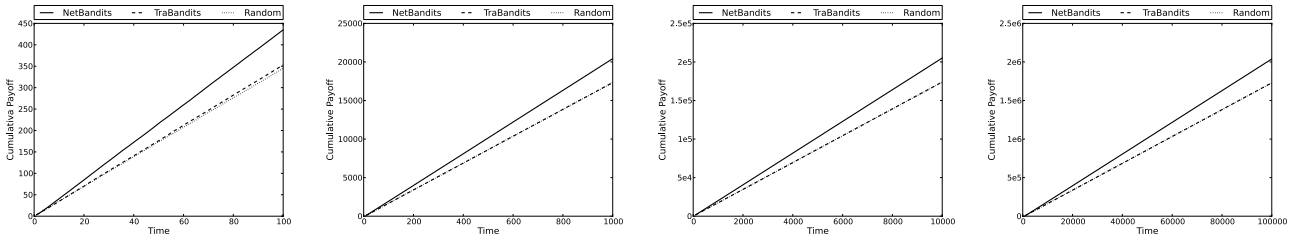
We present the results from $k$ =10, 100, 1000, and 10000 arms with dynamic network topology. In Figure 5 and Figure 6 we present the results of average payoff and cumulative payoff; our NetBandits outperforms the other baselines. TraBandits does not work well, indicating that best single arm does not always have the best payoff in a network but also depends on its relations. As per the network construction, the average payoff for each node is around $k/3$ if the node and its relations provide feedback, and the average payoff of TraBandits and Random is around $k/3$. For example, as shown in Figure 5, when $k = 10$ the payoff ranges from 3.2 to 3.6; when $k = 100$ the payoff ranges from 34 to 35; when $k = 1000$ the payoff ranges from 340 to 350; when $k = 10000$ the payoff ranges from 3450 to 3550. However, NetBandits usually performs better except the earliest time points, and its value is greater than 4.2 when $k = 10$, 40 when $k = 100$, 400 when $k = 1000$, and 4500 when $k = 10000$. This is because NetBandits performs more exploration than exploitation to begin with. Figure 6 shows that our algorithm obtains the best cumulative payoff over all rounds. As average payoff improves, NetBandits also exhibits higher cumulative payoff. This indicates that more early exploration improves later selections, leading to a fairer assessment of the performance of the different algorithms.

The running time of NetBandits according to different numbers of arms and network topology is also shown in Table 1, and demonstrates the running time increases rapidly as the number of the scale of the networks increase. For example $k = 100$ is slower than $k = 10$ by more than $k^2$ but



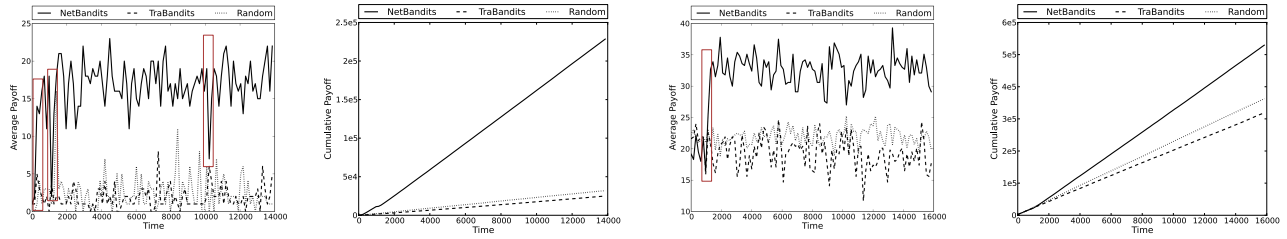

(a) *k=10* (b) *k=100* (c) *k=1000* (d) *k=10000*

Figure 5: The average payoff at each round in dynamic networks.




(a) *k=10* (b) *k=100* (c) *k=1000* (d) *k=10000*

Figure 6: The cumulative payoff at each round in dynamic networks.




(a) Average payoff(Del) (b) Cumulative payoff(Del) (c) Average payoff(LFM) (d) Cumulative payoff(LFM)

Figure 7: The average payoff and cumulative payoff for two real-world datasets.

| Arms | 10 | 100 | 1,000 | 10,000 |
|---|---|---|---|---|
| Avg of invoked arms | 3 | 33 | 333 | 3333 |
| Total round | 100 | 1,000 | 10,000 | 100,000 |
| Time (second) | 0.1 | 23.4 | 2034.2 | 173,628.3 |

Table 1: Running time results of NetBandits on four synthetic datasets.

less than $k^3$. The time taken depends on the size of the network, including the number of nodes and edges. The time complexity of NetBandits is $O(TKN\Omega)$, where $T$ is the total number of rounds, $K$ is the number of arms, $N$ indicates the average number of invoked arms, and $\Omega$ indicates the time taken to compute the parameters; it is no more than $O(TK^2\Omega)$ where $N = K$. It can be improved by calculating each arm in parallel for a large number of arms.

## 7.4 Real-world Datasets Experiments

We also test our algorithm on two publicly available real-world datasets[1]: Delicious Bookmarks, a dynamic dataset, denoted by Del; and Last.FM, a static dataset, denoted by LFM.

Delicious Bookmarks is a social network for storing, sharing, and discovering web bookmarks. The Del dataset contains 1,861 nodes and 7,668 edges and 69,226 URLs described by 53,388 tags. Payoffs are created using the information about the bookmarked URLs for each user: the payoff is 1 if the user bookmarked the URL, otherwise the payoff is 0. Pre-processing is performed by breaking the tags down into smaller fragile items made up of single words, ignoring the underscores, hyphens, and dashes. Each word is represented using the TF-IDF context vector based on the words of all tags, i.e., these feature vectors are the context vectors. PCA was performed on the dataset and the first 16 principle components selected as context vectors building a linear function based on payoff records for each user. This linear function generates a payoff when given a new context. At each round $t$, we provide $x_{k,t} \in \mathbb{R}^{16}$ for all users $k$.

The Last.FM dataset is a music website that builds a detailed profile of each user's musical taste by recording details of the tracks that the user listened to from a range of digital devices. LFM contains 1,892 nodes and 12,717 edges and has 17,632 artists described by 11,946 tags. We use the listened-to artists information to construct payoffs: if the user listened to an artist at least once the payoff is 1, otherwise the payoff is 0. Similar pre-processing is performed as Delicious Bookmarks. Compound tags are broken down into several corresponding single words resulting in 6,036 words. We represent context features using the TF-IDF features, and after PCA the first 16 principle components are selected as context vectors. For each user we then build a linear function based on payoff records. This linear function

[1]http://grouplens.org/node/462

can generate a payoff when given a new context. At each round, we provide $x_{k,t} \in \mathbb{R}^{16}$ for all users $k$.

We construct the network topology according to the social network of the users. Neighborhood is considered as relationship. The linear payoff function for each user is learned in advance and unknown to the algorithm, which decides its next selection according to previous feedback.

For the Del dataset, there exists the timestamp information that records when contact relationships were created, and we can therefore construct a dynamic network according to the timestamps. Timestamps are from 1146752335000 to 1288104100000; we therefore divide them into 14 groups according to the first three numbers (114, 115, ..., 128). We set the total rounds $T = 14000$ and update the network every 1000 rounds. For the Last.FM dataset, there is no time information, thus we construct a static network.

The results of average payoff and cumulative payoffare shown in Figure 7. Our algorithm outperforms the other baselines. Although the two networks have a similar number of users, LFM has more relationships and the average and cumulative payoff results are higher than Del. For the average payoff of Del, there exist three low intervals marked by (red) rectangles in Figure 7(a). These occurred at the beginning and close to round $t = 1000$ and $t = 10000$. Since many new nodes and edges are added at these rounds and NetBandits performs more exploration than exploitation. The payoffs improve after exploration. For the average payoffresults of LFM, there is a low interval at the start, denoted by the (red) rectangle in Figure 7(c), because NetBandits is trying to select possible better arms and perform exploration; then later the performance improves. Figure 7(b)(d) show that the cumulative payoff results of NetBandits increase faster, and are much greater than the other algorithms, and demonstrate that exploration does not hurt the total performance.

# 8. CONCLUSION AND FUTURE WORK

In this paper we formalize a new bandit problem, termed networked bandits. We presented the novel problem of how to select the arm with multiple payoffs in networked bandits by considering a multi-armed bandit of interconnected arms, one of which can invoke other related arms at each round. After selecting an arm, we can obtain payoffs from this arm and its relations.

We consider this approach in the contextual bandit setting and assume disjoint linear payoffs for arms. We propose a new networked bandit algorithm NetBandits that considers the uncertainty of the payoffs using integrated confidence sets. We also provide a regret bound for our solution. Our experiments show that it is better to consider both the network topology and the payoffs of arms, and we observe that our approach performs well in this setting.

The networked bandit problem requires further work. Some interesting problems still remain, such as how to model $N_t(a)$. In our work we do not make any assumption about the structure of the network topology; for example the hub may have higher priority, and it is possible to find a more efficient method for some fixed structures.

Another problem is arm complexity. We assume that one arm invokes other arms, which in turn can invoke other arms sequentially, with processing occurring at the same time. However in some real applications, the structure is possible to be much more complex and evolve over time, which is likely to delay the payoffs.

# 9. ACKNOWLEDGMENTS

This work is supported by Australian Research Council Projects FT-130101457 and DP-140102164.

# 10. REFERENCES

[1] Y. Abbasi-Yadkori, C. Szepesvári, and D. Tax. Improved algorithms for linear stochastic bandits. In *NIPS*, pages 2312–2320, 2011.

[2] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD*, pages 7–15. ACM, 2008.

[3] J.-Y. Audibert and S. Bubeck. Regret bounds and minimax policies under partial monitoring. *JMLR*, 9999:2785–2836, 2010.

[4] J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theor. Comput. Sci.*, 410(19):1876–1902, 2009.

[5] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *JMLR*, 3:397–422, 2003.

[6] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[7] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J COMPUT*, 32(1):48–77, 2002.

[8] Z. Bnaya, R. Puzis, R. Stern, and A. Felner. Social network search as a volatile multi-armed bandit problem. *HUMAN*, 2(2):pp–84, 2013.

[9] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[10] S. Buccapatnam, A. Eryilmaz, and N. B. Shroff. Multi-armed bandits in the presence of side observations in social networks. *OSU, Tech. Rep*, 2013.

[11] N. Cesa-Bianchi, C. Gentile, and G. Zappella. A gang of bandits. In *NIPS*, 2013.

[12] W. Chu, L. Li, L. Reyzin, and R. E. Schapire. Contextual bandits with linear payoff functions. In *AISTAS*, pages 208–214, 2011.

[13] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *KDD*, pages 160–168. ACM, 2008.

[14] V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, pages 355–366, 2008.

[15] T. Iwata, A. Shah, and Z. Ghahramani. Discovering latent influence in online social activities via shared cascade poisson processes. In *KDD*, pages 266–274. ACM, 2013.

[16] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *ADV APPL PROBAB*, 6(1):4–22, 1985.

[17] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670. ACM, 2010.

[18] S. A. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *KDD*, pages 33–41. ACM, 2012.

[19] V. H. Peña, T. L. Lai, and Q.-M. Shao. *Self-normalized processes: Limit theory and Statistical Applications*. Springer, 2008.

[20] H. Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.

[21] P. Rusmevichientong and J. N. Tsitsiklis. Linearly parameterized bandits. *MATH OPER RES*, 35(2):395–411, 2010.

[22] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, pages 807–816. ACM, 2009.