

輪講資料 04/11/4 @ IDEON Fall Retreat

Michal Feldman, Kevin Lai, Ion Stoica and John Chuang,
“Robust Incentive Techniques for Peer-to-Peer Networks”, EC 2004.

資料作成: 門林 雄基 (奈良先端科学技術大学院大学)

Abstract (直訳)

free riding は今日の P2P システムが直面する重要な問題だ。P2P 独特の制約のために問題が難しくなっている: ノード数の多さ、入れ替わりの激しさ、興味の非対称性、結託、コストのかからない ID、裏切り。この問題を解決するために P2P システムを Generalized Prisoner’s Dilemma (GPD, 囚人のジレンマの一般化) でモデル化し、返礼型の決定関数 (reciprocatative decision function) を一連のインセンティブ手法の基盤として提案する。これら一連の手法は完全に分散型である: discriminating server selection, maxflow-based subjective reputation, adaptive stranger policies. シミュレーションを通して、これらの手法を使うと利己的な利用者を最適な協調に駆り立てられることを示す。

1. Introduction

P2P, wireless ad-hoc network, online auction 等の分散システムでは元来、協調するためのインセンティブが働かない。結果としてユーザは自らの利便性を最大化することを試み、システム全体の利便性を下げてしまう。

この問題に対し、ゲーム理論的アプローチを採用する。特に、囚人のジレンマモデルを用いて個人の利便性と全体の利便性の緊張関係をとらえ、非対称な payoff matrix を用いて peer 間の非対称な取引を取り扱えるようにする。また学習にもとづく population dynamic model で個々の peer のふるまいを規定する。

social dilemma については広範囲に研究がなされているが、P2P アプリケーションには独特の制約がある:

- ノード数の多さ、入れ替わりの激しさ: 100,000 同時ユーザ、平均セッション寿命が数分。
- 興味の非対称性: Figure 1 参照。
- コストのかからない ID: 多くの P2P システムでは ID を切り替えることができってしまう。whitewash と呼ばれる。

従来の囚人のジレンマ向けの戦略は P2P ではうまく働かない。スケーラブルでロバストな、返礼型の決定関数にもとづくインセンティブ手法を提案する。これによって上記の問題を解決し、トレードオフを提供する:

- 差別的サーバ選択: 協調を促すためにはノード間の関係が持続することが重要。ところが、ノード数が多く、入れ替わりの激しい P2P システムでは同一ノードとのインタラクションが連続することはあまりない。解決策: 個々の peer が他の peer の自分に対するアクションの履歴をとり、差別的にサーバを選択すると、返礼型の決定関数の効き目が増す。具体的には、多くのノードと一定の入れ替わり頻度においても効く。

- 履歴の共有: 高い入れ替わり頻度と興味の非対称性に対応するためには履歴を共有することが必要。Figure 1: A, C のインタラクションを B が知らない場合、履歴を共有していると良いことがある。
- Maxflow に基づく主観的評価 (subjective reputation): 履歴を共有すると結託の可能性がある。Maxflow に基づくアルゴリズムを使うと参加ノードの 1/3 が結託しても協調を促すことができる。完全な Maxflow の計算量は $O(V^3)$ なので、ノードが増えると問題。ここでは平均して固定時間で動作する亜種をつくった。
- 適応的な新参者の取り扱い: コストのかからない ID のせいで、協調しない peer が協調しなかったことの報いから逃れることができ、結果的にシステム全体の協調を壊してしまう。返礼型の peer において、新参者の取り扱いを過去の新参者のふるまいにしたがって決めれば、peer が whitewash するインセンティブが減少し、whitewashing をほとんどなくせる。適応的な新参者の扱いは、ID の集中型割り当てや、新参者の入場料や、レート制御を必要としない。
- 短期的な履歴: 良い評価をもつ peer が裏切り、良い評価をつかって他の peer を悪用する危険性がある。長期的な履歴はこの問題を悪化させる。短期的な履歴をつかえば、裏切り者が協調をないがしろにすることを防げることを示す。

2.2 モデル

- Social dilemma: 協調する人間より、協調する他人を利用するが当人は協調しないという人間のほうが利益を得る。全員が協調すると、システム全体の利便性が最大化されるが、個人としての利益追求行動のために最適状態が達成されないというジレンマ。
- 非対称な取引: 相手の求めるサービスを提供することはできないが、相手のサービスを求める、など。
- 追跡不可能な脱出: 誰が脱出したのかを知ることはできない。
- 動的な参加者: Peer は自らの意思でいつでも、振る舞いを変えたり、入退場することができる。

2.3 Generalized Prisoner's Dilemma

Prisoner's Dilemma: 上記 Social dilemma の要件を満たす非協調型の繰り返しゲーム。個々のゲームで、二人のプレイヤーは脱出するか協調するかを選択できる。行動の結果として payoff を受け取る。プレイヤーは戦略にもとづき行動する。

Generalized Prisoner's Dilemma (GPD): Social dilemma の要件を満たし、かつ非対称な payoff matrix を使うもの。

Figure 2: GPD の payoff matrix.

Section 2.3 中段: social dilemma を成立させるための制約条件。

Figure 3: P2P file sharing, overlay routing のような P2P アプリケーションにおける payoff matrix で、上記の制約条件を満たすもの。以後、この payoff matrix を使う。

2.4 Population Dynamics

ラウンド毎にプレイヤーは 1) mutate, 2) learn, 3) turnover, 4) stay the same できる。mutate: ランダムに選んだ戦略に切り替え。learn: 高いスコアを上げるとされる戦略に切り替え。turnover: 同じ戦略をもつ新プレイヤーに入れ替え。

3. 返礼型の決定関数

決定関数: プレイヤーの行動履歴に基づいて、協調する・しないを決める。

寛容さ: 享受した利益と与えた利益の関係で量ることができる。

式 (1): エンティティ i の寛容さ。 p_i : i が与えたサービス、 c_i : i が受け取ったサービス。

式 (2): 正規化された寛容さ。式は、 j が考える i の正規化された寛容さを表している。

3.1 Simulation Framework

前述したモデルをシミュレータで実装。Figure 3 の非対称な payoff matrix を使う。様々な非協調的振る舞いによる問題を異なるシナリオで取り扱う。

Table 1: シミュレーションで用いるパラメータの既定値。個々のシナリオでは特定の状況や攻撃を織り込むため、特定のパラメータを変える。そして返礼型戦略の特性を変えて、個々の状況や攻撃に対処する。

3.2 Baseline Results

Table 1 の基本的なシナリオにおける結果。Figure 4(a): 60 player の場合。最終的に返礼型戦略をとるノードが大勢を占める。Figure 4(b): 120 player の場合。脱出戦略をとるノードが大勢を占める。

Figure 5(a): ラウンド毎のスコア全体平均。6 が最大 (全員が協調)、0 が最小 (全員が非協調的)。

4. 返礼に基づくインセンティブ手法

4.1.1 個人的履歴を用いたサーバ選択

Past donor, past recipients の固定長リストを保持しておき、その中からサーバを選ぶ手法。

Figure 6: ノード数が多い場合でも有効であることを示している。(Private Selective)

Figure 7: ノード数を固定し、入れ替わり率を変えている。入れ替わり率が高くなると、スケールしない。固定長リストでは効かなくなるため。

4.1.2 共有履歴

興味の非対称性と高い入れ替わり率に対処するため、共有履歴が必要。共有履歴とは: すべての peer がシステム内のすべてのインタラクションについて記録を持つ。

Figure 7: 共有履歴が入れ替わり率が高いときに有効であることを示している。

共有履歴の欠点：通信オーバーヘッド (DHT あるいは中央集中型によるもの) と、結託への脆弱性。

4.2.1 結託

客観的評価にたよる戦略は結託によわい。このため主観的に評価を計算する。ここでは Maxflow にもとづく主観的アルゴリズムを提案する。Maxflow はグラフ理論の問題。Figure 8 を例として解説。

Maxflow アルゴリズムを応用。vertex: ノード、edge: 享受したサービス。

Figure 9: すべての結託者 C が互いに高い評価を下したとする。ノード A がノード B について主観的評価を計算するとき、高い (嘘の) 評価に影響されないことに注目。ここでは Maxflow はゼロ。

式 (3): i が考える j の主観的評価。

Algorithm 1: Maxflow アルゴリズムの計算量は最悪 $O(V^3)$ であるので、手抜きアルゴリズムを提案。フローが存在する場合でも返さない可能性がある。その代わり平均して固定時間で動作する。

Figure 10: 結託者が存在するときの、主観的評価の有効性。X 軸についてまったく説明がないが、おそらく結託者の数だともわれる。

4.3 コストのかからない ID

中段: 新参者の取り扱いを過去の新参者のふるまいにしたがって決める式。

Figure 11: 新参者の取り扱い方針とスコアの関係。共有履歴の場合。Stranger Cooperate: 新参者つつねに協調する、Stranger Defect: 新参者は相手にしない、Stranger Adaptive: 新参者の扱いは過去の新参者の振る舞いで決まる。

Figure 12: 個人的履歴の場合。これらを総合して、Stranger Adaptive が最も有効であると結論づけている。

4.4 Traitors

Figure 13 上のグラフ 2 つ: 長期的履歴 (履歴がクリアされない) における裏切りの効果。裏切りが有効であることがわかる。

下のグラフ 2 つ: 短期的履歴を用いた場合。裏切りは迅速に発見でき、全体としての協調レベルは高いまま維持される。