OptOrder: optimum serial dictatorship in stochastic matching

Someone

Abstract

This paper studies a simplified version of the stochastic matching proposed in [Chen *et al.*, 2009]. Previous study want to find the optimal policy for a matching system to maximize the cardinality of a matching. In that setting, an agent may reject the assigned candidate (with a certain probability) and stay in the market or leave. Finding the optimum policy to match meets great difficulties mainly from two aspects. First, there is no evidence that the optimal policy can be expressed in polynomial space. Second, collecting the acceptance ratio of each pair usually incurs considerable costs.

To express a policy efficiently and solve the optimum solution efficiently, we choose serial dictatorship and consider the case where the acceptance ratios are identical for each pair. This paper aims to give an algorithm with a graph and the acceptance ratio as input and outputs a optimum order for serial dictatorship. We first describe a straightforward integer linear programming with $O(|E|^4)$ entries. Then, we reduce the number of entries to $O(|E|^2)$ by an important observation. Finally, in the experimental part, we compare methods to solve even larger graphs.

1 Introduction

Motivated by applications like kidney exchange and online dating, [Chen *et al.*, 2009] defines stochastic matching. The goal to maximize the cardinality of a stochastic matching. The problem can be modeled in a graph. Any edge has a probability to be fake. (In our work, we assume that these probabilities are identical.) When a fake edge is probed to match, it rejects the prob and the edge is removed from the graph. Otherwise, it accepts the prob and both ends of the edge are removed from the graph. They show a greedy algorithm with 0.25 approximation compared to the optimal algorithm. The approximation ratio was improved by [Adamczyk, 2011; Costello *et al.*, 2012]. When compared to the offline optimal, this problem is closely related to the randomized greedy matching problem. [Poloczek and Szegedy, 2012; Aronson *et al.*, 1995] show a greedy algorithm that can match more than one half agents and [Goel and Tripathi, 2012] improve the lower bound significantly.

The current setting faces two main drawbacks to prevent it from practical use.

First, to know the exact probability of a prob being accepted is difficult. Typically, in order to know the exact probabilities, the market needs to collect many data and conduct careful analyses on historical data. It is worthy in important markets like kidney exchange market [Dickerson *et al.*, 2013; Dickerson and Sandholm, 2015]. However, in many less important market, collecting acceptance ratios is quite costly and unworthy. Similar to online dating problem, let us consider a friend recommendation problem. Someone (called Alice) has many friends, x of them are single boys and y are single girls. Alice wants to introduce boys to girls for marriage. She wants to maximize the number of matched couples. Every boy or girl has some requirements that must be satisfied. However, even though a boy and a girl satisfy each other's requirements, they may still not be willing to match. In this case, it is hard and costly for Alice to survey the acceptance ratio of each pair. Similar problem also appear in other applications like roommate problem [Roth, 1982], but never serious considered as far as we know.

Second, as mentioned above, most study focus on giving worst case guarantees, compared both to the optimal solution and the offline optimum. The main difficulty preventing people to desire the optimum solution is that the optimal solution maybe not able to be expressed in polynomial space. To solve the stochastic matching problem in practical, there is no evidence that the existing algorithms can work well enough.

In this paper, a simplified setting of stochastic matching is considered, which can still be utilized to solve many real life applications. We assume that all the edges have the same acceptance ratio. This assumption avoid the difficulty in surveying the acceptance probabilities and provides an essential condition for our algorithm. As for the expressiveness of the solution, we improve it by serial dictatorship. Although, serial dictatorship may reduce the expected number of matched agents, our experimental results demonstrate that the influence is not big. With the solution, anyone can conduct the matching without any further help of a computer. One only needs $O(1)$ computation in average for each prob. We are going to release our work as an open source tool. With the help of this tool, anyone is able make precise decisions for stochastic matching problems around themselves.

The following part of this paper is organized as follows. Section 2 defines the serial dictatorship and show some fundamental good properties of it. Section 3 shows the construction of the integer linear programming step by step and prove its correctness. Section 4 explores methods to expand our algorithm to larger graphs. Section 5 shows our experimental results. Section 6 concludes this paper.

2 The settings

In this section, we will describe our model and the serial dictatorship. After that, some desired properties of serial dictatorship are demonstrated.

2.1 The model

Our problem is modeled in an undirected graph $G = (V, E)$, $V = \{v_1, v_2, \ldots, v_n\}$ denotes the set of agents. $e_{ij} \in E$ denotes an edge between v_i and v_j. For any edge $e_{ij} \in E$, it has two states, *exist* or *fake*. The probability of "exist" is a constant p. The goal of the matching is to maximize cardinality. Each time, the system prob a pair of agents v_i and v_j with an edge e_{ij} connecting them.

- If e_{ij} exists, then v_i and v_j agree to match with each other, both of the two agents and the edges attached to them are removed from the graph. The number of matched agents increases by 2.
- If e_{ij} is fake, v_i and v_j refuses to match. The system removes e_{ij} from E.

2.2 ESD algorithm

To maximize the number of matched agents, we apply *edge serial dictatorship algorithm* (ESD). The outline is described In Algorithm 1.

Algorithm 1 ESD algorithm

Require: A sorted array of E, denoted by $S = s_1, s_2, \ldots, s_m$ and the set of vertices V.
Ensure: A set of disjoint edges A, such that each two agents connected by an edge in A agree to match.

$A = \emptyset$
for do $i = 1, 2, \ldots, m$
  Let v_x and v_y be the two agents connected by s_i
  if then $v_x \in V$ and $v_y \in V$
    Prob v_x and v_y whether they agree to match with each other.
    if B **then**oth v_x and v_y accept to match
      Put s_i into A
      Remove v_x and v_y from V
    end if
  end if
end for
Outputs A as the set of matched edges.

Algorithm 1 is a typical serial dictator algorithm. The system just picks edges one by one in some order (denoted by S) and try to match the current edge if both ends of the edge are still in the graph. Our goal is to maximize the expected size of the output set A among all S.

2.3 Desired properties

We restrict our focus on the serial dictatorship for many reasons.

Usability. The solution of ESP has a size of $O(|E|)$, which equals to the size of input. The executor of the matching only needs to use O(1) computation for each prob. These two properties make the solution perfect to be used as handouts. In other words, the order can be generated by a computer and any people can execute the matching without a computer.

Deterministic. A great advantage of ESD is that it outputs a deterministic solution. One can count the expected number of matched agents easily. Many previous algorithms lack this property. In our setting, given the serial S, the expected number of matched agents $c(S)$ can be counted by the following equation.

$$c(S) = 2 * \sum_{i=1}^{m} p(1-p)^{\sum_{j=1}^{i-1} \delta(s_j, s_i)} \quad (1)$$

In equation 1, $\delta(s_i, s_j)$ is an indicator of whether s_j appears before s_i and they have a common vertex. In other words, $\delta(s_j, s_i) = 1$ only when s_j is probed before s_i and they have a common vertex, otherwise $\delta(s_j, s_i) = 0$. In the integer linear programming, we will use variables δ_{ij}, which holds the same meaning as $\delta(s_i, s_j)$ here.

3 Optimizing the serial

Algorithm 1 has given a framework of serial dictatorship. The problem is how to find the serial S that maximizing the expected cardinality of the matching. In this section, we will introduce the integer linear program (ILP) we used to find the optimum S step by step.

3.1 Overview of the algorithm

The objective function of our ILP is a representation of Equation 1. As for the constraints, if we faithfully write the constrains according to definition of serial dictatorship and the objective function, we need $O(|E|^4)$ variables in the ILP, which is insufferable. We are not going to discuss in details for this idea.

To reduce the number of variables, we introduce a novel method. We first relax the solution space to obtain a relaxed-form ILP. Then, we show that the solution of the relaxed-form ILP can be mapped to the optimal serial S.

To relax the solution space,

References

[Adamczyk, 2011] Marek Adamczyk. Improved analysis of the greedy algorithm for stochastic matching. *Information Processing Letters*, 111(15):731–737, 2011.

[Aronson *et al.*, 1995] Jonathan Aronson, Martin Dyer, Alan Frieze, and Stephen Suen. Randomized greedy matching. ii. *Random Structures & Algorithms*, 6(1):55–73, 1995.

[Chen *et al.*, 2009] Ning Chen, Nicole Immorlica, Anna R Karlin, Mohammad Mahdian, and Atri Rudra. Approximating matches made in heaven. *Automata, Languages and Programming*, pages 266–278, 2009.

[Costello *et al.*, 2012] Kevin P Costello, Prasad Tetali, and Pushkar Tripathi. Stochastic matching with commitment. In *Automata, Languages, and Programming*, pages 822–833. Springer, 2012.

[Dickerson and Sandholm, 2015] John P Dickerson and Tuomas Sandholm. Futurematch: Combining human value judgments and machine learning to match in dynamic environments. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[Dickerson *et al.*, 2013] John P Dickerson, Ariel D Procaccia, and Tuomas Sandholm. Failure-aware kidney exchange. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 323–340. ACM, 2013.

[Goel and Tripathi, 2012] Gagan Goel and Pushkar Tripathi. Matching with our eyes closed. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 718–727. IEEE, 2012.

[Poloczek and Szegedy, 2012] Matthias Poloczek and Mario Szegedy. Randomized greedy algorithms for the maximum matching problem with new analysis. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 708–717. IEEE, 2012.

[Roth, 1982] Alvin E Roth. Incentive compatibility in a market with indivisible goods. *Economics letters*, 9(2):127–132, 1982.