ILP formulation and LP relaxation

Integer Linear Programming based approach

February 23, 2022

Table of Contents

Set cover

Input : $(U, S_1, S_2, \cdots, S_m)$

1. introduce a variable x_i for every set S_i
2. $x_i = 1$ when S_i is selected, and $x_i = 0$ otherwise.

LP formulation

derive the linear programming relaxation

Minimize $\sum_{i=1}^{m} x_i$
Subject to

- $\sum_{i: v \in S_i} x_i \geq 1 \; \forall v \in U$
- $x_i \leq 1 \; \forall i \in \{1, 2, \cdots, m\}$
- $x_i \geq 0 \; \forall i \in \{1, 2, \cdots, m\}$

Weighted LP optimization

changing the cost method

Minimize $\sum_{i=1}^{m} w_i.x_i$
Subject to

- $\sum_{i:v\in S_i} x_i \geq 1 \; \forall v \in U$
- $x_i \leq 1 \; \forall i \in \{1, 2, \cdots, m\}$
- $x_i \geq 0 \; \forall i \in \{1, 2, \cdots, m\}$

Weighted LP optimization

optimal fractional solution x^*

$$x_i^* \in [0,1]$$

we cannot round the x_i^* to the nearest integer, because if an element u belongs to 100 sets, it could be that $x_i^* = \frac{1}{100}$ for each of those sets, and we would be rounding all those numbers to zero, leaving the element u not covered.

Weighted LP optimization

Simply rounding doesn't work here

If we knew that every element u belongs to at most k sets, then we could round the numbers $\geq \frac{1}{k}$ to 1, and the numbers $\leq \frac{1}{k}$ to zero.

k - approximation

Unfortunately, k could be very large, even $\frac{n}{2}$

Normalized Probability

cosider x_i^* as a probability

we can think of the solution x^* as describing a probability distribution over ways of choosing some of the subsets $S_1, \cdots, S_m$, in which we choose S_1 with probability x_1^* , S_2 with probability x_2^* , and so on.

Randomization

Algorithm *RandomPick*

- Input: values $(x_1, \ldots, x_n)$ feasible for (3)
- $I := \emptyset$
- for $i = 1$ to n
  - with probability x_i, assign $I := I \cup \{i\}$, otherwise do nothing
- return I

RandomPick

Expected behaviour

Using this probabilistic process, the expected cost of the sets that we pick is $i \sum_{i=1}^{m} w_i.x_i^*$,

Expected behaviour

Unfortunately, the solution that we construct could have a high probability of missing some of the elements

Randomization

Algorithm *RandomizedRound*

1. Input: $x_1, \ldots, x_n$ feasible for (3)
2. $I := \emptyset$
3. while there are elements u such that $u \notin \bigcup_{i \in I} S_i$
   - for $i := 1$ to n
     - with probability x_i, assign $I := I \cup \{i\}$, otherwise do nothing
4. return I

RandomidedRound

Fact

There is a probability at most e^{-100} that the while loop is executed for more than $ln|U| + 100$ times. In general, there is a probability at most e^{-k} that the while loop is executed form more than $ln|U| + k$ times.

RandomidedRound

Fact

Fix any positive integer parameter t and any feasible solution $(x_1, \ldots, x_m)$ for LP formulation. Then the expected size of I in Algorithm RandomizedRound on input $(x_1, \ldots, x_m)$ after t iterations. (or at the end of the algorithm, if it ends in fewer than t iterations) is at most

$$t.\sum_{i=1}^{m} w_i.x_i$$

RandomidedRound

Fact

Given an optimal solution $(x_1^*, \cdots, x_m^*)$ to (3), algorithm RandomizedRound outputs, with probability $\geq .45$, a feasible solution to the set cover problem that contains at most $(2ln|U| + 6) \cdot$ opt sets.