CMPUT 675: Approximation Algorithms	**Fall 2014**

Lecture 10 (Sep 24): Knapsack and MAX SAT

Lecturer: Zachary Friggstad *Scribe: Nikos Fasarakis-Hilliard*

10.1 LP relaxation for the Knapsack Problem

We formulate the Knapsack problem as an integer program and consider the quality of its LP relaxation. The Knapsack problem was previously introduced in Lecture 6.

Definition 1 *In the* Knapsack *problem, we are given a set of items, $I = \{1, \ldots, n\}$, each with a weight $w_i \geq 0$, and a value $v_i \geq 0$. We are also given a weight capacity $C \geq 0$. The goal is to find a maximum-value subset $X \subseteq I$ with $\sum_{i \in X} w_i \leq C$.*

We can formulate the problem as the following integer program, in which variable $x_i \in \{0, 1\}$ for each $i \in I$, indicates whether item i is chosen to be in the solution or not:

$$\begin{aligned} \text{maximize: } & \sum_{i \in I} v_i x_i & \textbf{(KP-IP)} \\ \text{subject to: } & \sum_{i \in I} w_i x_i \leq C, & (10.1) \\ & x_i \in \{0,1\}, \text{ for each } i \in I. & (10.2) \end{aligned}$$

Constraint (10.1) ensures the total weight does not exceed the capacity and constraints (10.2) restrict variables x_i to be binary. We obtain a linear programming relaxation for problem **KP-IP** by replacing the constraints $x_i \in \{0, 1\}$ with linear constraints $0 \leq x_i \leq 1$, for each $i \in I$:

$$\begin{aligned} \text{maximize: } & \sum_{i \in I} v_i x_i & \textbf{(KP-LP)} \\ \text{subject to: } & \sum_{i \in I} w_i x_i \leq C, & (10.3) \\ & x_i \in [0,1], \text{ for each } i \in I. & (10.4) \end{aligned}$$

This is a relaxation of problem **KP-IP** in the sense that the set of feasible solutions of problem **KP-LP** is a superset of the feasible solutions to problem **KP-IP**.

The integrality gap of **KP-LP** can be very bad. For example, suppose the Knapsack instance consisted of a single item $I = \{i\}$ with $v_i = w_i = 2$ for some large value D. Furthermore, suppose the capacity C was only 1. The the optimum integer solution is 0 since nothing fits, yet the optimum LP solution has $x_i = 1/2$ with value $v_i/2 = 1$.

The problem is that i did not fit on its own. We will discard such items and now assume $w_i \leq C$ for each item i. Clearly the optimum integer solution is the same since none of the discarded items fit on their own. We will show the integrality gap is at least $1/2$ under this assumption. There are quite a few ways to do this; we will use the properties of extreme points.

Extreme point characterization

In Lecture 9 it was shown that there is an optimum solution that is also an extreme point for any linear program that included all nonegativity constraints $\mathbf{x} \geq 0$. So, let $\mathbf{x}^*$ be an optimum solution that is also an extreme point and let $\mathbf{A}[\mathbf{x}^*]$ be the matrix of tight constraints. Also recall that we saw that the rank of $\mathbf{A}[\mathbf{x}^*]$ is equal to the number of variables. In this case, $\mathbf{A}[\mathbf{x}^*] = n$.

Let $A = \{i \in I : x_i^* = 1\}$ denote the set of items that are "fully selected" by $\mathbf{x}^*$ $B = \{i \in I : 0 < x_i^* < 1\}$ denote the set of items that are "fractionally selected" by $\mathbf{x}^*$.

In fact, B is a very small set.

Claim 1 $|B| \leq 1$

Proof. Since $\text{rank}(\mathbf{A}[\mathbf{x}^*]) = n$, then there are at least n tight constraints. Therefore, there are at least $n-1$ tight constraints of the form $0 \leq x_i^*$ or $x_i^* \leq 1$. Since we cannot have both $x_i^* = 0$ and $x_i^* = 1$ for any $i \in I$, then for at least $n-1$ items i, $x_i^* = 0$ or $x_i^* = 1$. ■

Theorem 1 *Both A and B are feasible and at least one of them has value at least $OPT_{LP}/2$.*

Proof. To see A is feasible:

$$\sum_{i \in A} w_i = \sum_{i \in A} w_i x_i^* \leq \sum_{i \in I} w_i x_i^* \leq C. \tag{10.5}$$

In addition, B is feasible by Claim 1: $|B| \leq 1$ and $w_i \leq C$ for each $i \in I$.

Also,

$$\begin{aligned} v(A) + v(B) &\geq \sum_{i \in A} v_i x_i^* + \sum_{i \in B} v_i x_i^* \\ &= \sum_{i \in I} v_i x_i^* = \text{OPT}_{\text{LP}}. \end{aligned} \tag{10.6}$$

Therefore, either $v(A)$ or $v(B)$ is at least $\text{OPT}_{\text{LP}}/2$.

■

In other words, if we have $w_i \leq C$ for each item $i \in I$ then the integrality gap of **KP-LP** is at least $\frac{1}{2}$.

10.2 The MAX SAT problem

In the maximum satisfiability problem (MAX SAT), we are given clauses $C_1, C_2, \ldots, C_m$, each a disjunction of literals over variables $x_1, x_2, \ldots, x_n$ (e.g. $x_1 \vee \bar{x}_2 \vee x_3$). Each of the variables x_i may be set to either true of false. The objective of the problem is to find a truth assignment that satisfies the maximum possible number of clauses.

To formulate MAX SAT as a linear program, we will use two sets of variables:

- $z_i \in \{0,1\} \equiv$ "Is x_i set to true?"
- $y_C \in \{0,1\} \equiv$ "Is clause C satisfied?"

For a clause C, we say $x_i \in C$ to mean variable i appears positively as a literal in C and $\bar{x}_i \in C$ to mean variable i appears negatively in C. For example, if $C = x_1 \vee \bar{x}_2 \vee x_3$ then we say $x_1, x_3 \in C$ and $\bar{x}_2 \in C$.

The following is an LP relaxation for MAX SAT:

$$\text{maximize: } \sum_C y_C \qquad \textbf{(MS-LP)}$$

$$\text{subject to: } y_C \leq \sum_{i:x_i \in C} z_i + \sum_{i:\bar{x}_i \in C} (1 - z_i), \text{ for each clause } C, \qquad (10.7)$$

$$z_i, y_C \in [0,1], \quad \text{for each clause } C \text{ and each variable } x_i \qquad (10.8)$$

This is a relaxation in the sense that in any $\{0,1\}$-integer solution, we can only have $y_C = 1$ (i.e. clause C is satisfied) if at least one of the literals appearing in C is true under the corresponding $\{0,1\}$ assignment to the z_i variables. Conversely, any truth assignment to the x_i variables corresponds to a $\{0,1\}$-assignment to the variables of **MS-LP** where we may set $y_C = 1$ if and only if C under this truth assignment. In particular, $OPT_{\text{LP}} \geq OPT$.

Algorithm

Let $(\mathbf{y}^*, \mathbf{z}^*)$ be an optimum LP solution. Then:

$$\text{Set } x_i \text{ to } \begin{cases} \text{TRUE} & \text{with probability } z_i^*, \\ \text{FALSE} & \text{with probability } 1 - z_i^*. \end{cases}$$

We prove the following claim:

Claim 2 *For any clause C with, say, k literals,* $\Pr[C \textit{ is satisfied }] \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \cdot y_C^*$.

Proof. Instead of computing $\Pr[C$ is satisfied $]$, we will compute $\Pr[C$ is not satisfied $]$.

$$\begin{aligned} \Pr[C \text{ is not satisfied }] &\overset{(a)}{=} \prod_{x_i \in C} (1 - z_i^*) \prod_{\bar{x}_i \in C} z_i^* \\ &\overset{(b)}{\leq} \left(\frac{\sum_{x_i \in C} (1 - z_i^*) + \sum_{\bar{x}_i \in C} z_i^*}{k} \right)^k \\ &= \left(\frac{k - \sum_{x_i \in C} z_i^* + \sum_{\bar{x}_i \in C} (1 - z_i^*)}{k} \right)^k \\ &\overset{(c)}{\leq} \left(1 - \frac{y_C^*}{k}\right)^k, \end{aligned} \qquad (10.9)$$

where

(a) follows since x_i's are sampled independently.

(b) follows from the Arithmetic-Geometric Mean Inequality: for $a_1, \ldots, a_k \geq 0$ we have $\left(\prod_i a_i\right)^{1/k} \leq \frac{1}{k} \cdot \sum_i a_i$.

(c) follows from constraint (10.7) in **MS-LP**.

It follows that:

$$\Pr[C \text{ is satisfied }] = 1 - \Pr[C \text{ is not satisfied }] \geq 1 - \left(1 - \frac{y_C^*}{k}\right)^k. \tag{10.10}$$

Now consider the function $g(w) \triangleq 1 - \left(1 - \frac{w}{k}\right)^k$ on $[0,1]$. Then, it can be shown that $g(w)$ is *concave* on $[0,1]$ (recall $k \geq 1$). For any function that is concave on $[0,1]$, the graph of g lies above the line segment joining two points $\big(0, g(0)\big)$ and $\big(1, g(1)\big)$, i.e.:

$$g\big(t\big) \geq (1-t) \cdot g(0) + t \cdot g(1), \text{ for all } t \in [0,1]. \tag{10.11}$$

For $t = y_C^* \in [0,1]$, we see

$$\Pr[C \text{ is satisfied }] \geq g(y_C^*) \geq \left(1 - \big(1 - \frac{1}{k}\big)^k\right) \cdot y_C^*. \tag{10.12}$$

■

Theorem 2 *The expected number of satisfied clauses is at least* $\left(1 - \frac{1}{e}\right) OPT_{LP} \geq \left(1 - \frac{1}{e}\right) OPT.$

Proof. First note that $(1 - 1/k)^k \leq 1/e$ for any $k \geq 1$ where $e \approx 2.718$ is the base of the natural logarithm.

Then,

$$\begin{aligned} \mathrm{E}[\#\text{ satisfied clauses}] &= \sum_C \Pr[C \text{ is satisfied}] \\ &\geq \left(1 - \frac{1}{e}\right) \cdot \sum_C y_C^* \qquad \big(\text{Follows from Claim 2 and } (1-1/k)^k \leq 1/e\ \big) \\ &= \left(1 - \frac{1}{e}\right) \cdot \mathrm{OPT}_{\mathrm{LP}} \\ &\geq \left(1 - \frac{1}{e}\right) \cdot \mathrm{OPT}. \end{aligned} \tag{10.13}$$

■

What we have just shown is that our randomized rounding algorithm finds, in expectation, at least $(1-1/e) \cdot OPT_{\mathrm{LP}} > 0.632 \cdot OPT$ clauses. This may be viewed as a slightly weak statement because this is only in expectation (i.e. not guaranteed to happen).

Section 5.2 of the Williamson and Shmoys text presents a technique that can be adapted to "derandomize" this algorithm. In particular, this yields a deterministic, polynomial time algorithm that rounds an optimal LP solution to satisfy at least $(1-1/e) \cdot OPT_{\mathrm{LP}}$ constraints.

Even without this derandomization, this still provides a bound on the integrality gap. That is, since the expected number of clauses is at least $(1-1/e) \cdot OPT_{\mathrm{LP}}$, then there must be *some* solution that satisfies at least this many clauses. At any rate, in this class we will be happy with randomized approximation algorithms that find a solution whose *expected* cost or value is within some bound of the optimum.

Finally, note that this rounding algorithm favours small clauses. That is, $1 - (1-1/k)^k$ is larger for small k. Next lecture, we will see an alternative algorithm for MAX SAT that favours large clauses. The worst-case ratio of this algorithm will only be $1/2$, but running these two algorithms and taking the better of the two solutions will be a $3/4$-approximation, which is better than both $1 - 1/e$ and $1/2$.