AM 221: Advanced Optimization	Spring 2016
Prof. Yaron Singer	*Lecture 22 — April 18th*

1 Overview

In this lecture, we will study the *pipage rounding* technique which is a deterministic rounding procedure that can be used to round the solution of a continuous relaxation of a combinatorial problem. Pipage rounding is an interesting framework in its own right, and creates powerful approximation algorithms. More importantly, pipage rounding provides a gateway to the theory of submodular optimization. In this lecture we will describe pipage rounding for the Max-Cover problem. In the next lecture we will generalize this framework for submodular optimization.

2 Relaxations of Max-Cover

Let us look again at the Max-Cover problem. Recall that in this problem we are given sets $T_1, \ldots, T_n$ over universe of m elements, and the goal is to select $S^* \in \text{argmax}_{S:|S|\leq k} \left|\bigcup_{i\in S} T_i\right|$.

Integer programming formulation of Max-Cover. As we have already seen a few times, Max-Cover can be expressed as the following IP:

$$\begin{aligned} \max \sum_{i=1}^{m} z_i & \\ \text{s.t.} \sum_{j=1}^{n} x_j &\leq k \\ z_i &\leq \sum_{j:i\in T_j} x_j \\ x_i, z_j &\in \{0,1\} \end{aligned}$$

An equivalent way to write the above program is by expressing the variables associated with the elements covered z_i in terms of the variables associated with the sets that cover them x_i:

$$\begin{aligned} \max \sum_{i=1}^{m} \min\left\{1, \sum_{j:i\in T_j} x_j\right\} & \\ \text{s.t.} \sum_{j=1}^{n} x_j &\leq k \\ x_j &\in \{0,1\} \end{aligned}$$

LP relaxation. The formulation on the right-hand side can be relaxed as $\max_{\mathbf{x}\in P} L(\mathbf{x})$ with:

$$L(\mathbf{x}) = \sum_{i=1}^{m} \min\left\{1, \sum_{j:i\in T_j} x_j\right\}$$

and $P = \{\mathbf{x} \in [0,1]^n : \sum_{j=1}^{n} x_j = k\}$. Note that the objective function is concave. Specifically, it is piecewise linear, so the relaxation can be reformulated as an LP and solved efficiently.

Randomized relaxation. As we saw in the previous lecture, when discussing the dual Min-Cover problem, a fractional solution $\mathbf{x} \in [0,1]^n$ can also be interpreted as a random solution where each set $j \in [n]$ is included independently with probability x_j. Let us compute the expected value of this random solution. For each element $i \in [m]$, the probability that it is covered by at least one set is:

$$1 - \prod_{j:i\in T_j}^{n} (1 - x_j)$$

so the expected value associated with fractional solution $\mathbf{x}$ is:

$$F(\mathbf{x}) = \sum_{i=1}^{m} \left(1 - \prod_{j:i\in T_j}^{n} (1 - x_j)\right)$$

This leads to another relaxation of the Max-Cover problem: $\max_{\mathbf{x}\in P} F(\mathbf{x})$.

Relationship between the relaxations. Both relaxations (the one with F and the one with L) are over the same polytope P. They are relaxations in the sense that when $\mathbf{x}$ is taken to be an integral solution (*i.e.* $\mathbf{x} \in \{0,1\}^n$), their values coincide and are equal to the Max-Cover value function. Note however that the relaxation with L can be efficiently solved (L is piecewise-linear) but the one with F cannot. The following lemma shows that the two relaxed objective functions are within a constant multiplicative approximation ratio.

Lemma 1. $\forall \mathbf{x} \in [0,1]^n,\ F(\mathbf{x}) \geq (1 - 1/e)\, L(\mathbf{x})$.

Proof. Since both L and F sum over all elements in the universe a_i where $i \in [m]$, it suffices to show the bound on a given element. For a given element a_i, w.l.o.g. suppose a_i is covered by sets $T_1, \ldots, T_c$ for some $c \leq n$. Remember the Arithmetic-Mean-Geometric-Mean (AM-GM) inequality from the previous lecture:

> ***Arithmetic-Mean-Geometric-Mean(AM-GM) inequality:*** *If $x_1, \ldots, x_c$ are positive, then:*
>
> $$\prod_{j=1}^{c}(1 - x_j) \leq \left(1 - \frac{\sum_{j=1}^{c} x_j}{c}\right)^c$$

In our case the AM-GM inequality implies:

$$1 - \prod_{j=1}^{c}(1 - x_j) \geq 1 - \left(1 - \frac{\sum_{j=1}^{c} x_j}{c}\right)^c$$

We will now distinguish two cases:

If $\sum_{j=1}^{c} x_j \geq 1$:

$$1-\prod_{j=1}^{c}(1-x_j) \geq 1-\left(1-\frac{\sum_{j=1}^{c} x_j}{c}\right)^c \geq 1-\left(1-\frac{1}{c}\right)^c \geq 1-\frac{1}{e}=\left(1-\frac{1}{e}\right)\min\left\{1,\sum_{j=1}^{c} x_j\right\}$$

If $\sum_{j=1}^{c} x_j < 1$: Then define $g(t)=1-\left(1-\frac{t}{c}\right)^c$. It is easy to verify that g is concave, hence:

$$g(t) \geq (1-t)g(0)+tg(1)=t\left(1-(1-1/c)^c\right) \geq t(1-1/e) \tag{1}$$

We then get:

$$1-\prod_{j=1}^{c}(1-x_j) \geq 1-\left(1-\frac{\sum_{j=1}^{c} x_j}{c}\right)^c \geq (1-1/e)\sum_{j=1}^{n} x_j=(1-1/e)\min\left\{1,\sum_{j=1}^{c} x_j\right\}$$

where the second inequality used (1) with $t=\sum_{j=1}^{c} x_j$. □

3 The Pipage Rounding Framework

Let us now introduce the general theorem of the pipage rounding procedure, that we will then apply to the Max-Cover relaxations introduced above.

Theorem 2. *Let $L:[0,1]^n \to \mathbb{R}$ and $F:[0,1]^n \to \mathbb{R}$ be two functions such that:*

- $\max_{\mathbf{x}\in P} L(\mathbf{x})$ *and* $\max_{\mathbf{x}\in P} F(\mathbf{x})$ *are two relaxations of the same integer program* $\max_{\mathbf{x}\in Q} I(\mathbf{x})$*;*
- $F(\mathbf{x}) \geq \alpha L(\mathbf{x})$ *for any* $\mathbf{x}\in P$ *and some* $\alpha > 0$*;*
- $\max_{\mathbf{x}\in P} L(\mathbf{x})$ *can be solved in poly-time;*
- *for all **fractional** $\mathbf{x}\in P$, $\exists$**integral** $\bar{\mathbf{x}}\in Q$ computable in poly-time and $F(\bar{\mathbf{x}}) \geq F(\mathbf{x})$.*

Then there exists a poly-time α-approximation algorithm for $\max_{\mathbf{x}\in Q} I(\mathbf{x})$.

Proof. Let $\mathbf{x}_L^*$ be an optimal solution to $\max_{\mathbf{x}\in P} L(\mathbf{x})$ which is computable in poly-time. Then:

$$F(\bar{\mathbf{x}}_L^*) \geq F(\mathbf{x}_L^*) \geq \alpha L(\mathbf{x}_L^*) \geq \alpha\texttt{OPT}$$

so $\bar{\mathbf{x}}_L^* \in Q$ is a feasible solution to the integral problem and α-approximation to $\max_{\mathbf{x}\in Q} I(\mathbf{x})$. □

Application to Max-Cover. If we want to apply the theorem to the relaxations L and F of the Max-Cover problem, the first three assumptions are clearly satisfied using the functions L and F we defined above with $\alpha = 1-1/e$. What is left is to show that the last assumption is satisfied. That is, we need to give a magical rounding procedure that can take any vector $\mathbf{x}\in P=\{\mathbf{x}\in[0,1]^n : \sum_{i\in[n]} x_i = k\}$ and turn it into a vector $\bar{\mathbf{x}}\in Q=\{\mathbf{x}\in\{0,1^n\} : \sum_{i\in[n]} x_i = k\}$ without decreasing the value of F.

4 The Pipage Rounding Rounding

To implement the pipage rounding framework we will use the following simple meta-rounding procedure. The rounding follows n rounds. In each round we pick two coordinates that are fractional; assume we can easily find two points in P, each with one less fractional component, and call these points $\mathbf{x} + \alpha_{\mathbf{x}} d_{\mathbf{x}}$ and $\mathbf{x} - \beta_{\mathbf{x}} d_{\mathbf{x}}$ (these are obnoxious names for points, but they will serve us well soon). Assume also, that we know that one of these points has larger value for F than the value of the original point $\mathbf{x}$, i.e. $F(\mathbf{x}+\alpha_{\mathbf{x}} d_{\mathbf{x}}) \geq F(\mathbf{x})$ or $F(\mathbf{x}-\beta_{\mathbf{x}} d_{\mathbf{x}}) \geq F(\mathbf{x})$ (or both). Then, the algorithm will update $\mathbf{x}$ to be $\mathbf{x} + \alpha_{\mathbf{x}} d_{\mathbf{x}}$ if $F(\mathbf{x} + \alpha_{\mathbf{x}} d_{\mathbf{x}}) \geq F(\mathbf{x})$ and $\mathbf{x} - \beta_{\mathbf{x}} d_{\mathbf{x}}$ otherwise. Repeating this n times guarantees that we have a rounded solution. We formally describe this procedure below.

Algorithm 1 Pipage Rounding algorithm

Input. fractional $\mathbf{x} \in [0,1]^n$, polytope P and function F
while $\mathbf{x}$ is fractional **do**
    $\mathbf{x} + \alpha_{\mathbf{x}} d_{\mathbf{x}} \leftarrow$ a point in P that has one less fractional coordinate than $\mathbf{x}$
    $\mathbf{x} - \beta_{\mathbf{x}} d_{\mathbf{x}} \leftarrow$ anoter point in P that has one less fractional coordinate than $\mathbf{x}$
    if $F(\mathbf{x} + \alpha_{\mathbf{x}} d_{\mathbf{x}}) \geq F(\mathbf{x})$ **then**
        $\mathbf{x} \leftarrow \mathbf{x} + \alpha_{\mathbf{x}} d_{\mathbf{x}}$
    else
        $\mathbf{x} \leftarrow \mathbf{x} - \beta_{\mathbf{x}} d_{\mathbf{x}}$
    end if
end while
Return. $\mathbf{x}$

The Pipage rounding procedure generates a series of fractional points:

$$\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(l)}$$

where each point $\mathbf{x}^{(i+1)}$ has at least one fractional coordinate less than $\mathbf{x}^{(i)}$. The algorithm terminates after n steps since in each step the number of fractional coordinates decreases by one. Thus, the point $\mathbf{x}^{(l)}$ is integral, and $l \leq n$.

What's left is to show that in every iteration of the `while` loop when we have a point $\mathbf{x}$ we can find two points $\mathbf{x} + \alpha_{\mathbf{x}} d_{\mathbf{x}}$ and $\mathbf{x} - \beta_{\mathbf{x}} d_{\mathbf{x}}$ that have one fractional coordinate less than $\mathbf{x}$, and that for these points we will always have that $F(\mathbf{x}) \leq \max\{F(\mathbf{x} + \alpha_{\mathbf{x}} d_{\mathbf{x}}), F(\mathbf{x} - \beta_{\mathbf{x}} d_{\mathbf{x}})\}$. To do so, we will define the notion of *convexity in a direction*, which is analogous to directional derivates:

Definition. $F : [0,1]^n \to \mathbb{R}$ *is* ***convex in direction*** $\mathbf{d}_{\mathbf{x}}$ *if* $g(t) = F(\mathbf{x} + t \cdot \mathbf{d}_{\mathbf{x}})$ *is convex (in* t*).*

The following lemma now unravels some of the mystery behind the obnoxious point naming.

Lemma 3. *Assume that* $F : [0,1]^n \to \mathbb{R}$ *is convex in direction* $\mathbf{d}_{\mathbf{x}}$*, then for any* $\alpha_{\mathbf{x}}, \beta_{\mathbf{x}} \in [0,1]$*:*

$$F(\mathbf{x}) \leq \max\{F(\mathbf{x} + \alpha_{\mathbf{x}} \mathbf{d}_{\mathbf{x}}), F(\mathbf{x} - \beta_{\mathbf{x}} \mathbf{d}_{\mathbf{x}})\}.$$

Proof. We know that $g(t) = F(\mathbf{x} + t\mathbf{d_x})$ is convex in t so:

$$\begin{aligned}
F(\mathbf{x}) &= \\
&= g(0) \\
&= g\left(\frac{\beta_{\mathbf{x}}}{\alpha_{\mathbf{x}} + \beta_{\mathbf{x}}} \cdot \alpha_{\mathbf{x}} + \frac{\alpha_{\mathbf{x}}}{\alpha_{\mathbf{x}} + \beta_{\mathbf{x}}} \cdot -\beta_{\mathbf{x}}\right) \\
&\leq \frac{\beta_{\mathbf{x}}}{\alpha_{\mathbf{x}} + \beta_{\mathbf{x}}} \cdot g(\alpha_{\mathbf{x}}) + \frac{\alpha_{\mathbf{x}}}{\alpha_{\mathbf{x}} + \beta_{\mathbf{x}}} \cdot g(-\beta_{\mathbf{x}}) \\
&= \frac{\beta_{\mathbf{x}}}{\alpha_{\mathbf{x}} + \beta_{\mathbf{x}}} \cdot F(\mathbf{x} + \alpha_{\mathbf{x}}\mathbf{d_x}) + \frac{\alpha_{\mathbf{x}}}{\alpha_{\mathbf{x}} + \beta_{\mathbf{x}}} \cdot F(\mathbf{x} - \beta_{\mathbf{x}}\mathbf{d_x}) \\
&\leq \left(\frac{\beta_{\mathbf{x}}}{\alpha_{\mathbf{x}} + \beta_{\mathbf{x}}} + \frac{\alpha_{\mathbf{x}}}{\alpha_{\mathbf{x}} + \beta_{\mathbf{x}}}\right) \max\{F(\mathbf{x} - \beta_{\mathbf{x}}\mathbf{d_x}), F(\mathbf{x} + \alpha_{\mathbf{x}}\mathbf{d_x})\} \\
&= \max\{F(\mathbf{x} - \beta_{\mathbf{x}}\mathbf{d_x}), F(\mathbf{x} + \alpha_{\mathbf{x}}\mathbf{d_x})\}.
\end{aligned}$$

□

4.1 Applying pipage rounding in Max-Cover

In our case, as long as $\mathbf{x} \in P$ is fractional, then there must be at least two coordinates that are fractional since $\mathbf{x}$ must respect $\sum_{i\in[n]} x_i = k$. At a given iteration of the pipage rounding procedure, let p and q be fractional coordinates that are selected by the algorithm. For the point $\mathbf{x}$ Let us define the direction $d_{\mathbf{x}} = \mathbf{e}_p - \mathbf{e}_q$ (where e_j is the jth basis vector), $\alpha_b x = \min\{1 - x_p, x_q\}$ and $\beta_{\mathbf{x}} = \min\{1 - x_q, x_p\}$. From the above, what remains is to show that the function F is indeed convex in the directions $\mathbf{e}_p - \mathbf{e}_q$ for any $p, q \in [n]$.

Lemma 4. *$F(\mathbf{x}) = \sum_{i=1}^{m} \left(1 - \prod_{j=1}^{n}(1 - x_j)\right)$ is convex in every direction $\mathbf{e}_p - \mathbf{e}_q$, $\forall p, q \in [n]$.*

Proof. It is sufficient to prove that each term in the sum is convex in a given direction $\mathbf{e}_p - \mathbf{e}_q$. Each term in the sum represents the contribution on a single element in the universe. Consider the element a_i for some $i \in [m]$. Define:

$$F^{(i)}(\mathbf{x}) = 1 - \prod_{j:i\in T_j} (1 - x_j)$$

and

$$g^{(i)}(t) = F^{(i)}\left(\mathbf{x} + t(\mathbf{e}_p - \mathbf{e}_q)\right)$$

We need to prove that $g^{(i)}(t)$ is convex, for every $i \in [m]$. To show this, note that there are four cases to consider:

- Neither set T_p nor T_q cover a_i: in this case we have

  $$g^{(i)}(t) = 1 - \prod_{j:i\in T_j} (1 - x_j)$$

  since the function is independent of t, it is constant and trivially convex;

- The set T_q covers a_i but T_p does not: in this case

$$g^{(i)}(t) = 1 - \left(\prod_{j:i\in T_j j\neq q} (1-x_j)\right)(1-(x_q-t))$$

The function is affine in t and therefore convex;

- The set T_p covers a_i but T_q does not: in this case, as above the function is affine;

- Both sets T_p and T_q cover a_i: in this case

$$\begin{aligned} g^{(i)}(t) &= 1 - \left(\prod_{j:i\in T_j\ j\neq p,q} (1-x_j)\right)(1-(x_q-t))(1-(x_p+t)) \\ &= 1 + \left(\prod_{j:i\in T_j\ j\neq p,q} (1-x_j)\right)\big(t^2+(t+1)(x_p+x_q)-1\big) \end{aligned}$$

and the above function is clearly convex.

So, in all cases we have that the function is coves. Since we have a sum of convex functions with positive coefficients the function is convex. □

Putting everything together we can conclude with out main theorem:

Theorem 5. *For the Max-Cover problem the pipage rounding procedure produces a $1-1/e$ approximation algorithm.*

5 Discussion and Further Reading

The pipage rounding method we showed actually gives an approximation ration better than $1-1/e$: the approximation ratio is $1-(1-1/c)^c$ where c is the *frequency* of the instance (recall the definition of frequency from the previous lecture: it is the largest number of set that can cover an element in the instance to the problem; in the Max-Cover case the frequency is n). For more information about the pipage rounding method see [1].

References

[1] Alexander A. Ageev and Maxim Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *J. Comb. Optim.*, 8(3), 2004.