Lecture 14

Last Class: Submodular Functions

These functions can be minimized efficiently. We didn't finish the details about how exactly how to do this, but we started piecing things together by showing that there's a continuous, convex function the **Lovász Extension** for every discrete sub modular function. It turns out that minimizing the Lovász extension (which can be done efficiently using i.e. Ellipsoid) gives a way to minimize the original problem.

Today we're going to do something superficially similar. We're going to map discrete combinatorial optimization problems to continuous optimization problems. And specifically, to something you're all very familiar with: linear programs.

This Class: Linear Programming Relaxations for Combinatorial Problems

Let's jump into an example:

Vertex Cover:

Undirected graph **G=(V,E)**

Goal:
Find vertex set C ⊆ V.
min |C|
such that for every edge e = (u, v) in E
either u ∈ C or v ∈ C.

This problem is NP-Hard: One of Karp's 21 Original NP-complete problems.

Let's write this in a form that looks more like the optimization problems we've been looking at:

Find some vector x with length |V|. Each entry in x is going to correspond to a node and will take value 1 if the node is in C, 0 otherwise.

Vertex Cover IP

min $\Sigma_{\{v \text{ in } V\}}$ x_v — **min 1^T x**
s.t.
for all v, x_v ∈ {0,1}
for all u,v x_u + x_v >= 1 — **Ax >= b**

This is an integer linear program because it involves the {0,1} constraint on x. Many hard optimization problems can be written in this way.

What's a natural relaxation of the vertex cover problem?

Relaxed Vertex Cover LP

min $\Sigma_{\{v \text{ in } V\}}$ x_v — $1^T x$
s.t.
for all v, 0 <= x <= 1 — actually we can just drop the <= constraint
for all u,v $x_u + x_v >= 1$

Fact: Any solution to the ILP is feasible for the LP

But: LP doesn't necessarily have an integral solution.
opt(Vertex Cover ILP) <= opt(Relaxed Vertex Cover LP)

```
         x_1
          /\
         /  \
        /    \
       /______\
x_2            x_3
```

What are the optimum assignments for this problem?
LP opt: 3/2
Vertex cover opt: 2

In many cases a fractional solution can still be helpful.

Procedure:

1) Solve LP to obtain (possibly non-integral) solution x*
2) Round x* to integral solution x~
3) Argue that cost(x~) isn't much greater than cost(x*)

As we'll see soon, the exact rounding procedure that works is problem dependent. For vertex cover it happens to be very simple.

$\tilde{x}_v = 1$ if $x^*_v >= 1/2$
$\tilde{x}_v = 0$ if $x^*_v < 1/2$

Claim: The rounded x~ is a valid solution to the vertex cover problem.

Can anyone tell me why this is true? It’s a short argument.

If $x^*_u + x^*_v \geq 1$, it must be that one of $x^*_u, x^*_v \geq 1/2$ so then one of $\tilde{x}_u, \tilde{x}_v = 1$. So we have a vertex incident to edge (u,v)

Claim: $\tilde{x}$ gives a 2 approximation to vertex cover

For all v, $\tilde{x}_v \leq 2 x^*_v$
so
$\sum \tilde{x}_v \leq 2 \sum x^*_v \leq 2 \cdot$opt(Vertex Cover)

Could we have come up with a more clever rounding scheme that does better than a factor of 2?

I claim that from our triangle example, we certainly couldn’t have done better than a factor of 4/3.
Can someone tell me why?

- opt(LP) = 3/2
- 3/2 * loss < 2 if loss < 4/3
- but we know that we can’t actually do better than 2 with an integer solution —> contradiction

Think about the complete graph with > 3 vertices.

Optimal non-integer solutions = n/2 (put 1/2 on every vertex)
Optimal integer solution = n-1 (need to include all but one node in the cover)

We can find examples where opt(IP)/opt(LP) = 2(n-1)/n —> 2

For these examples, we certainly won’t be able to lose less than a factor of two when rounding or we’ll get a solution better than optimal.

This limitation is called the **integrality gap = sup opt(Integer Program)/opt(Linear Program)**

So even before you come up with a good rounding scheme, you can use the integrality gap to get a good sense of how tight your relaxation is. In theory, you could imagine a rounding scheme that bounds the distance of the rounded solution from the optimal integer solution, instead of from the optimal LP solution. However, virtually all known rounding schemes are analyzed like ours.

It turns out that a 2 factor approximation is basically the best you can do for vertex cover.

- some complicated techniques get $2 - O(1/\sqrt{\log|V|})$

- it can be show that beating 1.3606 is NP-hard [Dinur, Safra 2005]
- unique games conjecture implies 2-ε is hard for any fixed ε [Khot, Regev 2008]

In general LP relaxation techniques are very powerful.
However we lucked out with vertex cover in that it has a very simple deterministic rounding scheme.

Randomized Rounding
Set Cover:

Vertex Cover is actually a special case of this

Given: Some set of elements {1,2,… n} and a collection of subsets
S_1, S_2, …, S_m
i.e. {{1,2,3}, {2,4} , {3,4}, {4, 5}, {5}}

Want to select the smallest number of subsets that covers every elements {1,2,… n}

Again we're going to have an x_i for each S_i

Integer Program:

min $\Sigma_{i=1:m}$ x_i
for all j in {1,… n} $\Sigma_{\{i:\ j \text{ in } S\_i\}}$ x_i >= 1
for all i in {1, … m} x_i ∈ {0,1}

Relaxation:

min $\Sigma_{i=1:m}$ x_i
for all j in {1,… n} $\Sigma_{\{i:\ j \text{ in } S\_i\}}$ x_i >= 1
for all i in {1, … m} 0 <= x_i <= 1

Naively, deterministic Rounding does not work for this problem:
subsets =
{1,2,3 }
{ 2,3,4}
{1, 3,4}
{1,2, 4}

opt puts each x_i at 1/3 —> everything set to 0

You could lower your threshold, but taking this to the extreme:
{1,2, … n-1}
{2,3, … n}
…

You can get all of the weights down to 1/n. And you only need to select 2 subsets!
In fact, when ever set x~_i has a low weight, it must be that its elements are covered in other sets.

Attempt 1:

Set x~_i = 1 with probability x*_i

This does *something*:

$E[\Sigma_{S\_i:\ j\ in\ S\_i}\ \tilde{x}\_i] = \Sigma_{\{S\_i:\ j\ in\ S\_i\}} E[\tilde{x}\_i] >= 1$

So, the expected number of sets covering element j is 1.

But we want to get every set covered with good probability. So let's set things to 1 with higher probability.

Attempt 2:

Set x~_i= 1 with probability alpha x*_i

What's the probability element j is not covered?
let $Y\_j = \Sigma_{\{i:\ j\ in\ S\_i\}}$ x~_i
E[Y_j] >= alpha

Prob j not covered =
Pr[Y_j < 1] =
Pr[Y_j < 1/E[Y_i] * E[Y_i]]
<= exp((1-1/E[Y_i])^2 E[Y_i]/2)

delta = (1-1/E[Y_i])

Which is <= 1/n^2 as long as E[Y_i] > c log n

Choose alpha = O(log n)

Union bounding over {1,…, n} our rounded solution is a valid vertex cover with probability 1/n

What's our approximation factor?

E[∑ x~] = alpha * ∑x*]

cost(x~) <= O(log n) cost(x*) <= O(log n)*opt

Get O(log n) approximation in expectation.

By Markov's inequality, with probability 1/2 you'll be <= 2*O(log n)*opt, so we can just repeatedly retry and take the minimum solution to get an O(log n) approximation with high probability.

"Metric Uncapacitated Facility Location"

- NP-hard
- 1.488 [Li 2011]

* D of clients
* F of facilities
* Metric distance function: $d : (F \cup D) \times (F \cup D) \rightarrow R_+$ (define $d_{ij} \triangleq d(ij)$)
* Facility cost $f : F \rightarrow R_+$ (define $f_i \triangleq f(i)$)

Output: $S \subseteq F$ that minimizes $\sum_{i \in S} f_i + \sum_{j \in D} \min_{i \in S} d_{ij}$

Primal) P

$$\text{Min} \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} d_{ij} x_{ij}$$

$\times \beta_{ij}$ s.t. $x_{ij} \leq y_i \quad \forall i \in F, j \in D$

$\times \alpha_j$ $\sum_{i \in F} x_{ij} \geq 1 \quad \forall j \in D$

$x_{ij}, y_i \geq 0$

Dual) D

$$\text{Max} \sum_{j \in D} \alpha_j$$

s.t. $\sum_{j \in D} \beta_{ij} \leq f_i \quad \forall i \in F$

$\alpha_j - \beta_{ij} \leq d_{ij} \quad \forall i \in F, j \in D$

$\alpha_j, \beta_{ij} \geq 0$

Q: How to derive the dual program of (P)?

Interpretation of the dual vars:

For each client $j \in D$: α_j → The amount client j is willing to contribute

For each (facility, client): β_{ij} → The amount client j is willing to contribute to i

Primal-Dual Method: (Key insight: any feasible dual sol, is a lower bound for primal)

- Feasible integer primal solution for P
- Feasible solution for D
- Proof for cost(primal integer sol) $\leq \lambda \cdot$ cost (dual solution)

* Proposed by Dantzing, Ford and Fulkerson ('56) as a means of solving LP
* Later used in designing approx alg for NP-hard problems.
* General framework

General approach

- Begin with a dual feasible sol (all 0s typically) [Maintain feasibility]
- Raising dual variable(s) in a controlled manner
- Once some dual constraints get tight, set their corresponding primal val to some integral value.

Repeat untill a feasible primal soln is obtained

Approx. Alg. for metric UFL via Primal-Dual) [Jain-Vazirani 2001]

Phase 1)

- start with $\alpha, \beta = 0$.
  * Each client is unconnected
  * Each facility is unopened
- Each unconnected client raises its dual var, α_j
  - Untill $\alpha_j = d_{ij}$ for some facility i
    - client j has paid enough to reach i
    - (i,j) becomes tight
  - From now on, β_{ij} also increases.
    - To maintain $\alpha_j - \beta_{ij} \le d_{ij}$
  - If for some facility i, $\sum_{j \in D} \beta_{ij} = f_i$
    - i is "temporarily open"
    - All "unconnected" clients with "tight" edges to i $\longrightarrow$ connected.
      * i is "connecting witness" for these clients.
  - Dual variables of "connected clients" are not raised anymore.

- $N(j) = \{ i \in F : \alpha_j \geq d_{ij} \}$
- $N(i) = \{ j \in D : \alpha_j \geq d_{ij} \}$
- T: set of tight facilities

$\Rightarrow$ if for all i, $\alpha_j < \beta_{ij} + d_{ij}$ (increase α_j)

if for some i, $\alpha_j = \beta_{ij} + d_{ij}$ but $\sum \beta_{ij} < f_i$ (increase α_j and β_{ij})

Remark) The cost of dual, $\sum_{j \in D} \alpha_j$ consists of

- Connection cost
- facility opening cost

in primal

However, a client may have paid towards openning several facilities, although it eventually connects to only one.

⇒ If we open every temporarily opend facility, cost(primal) ≫ cost(dual)

$O(n)$ v.s. $\Omega(n^2)$.

Thus a cleanup phase is needed.

Phase 2)

F_t : the set of temp opened facilities.

* (i, i') are conflicting if $\exists j$ s.t. $\beta_{ij}, \beta_{i'j} > 0$.

- Pick a maximal indep. set I of F_t (in the order in which they were addet to F_t)
- For client j,

  if connecting witness of j is in I, assign client j to facility i.

  "Directly connected"

- ~~Otw~~

  otw, connect client j to the conflicting facility of i, i'.

  "Indirectly connected"

Claim 1) No client j contributes to two diff open facilities.

Pf. By construction of I.

Claim 2) If f_i is open, let S_i be the set of clients directly connected to i. Then

$$f_i + \sum_{j \in S_i} d_{ij} = \sum_{j \in S_i} \alpha_j$$

Pf. Any client j with $\beta_{ij} > 0$ is directly connected to i.

$$\Rightarrow f_i = \sum_{j \in D} \beta_{ij} = \sum_{j \in S_i} \beta_{ij}$$ (At the time the constraint gets tight)

Note that we may have $\beta_{ij} = 0$ for clients who joine the facility later.

For clients directly connected to i, the dual constraint

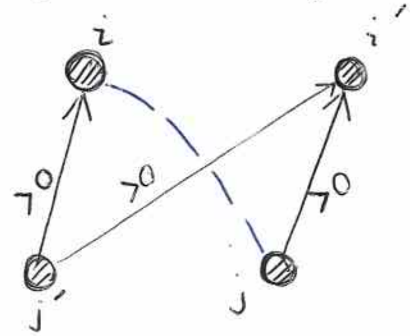
$$\alpha_j - \beta_{ij} \le d_{ij}$$

is tight.

$$\Rightarrow f_i = \sum_{j \in S_i} \beta_{ij} = \sum_{j \in S_i} (\alpha_j - d_{ij})$$

▨

Claim 3) Let client j be indirectly connected to i. Then, $d_{ij} < 3\alpha_j$



$\beta_{ij'} > 0$

$\beta_{i'j'} > 0$ $\Rightarrow$ $d_{ij'} \le \alpha_{j'}$, $d_{i'j'} \le \alpha_{j'}$

$\beta_{i'j} > 0 \Rightarrow d_{i'j} \le \alpha_j$

Facility i opens before i' (by construction.)

$\Rightarrow \alpha_j \geq \alpha_j'$ (increase at some rate and α_j' stops before α_j)

$$d_{ij} \le d_{i'j} + d_{i'j'} + d_{ij'} \le \alpha_j + \alpha_j' + \alpha_j' \le 3\alpha_j$$

$\Rightarrow \text{Cost}(\text{sol}) \le 3 \sum \alpha_j \le 3\ \text{OPT}(D)$

▤