COMPSCI 634: Geometric Algorithms 4/3/2014

Lecture #24: Geometric Set Cover

Lecturer: Pankaj Agarwal *Scribe: Chris Tralie*

1 Overview

The purpose of this lecture is to cover a few classic combinatorial optimization problems, including set cover, hitting set, and independent set, in a geometric context. Though the optimal set cover and hitting set problems are NP-hard, results from ε-nets help to give good approximation bounds for these algorithms for simpler set systems that arise in a geometric context.

The notes begin with some preliminary definitions of the dual range space and of the algorithms, and several results from ε-nets are reviewed. Then, a modified, weighted version of an ε-net is presented which leads to an approximation algorithm for the hitting set and set cover. After this, a new randomized hitting set algorithm with better bounds is presented, which was discovered by our very own T.A. Jiangwei Pan and professor Pankaj Agarwal. Finally, some basic ideas for independent set are shown in a geometric context

Throughout, I also highlight several open problems that were mentioned in class.

2 Dual Range Spaces

2.1 Definitions

Definition 1. *Let $\Sigma = (X, R)$ be a range space on the set X. Then*

$$\Sigma^T = (R, \{\{r_j | x_i \in r_j\} | x_i \in X\})$$

is the dual range space *associated with* Σ

In other words, the space becomes the set of ranges, and the ranges become sets of ranges that hit an $x \in X$. A slightly easier conceptualization of a range space for this purpose is a bipartite graph, where one set is the elements in X and the other set is the ranges, and there's a line between $x \in X$ and $r \in R$ if $x \in r$. The dual range space simply switches the roles of the two sets in the graph. Figure 1 shows an example with this construction.

Note also that if one constructs an incidence matrix A for the range space out of this bipartite graph, then A^T represents an incidence matrix for the dual range space. This makes the transpose symbol a natural choice for denoting the dual range space.

2.2 Geometric Example

One way to visualize dual range spaces in a geometric context is with points and rectangles. Define the following objects:

- X: Finite set of points in $\mathbb{R}^2$
- γ: A finite set of m rectangles $(\gamma_1, ..., \gamma_m)$

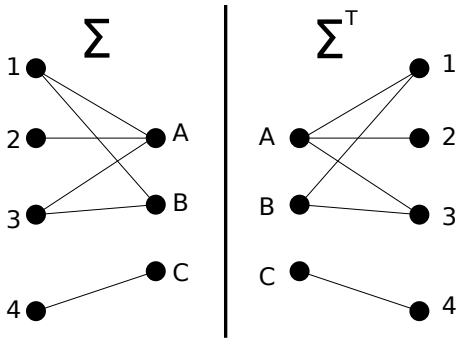

Figure 1: An example of a range space represented as a bipartite graph. The primal range space has $X = \{1,2,3,4\}, R = \{\{1,2,3\},\{1,3\}\{4\}\}$. The dual range space has $X = \{A,B,C\}, R = \{\{A,B\},\{A\},\{A,B\},\{C\}\}$

- $\Sigma : (X, \{\gamma \cap X | \gamma \in \Gamma\})$

  In other words, for each rectangle, create a range comprised of the points that are contained within that rectangle

- $\Sigma^T : (\Gamma, \{\{\gamma | x \in \gamma\} | x \in X\})$

  In other words, for each point, create a range out of the set of rectangles that contain it

  Figure 2 shows an example of such a space.

3 Geometric Hitting Set and Set Cover

3.1 Definitions

Definition 2. *For the range space* $\Sigma = (X,R)$, $H \subset X$ *is a* hitting set of Σ *if*

$$H \cap r \neq \emptyset \forall r \in R$$

Definition 3. *For the range space* $\Sigma = (X,R)$, $S \subset R$ *is a* set cover of Σ *if*

$$\cup_{s \in S} s = X$$

Note that the hitting set of a range space Σ is the same as a set cover of Σ^T. The goal is to find the smallest sized hitting set or set cover. Note also that the hitting set is closely related to an $\varepsilon -$ net. To see this, recall the definition of an $\varepsilon -$ net

Definition 4. $N \subset X$ *is an* $\varepsilon -$ net *of* X *if* $\forall r \in R$

$$|r| \geq \varepsilon |x| \implies N \cap r \neq \emptyset$$

If $\varepsilon = \frac{1}{N}$ in the definition of the the ε-net, where $N = |X|$ for the ranges space $\Sigma = (X,R)$, then the ε-net is certainly a hitting set for all of the ranges, because every range has size at least 1. Though the ε is quite small in this case, and it is only related to N, not to the optimal sized hitting set. Still, if one can improve the bound on ε, ε-nets may be useful for set systems of bounded VC-dimension because of the following theorem

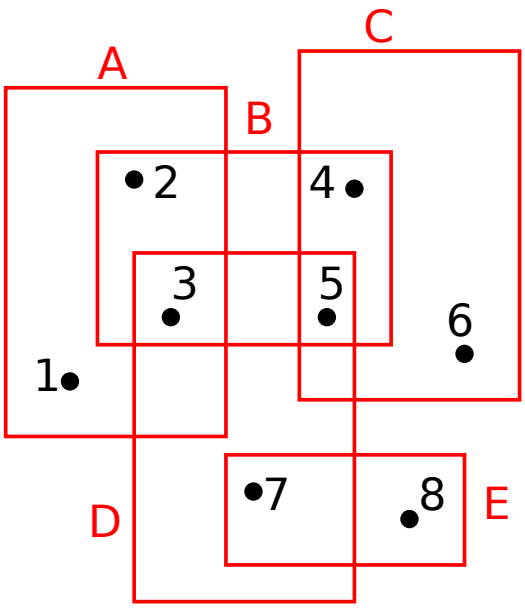

Figure 2: An geometric example of a range space. The primal range space consists of the points $X = \{1,2,3,4,5,6,7,8\}$ and the rectangles covering the ranges $R = \{A : \{1,2,3\}, B : \{2,3,4,5\}, C : \{4,5,6\}, D : \{3,5,7\}, E : \{7,8\}\}$. The dual range space consists of the rectangles $X = \{A,B,C,D,E\}$ and the points intersecting the rectangles $R = \{1 : \{A\}, 2 : \{A,B\}, 3 : \{A,B,D\}, 4 : \{B,C\}, 5 : \{B,C,D\}, 6 : \{C\}, 7 : \{D,E\}, 8 : \{E\}\}$

Theorem 1. *Given a range space $\Sigma = (X,R)$ with finite VC-dimension d, for any $\delta, \varepsilon > 0$, a random subset $N \subset X$ of size*

$$O\left(\frac{d}{\varepsilon}\log\frac{1}{\delta\varepsilon}\right)$$

is an ε-net of Σ with probability $\geq (1-\delta)$ [HP11]

Thus, the hope is to come up with better approximation algorithms for simple set systems using this theorem. As an example of where this may be useful, return to the ranges space with points and rectangles in Section 2.2. In fact, for this range space, an even better bound of size $O(\frac{1}{\varepsilon}\log\log\frac{1}{\varepsilon})$ has been shown recently in [AES10], while the ε-net of the dual range space is still $O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$ with constant probability.

As a side note, this implies that there is actually a gap in the bounds for computing set cover and the hitting set if ε-nets are used for the approximation.

3.2 An Approximation Algorithm

As mentioned before, the goal is to somehow reduce the hitting set to an ε-net. The main issue with the ε-net is that it is only guaranteed to cover heavy (high cardinality) ranges, but the hitting set requires all ranges to be covered, so before ε had to be set to a very small value $\frac{1}{N}$. To make this more convenient for the hitting set application, modify the definition of an ε-net to include weights for each element, so that small sets can effectively be given larger weights:

Definition 5. *For a range set $\Sigma = (X, R)$, define a map*

$$w : X \to \mathbb{Z}^+$$

And extend this maps to sets $S \in R$ so that

$$w(S) = \sum_{x \in S} w(x)$$

Then $N \subset X$ is a weighted ε-net of $(\Sigma = (X, R), w)$ *if* $\forall r \in R$

$$w(r) \geq \varepsilon w(x) \implies r \cap N \neq \emptyset$$

Also say that r is ε-light *if* $w(r) < \varepsilon w(x)$

Use this modified definition to devise an algorithm that estimates the weights w for a range space $(\Sigma = (X, R), w)$ that will lead to a good hitting set approximation with an ε-net. The algorithm is as follows:

Algorithm 1.

ALGORITHM HittingSetApprox1$(\Sigma = (X, R))$
Initially, $w(x) = 1 \forall x \in X$
while $\exists$ an ε-light range **do**
    Choose an ε-light range $r \in R$
    $w(x) \leftarrow 2w(x) \forall x \in r$
end while
return ε-net of (Σ, w)

The algorithm is very simple, but the analysis requires some tricks. To analyze this algorithm, let $w_i(x)$ be $w(x)$ after i iterations. Find an upper bound and a lower bound for $w_i(x)$. Also let H^* be an optimal hitting set algorithm of size k.

- To find an upper bound, observe that at each iteration, the weights of an ε-light range r are doubled. Since by definition $w_i(r) < \varepsilon$,

  $$w_{i+1}(X) = w_i(X) + w_i(r) \leq (1 + \varepsilon) w_i(X)$$

  Since all of the weights start off at 1, $w_0(X) = n$. Thus, the upper bound is

  $$w_i(X) \leq n(1 + \varepsilon)^i$$

- To find a lower bound, examine what happens to $w(H^*)$ after each iteration. Note that at each iteration, at least one element in H^* is doubled in weight. For the first k iterations, the minimum happens if these changes are spread out, so that a different element is doubled each time. Thus,

  $$w(H^*) \geq k + i$$

  Let $f(i) = k + i$ (spread the changes out evenly), and let $g(i) = k2^{i/k}$. Then $f(i) > g(i)$ over $[0, k]$, because $f(0) = g(0)$, $f(k) = g(k)$, $f'(0) > g'(0)$, and they are both convex functions. Also, each group of k iterations after the first k (for $i > k$), it is also true that the minimum is achieved by spreading the elements out. Therefore, the lower bound over all i elements doubled in weight is

$$w(H^*) \geq k2^{i/k}$$

To get $k = |H^*|$ involved in the upper bound, let

$$\varepsilon = \frac{\ln\sqrt{2}}{k}$$

, a choice which will become clear in a moment. Then

$$w_i(X) \leq (1+\varepsilon)^i n = \left(1+\frac{\ln\sqrt{2}}{k}\right)^i n \leq \exp\left(i\frac{\ln\sqrt{2}}{k}\right) n$$

Since $H^* \subset X$,

$$w_i(H^*) \leq w_i(X) \leq \exp\left(i\frac{\ln\sqrt{2}}{k}\right) n$$

Now combine the lower bound and the upper bound on $w_i(H^*)$

$$k2^{i/k} \leq n\exp\left(i\frac{\ln\sqrt{2}}{k}\right)$$

$$\ln(k) + \ln(2)\frac{i}{k} \leq \ln(n) + \ln\sqrt{2}\frac{i}{k}$$

In this step it is clear how clever the choice of $\varepsilon = \frac{\ln\sqrt{2}}{k}$ is (it allows us to subtract $\ln\sqrt{2}\frac{i}{k}$ from both sides of the inequality while maintaining a nonzero factor of $\frac{i}{k}$ on the left side)

$$\frac{i}{k}\ln(\sqrt{2}) \leq \ln\left(\frac{n}{k}\right)$$

$$i = O\left(k\log\frac{n}{k}\right)$$

The analysis so far has assumed the size of the optimal hitting set $k = |H^*|$, is known, but that information is not actually available up front. To estimate k, pick start with a small value of k (say 1), and do an exponential binary search, doubling k if the algorithm above doesn't converge in $(k/\sqrt{2})\log\left(\frac{n}{k}\right)$ steps.

When the algorithm finally terminates, the ε-net of the weighted range space (Σ, w) is an $O\left(k\log\frac{n}{k}\right)$ of the optimal hitting set of the ranges. In practice, to transform the weighted ε-net to an unweighted ε-net so that ordinary ε-net algorithms can be run, simply replicate the elements in (Σ, w) by their weights (this is why it was important that w be positive integer weights).

Open Question 1. *It is known that for a range space over points with ranges of discs, the size of the ε-net is $\Theta(\frac{1}{\varepsilon})$, so this algorithm gives a constant-sized approximation of the optimal hitting set for that special case. However, it is not known whether we can beat the above bound for the special case of points and rectangles*

3.3 Jiangwei and Pankaj's Approximation Algorithm

Algorithm 2.

ALGORITHM HittingSetApprox2($\Sigma = (X, R)$)

For the range space $\Sigma = (X, R)$

Let$\mu = \frac{2}{\ln 2} k \ln |R|^2 |X|$ (1)

Initially, $w(x) = 1 \forall x \in X$
Initially, $w(r) = 1 \forall r \in R$
for $i = 1$ to μ **do**

- Sample a random $\overline{x_i} \in X$ with the probability distribution $\Pr(x) = w(x)/w(X)$
- Sample a random range $\overline{r_i} \in R$ with the probability distribution $\Pr(r) = w(r)/w(R)$

$\forall r$ s.t. $x_i \in r, w(r) \leftarrow w(r)/2$
$\forall x \in r, w(x) = 2w(x)$
end for

Let $\Pi(x_i)$ be the number of indices $k \leq \mu$ where $\overline{x_k} = x_i$. Then a $1/(8k)$-net of the weighted range space (Σ, Π), is an $O(1)$ approximation of the optimal hitting set. More details can be found in [AP14], particularly in Section 4 of that paper.

4 Geometric Independent Set

The independent set problem asks for the largest set system such that each set is pairwise disjoint. This problem appears to be harder than hitting set and set cover to approximation. In particular, for some independent sets of size $n/2$, the best known polynomial approximation algorithm returns a set system within $\log^2 n$ size of the optimal.

One geometric example is, given a set of axis-parallel rectangles R, find the largest subset $S \subset R$ such that $\forall r_1, r_2 \in S, r_1 \cap r_2 = \emptyset$. An example is shown in Figure 3. An application of this example is to figure out how many city labels it is possible to display on a map without too much clutter (reduce to this problem by putting a bounding rectangle around each city label).

With the simpler example where all rectangles are unit-sized squares, a constant-factor approximation is possible with a simple greedy algorithm which takes a random square and removes the squares that intersect it, and repeats until there are no pairwise intersections. To extend this to squares of different sizes, do the same, but choose the squares to check in increasing order of size.

For rectangles, a $\log n$ approximation is possible with the following greedy algorithm:

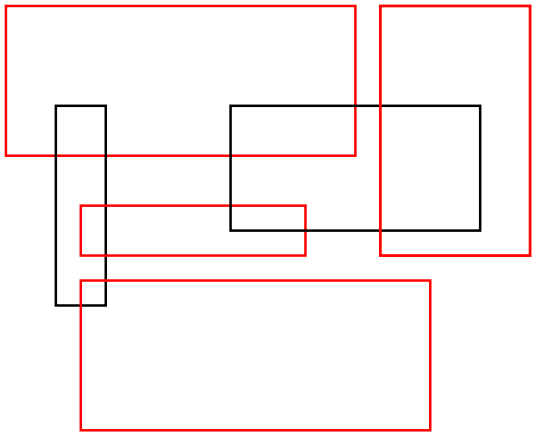

Figure 3: A geometric example of an independent set. The rectangles in the independent set are drawn with a red border

Algorithm 3.

ALGORITHM 2DIS(R)
Draw a vertical line l s.t. each side contains an equal number of vertices on the rectangles R
Define the sets

- $R_0 = \{r | l \cap r \neq \emptyset\}$
- $R^- = \{r | r$ lies to the left of $l\}$
- $R^+ = \{r | r$ lies to the left of $l\}$

Determine the independent set $I(R_0)$ with a 1D greedy algorithm.
Return $I(R) = I(R_0) \cup 2DIS(R^-) \cup 2DIS(R^+)$

It is also possible to approximate this problem by formulating it as an integer linear programming and then rounding, but this is slower.

Open Question 2. *Is there a simple $O(\log\log n)$ factor approximation for the independent set of axis-aligned rectangle problem?*

Open Question 3. *Is there a simple $O(1)$ factor approximation for the independent set of axis-aligned rectangle problem?*

References

[AES10] Boris Aronov, Esther Ezra, and Micha Sharir. Small-size \eps-nets for axis-parallel rectangles and boxes. *SIAM Journal on Computing*, 39(7):3248–3282, 2010.

[AP14] Pankaj K Agarwal and Jiangwei Pan. Near-linear algorithms for geometric hitting sets and set covers. *Proceedings of the 30th Annual Symposium on Computational Geometry*, 2014.

[HP11] Sariel Har-Peled. *Geometric approximation algorithms*, volume 173. American Mathematical Soc., 2011.