Lecture 11 and 12

*

September 13, 2010

Abstract

We will compute the intersection of some given half-planes i.e. region common to all half-planes by reducing it to convex hull problem. For this we will define a Duality transform for one to one mapping between intersection of Half-planes problem and Convex hull problem.In other words,we will solve this Intersection problem by reducing it to Convex hull problem.Then we will study some functions which can be used for this mapping.Then we will discuss how to find lower bound for some problems like sorting or Element Uniqueness problem with the help of decision tree.

1 Lecture-11:Intersection of Half Planes and Duality

Introduction

Given a set of n half-planes compute the intersection $\bigcap_{i=1}^{n} H_i$ i.e. region common to all half-planes. First we will define a Duality transform which can do one to one mapping between intersection of half-planes problem to convex hull problem.In other words,we will solve this Intersection problem by reducing it to Convex hull problem.

Half-Plane: A line divide any plane into two half-planes as shown in figure 1.

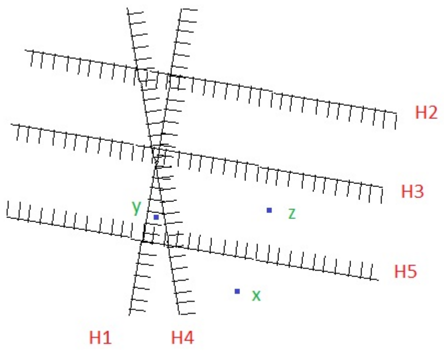


Figure 1 : Half-Planes

In figure 1 we have 5 half-planes. x and y are not common to half-planes because x is not

*Department of Computer Science and Engineering, IIT Delhi, New Delhi 1100116, India.

common to H_5 and y is not common to H_4. Point z is common to all half-planes. We are interested in common region to all half-planes. As clearly seen Half-plane is a convex set so intersection of half-planes will also be convex region unless it is:

1. Unbounded: We can easily bound this common region by using some kind of rectangular window as shown in figure 2.

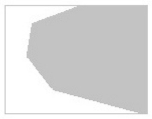
   

   Figure 2: Shaded Region is bounded by rectangular window

   This window should contain all intersection points of given half-planes or all corner points of common region. Thus we just need to include these four planes (constructing window) in given half-planes and the common region will always be bounded.
2. Empty: This is not easy to avoid so we will have to deal with the possibility that intersection may be empty.

Similarity with Linear Programming

As Half plane is represented by some inequality equation so finding common region to some given half planes is similar to finding solution in Linear programming.In L.P. we find one possible solution where in Intersection of half planes we are finding set of all such possible solutions.

No of equations in L.P. can be mapped to no. of half planes in Intersection and no. of variables in L.P. can be mapped to number of dimensions of planes.But in higher dimensions complexity of common region becomes very complicated.
For e.g. if we have n Half planes in 3-D whose intersection is not empty then what should be the complexity of common region. As described above common region is a convex set so a plane can describe only one phase so number of phases are bounded by no of planes i.e. n.So it would have maximum n phases, still a linear structure. but for higher dimensions than 3-D its complexity increase exponentially and is given by $\Theta(n^{\lfloor d/2 \rfloor})$. In terms of L.P. we can say that this is the reason we only find one possible solution in L.P. because its not feasible to find all possible solutions.

Finding Intersection of H.P.

1. First distinguish between Downward pointing planes and upward pointing planes as shown in figure 3.

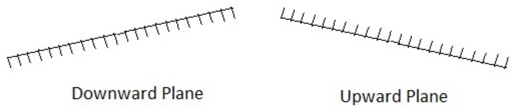


Figure 3

As H.P. are represented by some inequality so H.P. which are satisfied by $+\infty$ will be upward pointing H.P. otherwise downward pointing planes.we assume that we don't have vertical planes for now as this situation can always be avoided by rotating the axis.

2. Find intersection of all upward pointing planes.
3. Find intersection of all downward pointing planes.
4. Find the intersection of both regions found in step 2 and 3.

Algorithm in Detail

Step 4 : This can be done in $O(n)$ time.

We are given two regions one common to downward pointing half planes say R_1 , second common to upward pointing half planes say R_2.This method is based on the assumption that the points of R_1 and R_2 are given ordered.

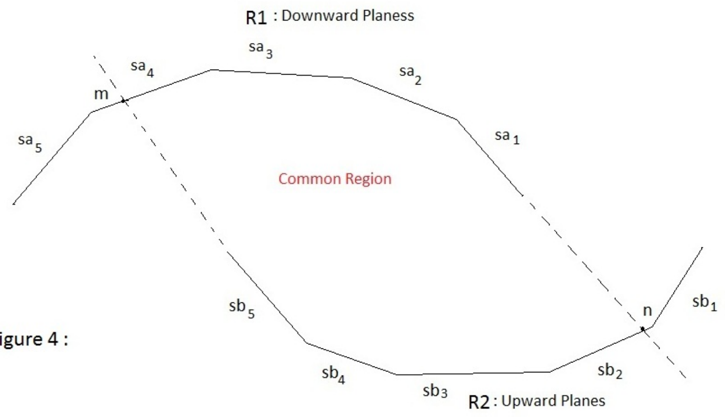


Figure 4 :

As can seen in figure 4, these both regions will intersect at exactly two points(m and n here) if intersection is not empty.we just need to find these two points to find common region.Number the sides of both regions as shown. for R_1 sa_1,sa_2,sa_3,..... and so on and for R_2 sb_1,sb_2,sb_3,..... and so on. Say la_i correspond to line for segment for sa_i likewise lb_i for sb_i. As we know that these regions are convex so we can say that that R_1 sa_1,sa_2,sa_3,.... are ordered in non decreasing slope and similarly sides of R_2 are numbered in non increasing slope.

Algorithm to find intersection point between R_1 and R_2

1. Initialize i and j to 1.
2. Check whether sa_i and sb_j intersect or not.if yes find intersection point and terminate.

3. Check whether la_i and sb_j intersect or not. Now there are two cases :

Case I: if they does not intersect then move to next line segment i.e. if segment is from R_2 increase j by 1 else increase i by 1 and go back to step 3.

Case II: If they do intersect then Go back to step 3 but change line segment to line and line to line segment i.e. if we were comparing la_i with sb_j now we will compare sa_i with lb_j

Analysis of Algorithm

Complexity: It can be easily seen time is proportional to total number of sides in both region i.e. $O(n)$

Termination: After executing step 3 two times we move to next side in any region at least once so it will terminate.

Correctness: (a) say sa_i and sb_j intersects.

(b) As every time we are comparing one line with other line segment so say we reach at sa_i or la_i sometime during execution. now there are following cases to consider:

i. if second line or line segment we choose is sb_j or lb_j then we are done. we will find the intersection point between sa_i and sb_j .

ii. if we choose lb_k or sb_k where k<j then there will be two case :

A. we are comparing sa_i and lb_k :now we claim that if we come to this condition then sa_i and lb_k will intersect and so according to algorithm we will be incrementing k till we compare sa_i and sb_j.

proof : Assume the contrary. now there are two cases :

Case I: sa_i lie below lb_k as shown in figure 5. then lb_j will also lie above sa_i so sa_i and lb_j will not intersect which will be a contradiction.

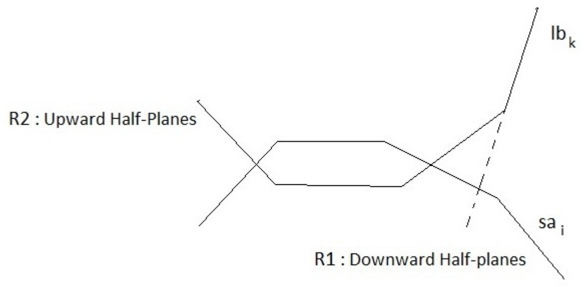


Figure 5

Case II: sa_i lie above lb_k as shown in figure 6. now we can see that there was some sa_n,where n<i such that lb_k will pass through it.In that condition algorithm would have proceeded with comparing la_n and sb_k and in that case sb_k will be shifted to $sb_{(k+1)}$ which contradict the situation where we are comparing sa_i and lb_k.

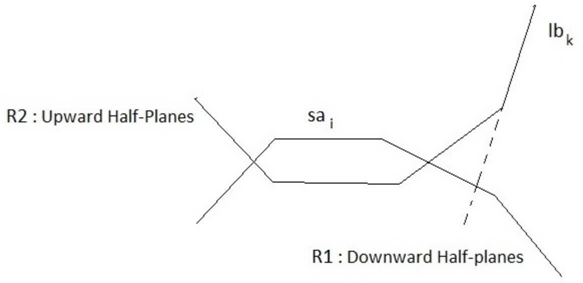


Figure 6 :

B. Else we are comparing la_i and sb_k so according to algorithm sb_k will be shifted to $sb_{(k+1)}$.

iii. if we choose lb_k or sb_k where k>j it means we must have passed through a situation when we were comparing la_x or sa_x with lb_j or sb_j where x< i and with explanation similar to given above it can be shown that current situation would not have been possible in such a scenario.

Step 3 : Finding intersection of all downward pointing planes

This intersection will never give us empty region because at least there is a one point which is common to all downward half-planes,$-\infty$.
We will do this by reducing it to convex hull problem.For this reduction we need a kind of one to one mapping from Intersection of half-plane problem to Convex hull problem.

so Let's define a relation D which provides us this Dual Mapping.

$$\text{D(point)} \rightarrow \text{line and D(line)} \rightarrow \text{point}$$

we represent point in 2-D by (a,b) and line by (m,c) where m is slope of line and c is its y intercept.

Desirable properties of D

we have a point p and line L.

1. D is one to one mapping from 2-D to 2-D
2. D(D(p))=p and D(D(L))=L
3. Incidence property : If p is incidence on L (i.e. L pass through p) then D(L) is incidence on D(p).
4. If L_1 and L_2 intersect at p then D(p) should pass through D(L_1) and D(L_2). As clearly seen this property directly follows from incident property.
5. Above-below property : If p lies above L then D(L) lies below D(p).

Mapping of Intersection problem to Convex Hull problem

Given a set H of downward pointing planes let I(H) denote intersection which is a convex region.Consider the duals of lines describing half planes H,denote it by S.Let CH(S) denote the Convex Hull of S.

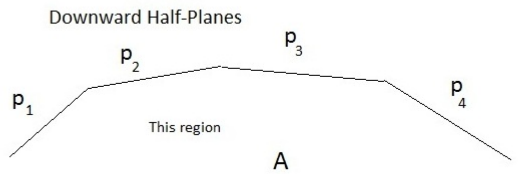


Figure 7:

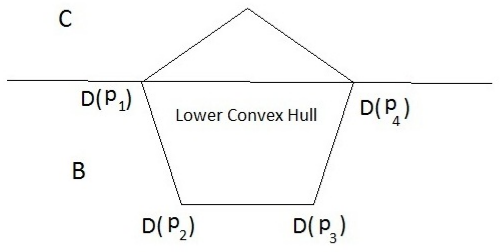


Figure 8 :

Now the claim is there is one to one mapping between A and B in figure 7 and figure 8. i.e. The downward convex chain describing I(H) is in a one to one correspondence with the points(duals of line describing H) on lower Hull.So to find I(H) first apply D to all planes,now these are converted to points. find convex hull of these points and again apply D to points of this lower Convex Hull to get back I(H).
Why this claim is true ?
we will prove this later. While finding intersection of downward half planes we get a convex region, but some H.P. may not be part of this output region. we need to recognize the H.P. which will surely be part of output region.

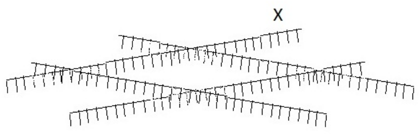


Figure :9

As shown in figure 9 H.P. x will not be the part of output.Actually only those H.P. will be part of output which have a minimum y at any x. i.e. if we move from left to right those planes which have a lowest point(minimum y) at any x will be included in output.when this problem is mapped to Convex Hull problem, there are some points which lie in convex hull and are not at boundary of convex hull these are duals of those half planes which are not part of output in intersection.

Proof for one to one mapping

Now we will prove this one to one mapping of downward convex chain with lower convex hull.

As we know any plane which have lowest depth at any position is part of output so it should have a mapping to lower convex hull. say we have such a plane given by line L it has lowest depth say between two points p1 and p2 as shown in figure 10.

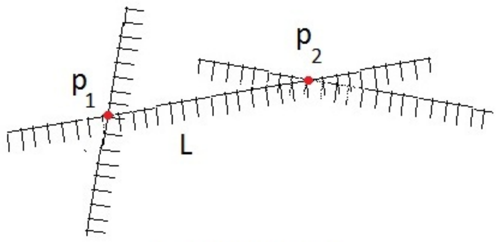


Figure 10 : L is a line

Thus line D(p1) and D(p2) intersect at P(L)(Incidence property).Now any point p on L between p1 and p2 (shown in figure). All lines except L, say L' lie above point p.Thus there exist a line D(p) passing through D(L) and is below all those points given by D(L')(Above-Below property). i.e. D(p) is a tangent to D(L) and D(L) is point on lower convex hull.

Now consider the dual environment, other direction of proof i.e. if there is a point on lower convex hull it will be mapped to half plane which must be part of output i.e. it will have a lowest depth at any point.

Draw a tangent T from any point q to convex hull such that all other points lie above it.say it intersect convex hull at p as shown in figure 11.

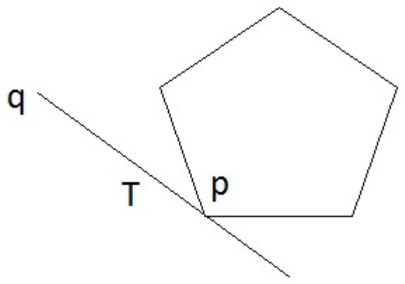


Figure 11: Tangent T with p and q points

All points of convex hull except p, say p' lie above Thus its primal counterpart D(p) is a line that pass through at least one point D(T) thats is lower than all other lines D(p').Thus D(p) half-plane attain lowest depth at some point so it will be part of output.
So those half-planes which are not part of output will be mapped to those points which lie inside convex region.

2 Lecture-12: Duality(continued...)

Introduction

We will see some functions which can provide us the mapping given above.Then we will discuss how to find lower bound for some problems like sorting or Element Uniqueness problem with the help of decision tree.

Function which can be used to provide D transform

we have point p :(a,b) and a line L :(m,c). $D(p)$ will be given by $(2a, -b)$ i.e. D(p) correspond to line $y = 2ax - b$ and D(L):$(y = 2ax - b)$ $\rightarrow$p:(a.b) Now we need to verify whether this function satisfies required properties or not.

Property 1: D(D(p))=p and D(D(L))=L
D(p)=$(2a, -b)$ $\Rightarrow$ D(D(p))=$(2a/2, -(-b))$=(a, b)
similarly it can be proved for line.

Property 2: Incidence property :
Suppose p:(a, b) lies on L:$(y = mx + c)$,which gives

$$b = am + c \tag{1}$$

D(p) is given by line:$(2a, -b)$ i.e. $y = 2ax - b$ and D(L) is given b point $(m/2, -c)$. it can be easily shown with the help of Equation 1 that D(L) lies on D(p).Hence proved.

Property 3: Above-below property
Say p lies above L,so p satisfies equation $y > mx + c$, which gives

$$b > am + c \tag{2}$$

Now D(L) should lie below D(p) $\Rightarrow$ point$(m/2, -c)$ should lie below $y = 2ax - b$ $\Rightarrow$ $p : (m/2, -c)$should satisfy equation $(y < 2ax - b)$ *or* $-c$ should be less than $(am - b)$ *or* b should be greater than $(am + c)$ which is true from equation 2 .Hence proved.

Dual transform using parabola

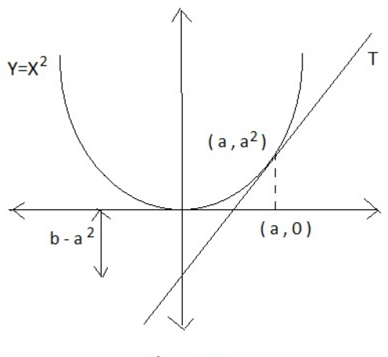


Figure 12

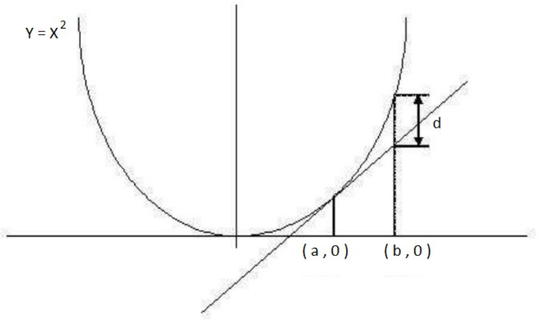
As shown in figure 12, we have a parabola $y = x^2$.The equation of tangent to parabola at point (a, a^2) will be $y = 2ax - a^2$. Now for any point p lying on parabola D(p) will be tangent to parabola at that point i.e. D$(a, a^2) \rightarrow y = 2ax - a^2$ which give same result as given by mapping function above.For points above parabola dual will be below parabola and like-wise for points below parabola.

In higher dimensions $(x_1, x_2, ...x_d)$ with paraboloid $x_d^2 = x_1^2 + x_2^2 + ... + x_{d-1}^2$

D$(a_1, a_2, ..., a_d)$= $2a_1x_1 + 2a_2x_2 + ... + 2a_{d-1}x_{d-1} - a_d$

Another nice property of parabola

So we take a point a in 1-D, then a is mapped to the tangent $y = 2ax - a^2$ to the parabola $y = x^2$ in 2-D. Let b be any other point, and d defines the vertical distance of the tangent at (a, a^2) from the parabola $y = x^2$ at the point (b, b^2) as shown in the figure 13.



$d = a^2 - (2ab - b^2) = (ab)^2 = dist^2(a, b)$ in 1-D
Hence, we can see that d can be used as a measure to compare the distance between two points in 1-D. Similar is true for higher dimensions and can be proved easily.So this method is used to create the voronoi diagram in d-dimension by computing the intersection of half-planes in $(d+1)$ dimension. Each point in the input set in d-dimension is mapped to a half-plane in $(d + 1)$ dimension

Lower bounds for Geometric Problems

Problem : Element Uniqueness :It is a decision problem.Given n elements $(x_1, x_2, ..., x_n)$,determine in all of them are unique or not. Some simple cases :

1 It is given that $x_i \in \{1, 2, 3, ..., n\}$.then using hashing and a array of size n it can be done easily in $O(n)$ time.

2 It is given that $x_i \in \{1, 2, 3, ..., n^2\}$.As we know range of elements,we can sort these n elements using Radix sort which will again take $O(n)$ time.then in one phase we can check for duplicate elements because if present, duplicate elements will be adjacent.

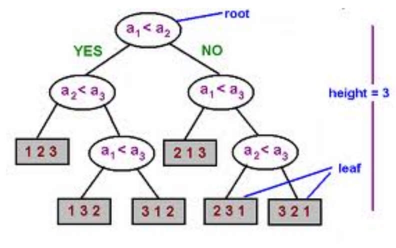
3 if $x_i \in \mathbb{R}$.as we can always sort the given number and if there are duplicate numbers they will be adjacent in sorted order.So E.U. is reducible to sorting hence lower bound of E.U. applies to lower bound of sorting. but can we solve E.U. faster without using sorting.

Before proving lower bound we need to specify model of computation.A model of computation specifies

1. What are the operations that can be performed ?
2. What operations can be counted as unit operation ?

Lower bound of sorting is found assuming certain model of computation : Comparison Based Model

Comparison Based Model



In comparison tree model we can only do comparison operations.The comparison operations are $\leq, \geq, <, >, =$. Comparison sort can be viewed abstractly in terms of decision trees.A decision tree is a fully binary tree that represents the comparison between elements that are performed by particular sorting algorithm on given input.The execution of algorithm corresponds to tracing a path from root to leaf.Comparison decides our path but answer is given at leaf.As sorting may have $n!$ permutations and correct algorithm should be able to generate any permutation so decision tree for that algorithm should have at least $n!$ leaves which say that height of tree or execution time(in worst case) must be of order $\Omega\ (n \log n)$

Linear Decision Tree

This model is chosen for E.U. problem.It allows operations only dealing with linear inequality. A general inequality could be $a_1x_1 + a_2x_2 + b \geq 0$.
As we are dealing with decision problem so we will be using decision tree so answer given at leaf will be only yes/no. some leaves will give yes as answer and some of then will be give no.To find lower bound, we just need to bound somehow number of leaves.

Lets state the problem again.
Given n elements $(x_1, x_2, ..., x_n)$, determine whether all of them are unique or not.
Assume input is a n tuple i.e. think of it as a single point in n dimensions.so input space will be $\mathbb{R}^n$. In $\mathbb{R}^n$ some subset say W $\subset \mathbb{R}^n$ contain all yes answer and the complemented part $\mathbb{R}^n$-W are the no answers. i.e. we will group all those points whose all n co-ordinates are different into W,these points are yes answers i.e. n coordinates given by these points are unique and rest of the points will correspond to leaves with no answer

For e.g. Let's see how does W look like in 2-D

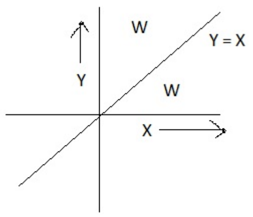


Figure 15 :

As shown in figure 15, $y = x$ is the area which correspond to leaves with answer no,rest of area will give yes as answer.

Similarly in n-dimensions, hyper-planes $x_i = x_j$, where i≠j will correspond to leaves with answer no. so union of all such hyper planes will be set of all leaves corresponding to no answer.

References