Note: These lecture notes are based on the textbook "Computational Geometry" by Berg et al.*and lecture notes from [3], [1], [2]*

1 Halfplane Intersection Problem

We can represent lines in a plane by the equation

$$y = ax + b$$

where a is the slop and b the y-intercept. Note that this can not represent vertical lines, which have infinite slope (but these do not occur in general position)

Every non-vertical line defines two *halfplanes*, consisting of the points above and below the line.

- *lower halfplane:* $y \leq ax + b$
- *upper halfplane:* $y \geq ax + b$

1.1 Halfplane Intersection

Halfplane Intersection Problem: Given a collection $H = \{h_1, \dots h_n\}$ of n closed halfplanes, compute their intersection

Note that a halfplane is a convex set so the intersection of any number of them is also convex. Unlike the convex hull, the intersection of halfplanes may be empty or unbounded.

A good way to represent the output would be to list the lines bounding the intersection in clockwise order.

How big can this output be? By convexity, each halfplane can appear only once as a side. Hence number of sides is $O(n)$.

Lowerbound? Can show via sorting it is $\Omega(n \log n)$. [**Jared:** *good exercise*]

Motivation: Generate convex shape approximations. Optimizations problems (e.g. linear programming, gradient descent) often occur over intersection of halfspaces.

Higher dimensions: This can be challenging since the number of facets in the intersection of n halfspaces in d-dimensions can be $\Theta(n^{\lfloor d/2 \rfloor})$.

We'll discuss two algorithms: (1) divide-and-conquer; and (2) duality based.

2 Divide-and-Conquer for Half-Plane Intersection

procedure DIVIDE-AND-CONQUER(H)
  If $n = |H| = 1$, return the halfplane
  Partition the halfplanes into H_1 and H_2 of size $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$
  Let K_1 and K_2 be the intersections of H_1 and H_2, computed ecursively
  Return the intersection of K_1 and K_2
end procedure

If we can perform the intersection (last step) in $O(n)$ time then the algorithm will have runtime given by $T(n) = 2T(n/2) + O(n)$, which will give $T(n) = O(n \log n)$.

Thus, we want to intersect two convex (possibly empty or unbounded) polygons in linear time.

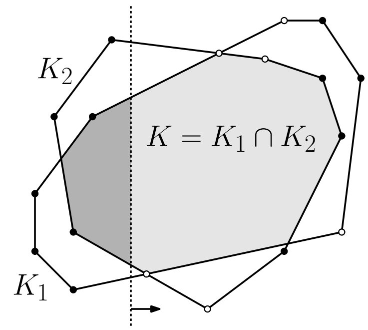

Figure 1. Plane Sweep Intersection of Polygons (from Berg et al.)

2.1 Plane Sweep

We compute the intersection of K_1 and K_2 via a plane sweep.

First, break both polygons into upper and lower chains. The *upper chain* of a polygon is just the sequence of points that are above the line induced by the leftmost and rightmost point in the polygon. The *lower chain* is the sequence of points below this line. So this step can be done in $O(n)$ time.

Next, sweep! By convexity, the sweep line intersects each polygon in at most two points. Hence, there are at most 4 points to consider at any point during the sweep. All we need to do is output the extremal points at each "event" (e.g. highest and lowest).

What about the total number of "events" that can happen as we sweep from left to right? The next event can be either the rightmost endpoint of one of the polygon edges stabbed (4 possibilities) or it can be the intersection of two edges (4 possibilities). Thus, we encounter $O(n)$ events; each event can be processed in $O(1)$ time (e.g. find the upper or lower point to output); and the next event can be computed in $O(1)$ time. Hence, total time is $O(n)$.

3 Envelopes and Duality for Half-Plane intersection

3.1 Lower Envelope Problem

We are given n lines $\{\ell_1, \ldots \ell_n\}$. The *lower envelope* is the intersection of their lower halfplanes. The *upper envelope* is the intersection of their upper halfplanes.

3.2 Duality Definition

- Given any point $p = (a, b)$, let p^* be the line $ax - b$.
- Given any line $\ell = ax + b$, let ℓ^* be the point $(a, -b)$

Primal and Dual planes

- **Primal plane:** Original (x,y) plane
- **Dual plane:** The new (a,b) plane

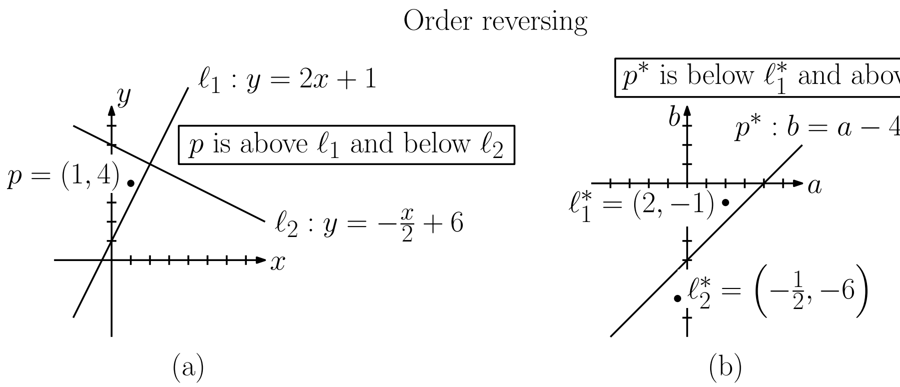

Figure 2. Order-Reversing Property of Duality (from Berg et al.)

3.3 Duality Properties

- **Self-Inverse:** $p^{**} = p$
- **Order Reversing:** p is above/on/below ℓ in the primal plane iff p^* is below/on/above ℓ^* in the dual plane.
- **Preserves Vertical Distances:** The vertical distance between p and ℓ equals the vertical distance between p^* and ℓ^*
- **Intersection preservation:** Lines ℓ_1 and ℓ_2 intersect at point p in the primal plane iff the dual line p^* passes through points ℓ_1^* and ℓ_2^*
- **Colinearity/Coincidence:** Three points are collinear in the primal plane iff their dual lines intersect at the same point.

Order Reversal Property To show point p is above line ℓ iff line p^* is below point ℓ^*

- Point $p = (q, r)$ is above the line $\ell = sx + t \iff r \geq sq + t$
- $r \geq sq + t \iff qs - r \leq t$
- $qs - r \leq t \iff$ line $p^* = qx - r$ is below point $\ell^* = (s, t)$.

3.4 Convex Hulls and Envelopes

Lemma: Let P be a set of points in the plane. Then the ccw order of the upper convex hull of P equals the left to right order of the lines on the lower envelope of the dual P^*. Symmetrically the order in the lower convex hull of P is the left-to-right order of lines in the upper envelope.

Proof: Assume no three points are co-linear. Consider two consecutive points q and r on the upper hull and let ℓ be the line through these two points. By definition of the hull, ℓ is above all points in P. Thus, by the order reversing property, all dual lines of P^* pass *above* point ℓ^*. This means that point ℓ^* lies on the lower envelope. Next, consider the dual lines q^* and r^*. By the incidence property, the dual point ℓ^* is the intersection of these two lines. (By general position, we assume the two points have different x-coordinates and so q^* and r^* have different slopes). Thus, the edges in the lower-envelope associated with the point ℓ^* are determined by the lines q^* and r^*.

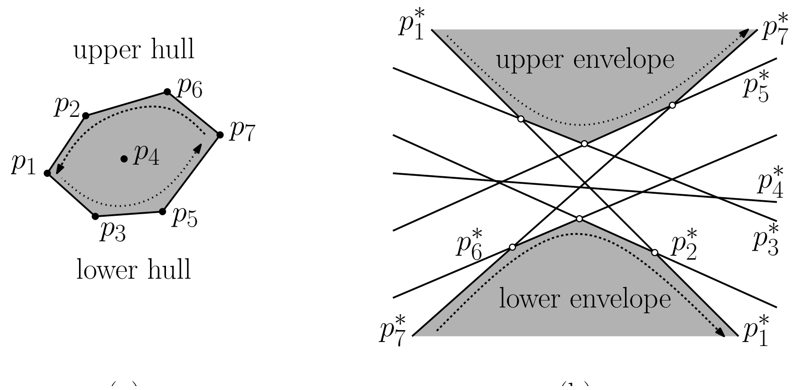

Figure 3. Envelope/Convex Hull Duality (from Berg et al.)

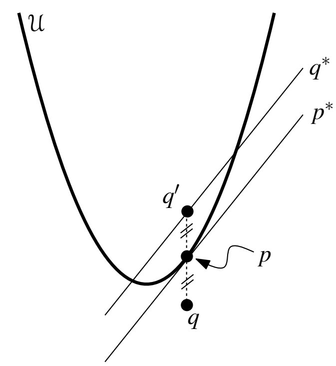

Figure 4. Duality: a geometric view ([2])

It remains only to show that the points in the lower envelope are ordered left-to-right. Note that as we move ccw along the upper hull (i.e. right-to-left), the slopes of the edges increase. Since the slope of the line in the primal plane determines the x-coordinate of the dual point, this means that we visit the lower envelope from left-to-right.

The proof for the upper envelope follows by symmetry. □

Note that we can perform half-plane intersection via computing the upper and lower envelope problems as follows. Given a set of half-planes to intersect, first find the upper envelope of the half-planes that are open to the top, then find the lower envelope of the half-planes that are open to the bottom. Then merge these two envelopes (in $O(1)$ time), by computing the leftmost and rightmost intersection points.

3.5 Duality: A geometric View

Let $\mathcal{U}$ be the parabola $y = x^2/2$. Consider some point (p_x, p_y) on $\mathcal{U}$. Note that $p^* = p_x x - p_y$. Then:

- p^* has same slope as the tangent line, since the derivative of $\mathcal{U}$ at p is p_x.
- p^* has same y-intercept as tangent line, since the tangent line intersects the y-axis at $(0, p_x^2/2)$.
  - the tangent line equation is: $p_y = p_x(p_x) - b$, which implies that $b = p_x^2 - p_y = p_x^2 - p_x^2/2 = p_x^2/2$.

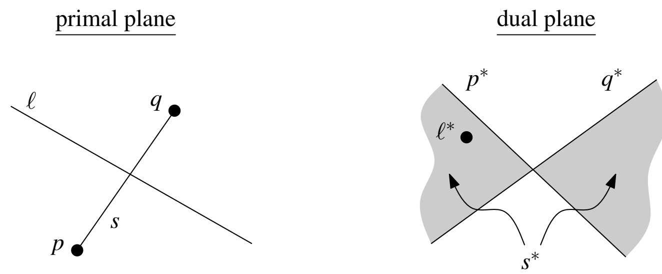

Figure 5. Dual of a line segment is a wedge ([2])

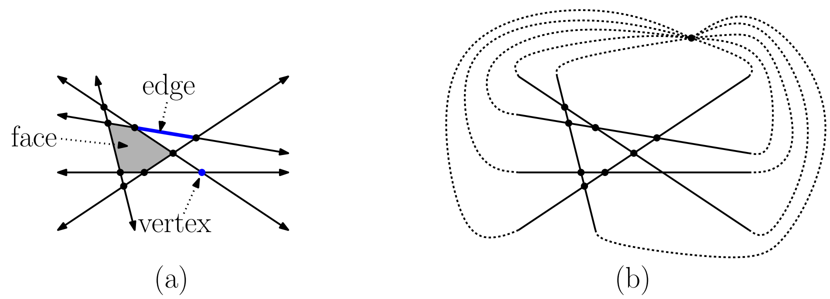

Figure 6. Basic elements of an arrangement. Adding a vertex at infinity creates a proper planar graph ([1])

Thus for p on $\mathcal{U}$, p^* is exactly the tangent line at p. What about for any point $q = (q_x, q_y)$ not necessarily on $\mathcal{U}$?

- q^* has the same slope as the tangent line at q_x
- q^* intersects the point $(q_x, q_y - (q_y - q_x^2/2))$

3.6 Some Observations

- Duality can be applied to objects other than lines and points. For example, we can take the dual of a line segment.
- The dual of a segment is a double wedge. See Figure 5.
- Q: What line would dualize to a point in the right side of the figure?

4 Arrangements

An application of duality. In a set of n points, are there 3 that are on a line. Naive algorithm takes $O(n^3)$ time. Faster algorithm uses duality and *arrangements*. Detecting 3 points on a line dualizes to detecting 3 lines intersecting at a point!

4.1 Arrangement Definition and Complexity

An *arrangement* of lines consists of faces, edges and vertices that are created via a set of n lines in the plane.

Combinatorial complexity.

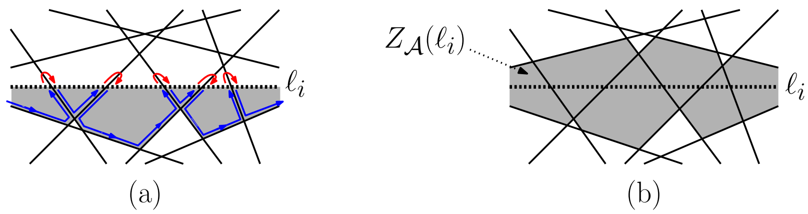

Figure 7. (a) adding ℓ_i to an arrangement; (b) the zone of line ℓ_i ([1])

Lemma: An arrangement of n lines in the plane has:

- $\binom{n}{2}$ vertices
- n^2 edges
- $\binom{n}{2} + n + 1$ faces

Proof: First note that the number of vertices equals the number of intersection points of lines. This is at most $\binom{n}{2}$ vertices (worst case is where every pair of lines intersects, which happens in general position).

Next we prove the bound on the edges by induction. Base case is trivial. Inductive step: When we add a new line to an arrangement of $n-1$ lines, we start with $(n-1)^2$ edges by the IH. Then the new line splits exactly one edge from each of the existing lines. Also, the new line is split into n edges by its intersection with the other lines. Thus, the total number of edges in the arrangement with the new line is $(n-1)^2 + (n-1) + n = n^2$

The number of faces follows by Euler's formula $v - e + f = 2$ □

Note that we can represent the arrangement fully as:

- The vertex and edge graph as shown in Figure 6
- A pointer for each edge to the two faces that edge is in
- For each face, a linked list that maintains the edges of the face in CCW order

Thus for $d = 2$, the complexity of an arrangement is $O(n^2)$. It turns out that , for all $d \geq 2$, the complexity of an arrangement is $O(n^d)$.

4.2 Building an Arrangement

We build an arrangement by adding one line after another. Let $L = \{\ell_1, \ldots \ell_n\}$ denote the set of lines. We show that line ℓ_i can be added in $O(i)$ time. This gives a $O(n^2)$ time algorithm for building the entire arrangement.

Suppose the first $i-1$ lines have been inserted and consider the insertion of ℓ_i.

First, determine the leftmost unbounded face containing ℓ_i. Assume that at $x = \infty$, the lines are sorted by increasing slope. Then can find in $O(i)$ time where ℓ_i falls in terms of slope. This determines the leftmost face containing ℓ_i.

Next, the line ℓ_i cuts through a bunch of edges and faces that we need to split up. To determine which edges are cut, we "walk" the line through the arrangement from one face to the next. When we enter a face, we need to determine through which edge ℓ_i exits that face. We do this in a brute-force manner by just walking the edges in ccw order till we find the edge at which ℓ_i exits the face. We then jump to the face on the other side of that edge. See Figure 7(a).

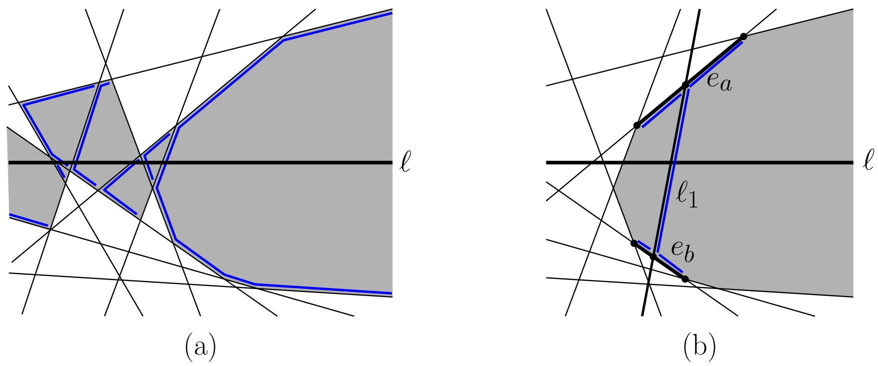

Figure 8. (a) all left bounding edges of ℓ; (b) rightmost face intersected by ℓ and the left bounding edges created by ℓ_1.([1])

Problem: Naively, we can say that we encounter $i-1$ lines, and hence pass through i faces. Since each face is bounded by at most i lines, we take $O(i^2)$ time. But this is slower than what was claimed. To improve the analysis, we need the zone theorem.

4.3 The Zone Theorem

Given an arrangement $\mathcal{A}(L)$ and a line ℓ not in L, the *zone* of ℓ in $\mathcal{A}(L)$, denoted $Z_{\mathcal{A}(\ell)}$, is the set of faces of the arrangement that are intersected by ℓ. The Zone theorem below states that the complexity of the zone (i.e. number of edges of all faces in the zone) is $O(n)$. The proof below is based on the proof from [1].

Theorem: Given an arrangement $\mathcal{A}(\mathcal{L})$ of n lines in the plane, and given any line ℓ in the plane, the total number of edges in all the faces that ℓ intersects is at most $6n$.

Proof: For simplicity, rotate the plane so that ℓ is horizontal. We call an edge a *left bounding edge* for the face lying to the left of it, and a *right bounding edge* for the face lying to the right (Figure 8(a)). Otherwise, call the edge *right bounding*. We show that there are at most $3n$ left bounding edges. A symmetrical argument works for right-bounding edges.

Base Case: $n = 1$, there is exactly one left bounding edge in ℓ's zone, and $1 \leq 3n = 3$.

Inductive Step: Consider an arrangement of n lines and let ℓ_1 be the line that intersect ℓ at the rightmost point. By the inductive hypothesis, there are at most $3(n-1)$ left-bounding edges in the zone for ℓ in the arrangement $\mathcal{A}(L - \ell_1)$.

Consider the rightmost face intersected by ℓ and note that all edges of this face are left-bounding edges. Line ℓ_1 intersects ℓ within this face and, by convexity, it intersects the boundary of this face in two edges, call them e_a and e_b. The insertion of ℓ into this face creates a new left bounding edge along ℓ, and also splits e_a and e_b into two new left bounding edges for a net increase of 3 edges (Figure 8(b)). Note that ℓ_1 can not contribute any other left-bounding edges to the zone: to the left of ℓ_1, the edges induced by ℓ_1 can form right-bounding edges only; to the right of ℓ_1, all other faces possibly touched by ell_1 are shielded from ℓ by either the line supporting e_a or the line supporting e_b.

Thus, the total number of left bounding edges on the zone of ℓ in $\mathcal{A}(L)$ is at most $3(n-1)+3 \leq 3n$. $\square$

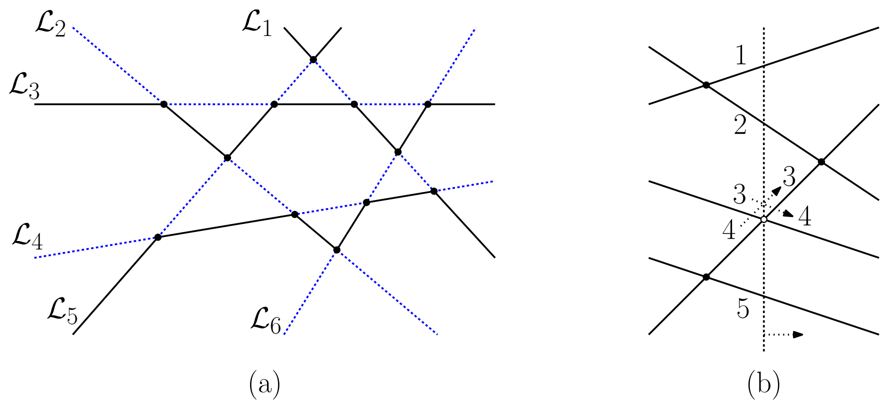

Figure 9. (a) Levels of an arrangement. (b) rightmost face intersected by ℓ and the left bounding edges created by ℓ_1.([1])

5 Some Applications

5.1 Three Points on a line?

Given a set P of n points, in $O(n^2)$, we can determine if there are 3 points on a line (i.e. are the points in general position?).

Algorithm:

1. Create the arrangement for the lines in P^*
2. If 3 lines in P^* intersect at the same vertex, then the points associated with those 3 lines in the primal plane are co-linear.

This takes just $O(n^2)$ time to compute the arrangement!

5.1.1 Levels of an Arrangement

We say that a point is at *level* k (denoted $\mathcal{L}_k$) in an arrangement if there are at most $k-1$ lines above the point and at most $n-k$ lines below. See Figure 9(a). The upper envelope of the lines is level 1 of the arrangement, and the bottom envelope is level n. Note that each point in the arrangement is generally on two levels, and that the levels are piecewise linear.

We can compute the levels of an arrangement in $O(n^2)$ time as follows. First we sort all the lines via slope. Next, for each line, ℓ in the arrangement, we first compute the level of the leftmost vertex ($x = -\infty$) of ℓ in this sorted order in $O(n)$ time. Next we walk along ℓ from left to right using the edge lists. We maintain the level as we walk. The level only changes at a vertex and its change can be computed by inspecting edges incident to that vertex.

Alternatively, we can use plane-sweep from left to right to do this in $O(n^2 \log n)$ time (keeping "events" in a priority queue - hence the $\log n$ factor). Whenever, we come to the next vertex, we swap the level numbers associated with the two lines at the intersection. See Figure 9(b). [1]

5.2 Ham-sandwich algorithm

We now discuss a discrete version of the celebrated "ham-sandwich" theorem.

[1] The $\log n$ factor can be eliminated via topological plane sweep, beyond the scope of this lecture.

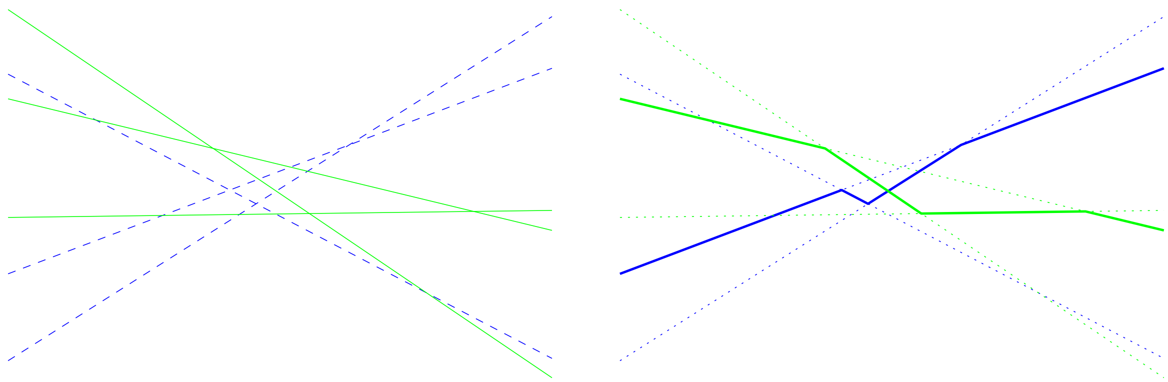

Figure 10. An arrangement of three green lines (solid) and three blue lines (dashed). The median levels for the green and blue are marked in bold on the right. ([3])

Given: R and B are two finite sets of points.

Goal: Output a line that *bisects* both R and B. That is in either open half-plane defined by the line, there are no more than $|R|/2$ points from R and no more than $|B|/2$ points from B.

Theorem: We can always find such a line in $O(n^2)$ time.

Proof: Assume $|R|$ and $|B|$ are both odd. If either set is not, remove an arbitrary point from it. Then, a bisector of the remaining sets will also bisect the original sets.

Now create an arrangement of both $\mathcal{A}(R^*)$ and $\mathcal{A}(B^*)$, and compute the levels for both the red and blue points. Let the *median* level for the red points be the level $(|R|+1)/2$, and note that it defines bisecting lines for R. The leftmost and rightmost segment of this median level are defined by the same line, ℓ_r, since slope alone determines the ordering of the lines at $\pm\infty$. Similarly, there is a line ℓ_b that determines the leftmost and rightmost segment of the median level for $\mathcal{A}(B^*)$.

Note that ℓ_r and ℓ_b have different slopes since no two point in $R \cup B$ have the same x coordinate.[2] Thus, since the median levels for $\mathcal{A}(R^*)$ and $\mathcal{A}(B^*)$ are piecewise linear functions, these median levels must also intersect somewhere in $\mathcal{A}(R^* \cup B^*)$ (see Figure 10(right)). In the primal plane, this intersection point represents a half-plane that bisects R and B. $\square$

5.2.1 Discussion

The continuous version of the theorem says that if we've got n objects in n dimensional space, there is a single hyperplane that splits each object into two pieces of equal area. The special case of $n = 2$ is called the pancake theorem.

5.3 Red Blue Matching

Given n red and n blue points in the plane, we can pair them up using non-intersecting line segments!

Proof (by recursive algorithm).

1. If $|P| = 2$, match up the two points with a line segment

[2]By general position assumption - can rotate the plane slightly to prevent this if it's true.

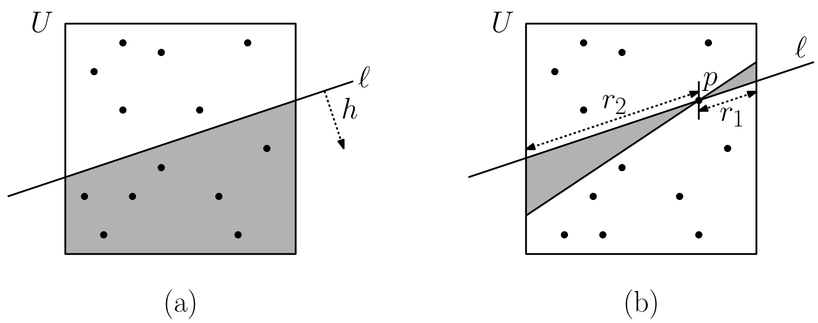

Figure 11. Computing Discrepancy. ([1])

2. If $|P| > 2$
   (a) Compute a ham-sandwich cutting line
   (b) If $|P|/2$ is odd, match the red and blue point on the cutting line. (If $|P|/2$ is even, there are no points on the cutting line.)
   (c) Recursively compute a matching for both P_1 and P_2, where P_1 and P_2 are the points on either side of the cut.

5.4 Halfplane Discrepancy

Discrepancy is a problem closely related to sampling. It measures how well a discrete sample can approximate a continuous structure. Initially, it was important in computer graphics for determining how well a collection of random rays would be spread over a continuous surface. But it has new applications in big data (finding how well a small sample represents a larger population), and also probability and randomized algorithms (e.g. discretizing continuous probability distributions).

We are given a set, P of n points in the unit square U, and we want for any halfplane h that the fraction of points lying in the halfplane is close to the area of $h \cap U$. If we define $\mu(h)$ to be the area of $h \cap U$ and $\mu_P(h) = |P \cap h|/|P|$, then we can define the *discrepancy* for a point set P and a halfplane h as:

$$\Delta_p(h) = |\mu_h - \mu_P(h)|.$$

We define the halfplane discrepancy of P as supremum (least upperbound) of this quantity over all h:

$$\Delta(P) = \sup_h |\mu_h - \mu_P(h)|.$$

The Discrepancy Problem

Given: A set of n points P lying in the unit square.

Goal: Compute $\Delta(P)$ and return a halfplane that achieves this discrepancy.

5.4.1 The Solution

The following lemma helps us compute $\Delta(P)$.

Lemma: Let h be the halfplane that maximizes discrepancy with respect to P. Let ℓ denote the line that bounds h. Then either (1) ℓ passes through one point of P and this point is the midpoint of the line segment $\ell \cap U$; or (2) ℓ passes through two points of P.

Proof: Consider a halfplane induced by a line ℓ that does not intersect any points. Then the discrepancy can be increased by either moving ℓ up or down (whichever direction increases discrepancy) until it hits a point. (See Figure 11 (a)). Thus ℓ must intersect at least one point.

Now consider the case where ℓ intersects just one point, but that this point does *not* form the midpoint of the line segment $\ell \cap U$. In this case, there are two line segments r_1 and r_2 of different length that are induced by the point (See Figure 11 (b)).

Assume $r_1 < r_2$, and consider the rotation of h about p by a small angle ϕ (ϕ given in radians). Then the asymptotic approximation to the gain due to the area on the right is $r_1^2\phi/2$, since the triangle giving the area gain can be approximated by an sector of angle ϕ in a circle of radius r_1.[3] The loss due to the area on the left is similarly asymptotically $r_2^2\phi/2$.

Thus this rotation will decrease the area of the region lying below h. Similarly a rotation in the opposite direction will increase the area of the region lying below h. Hence, it is again possible to increase the discrepancy via rotation in the appropriate direction. $\square$

Given this lemma, we can solve the problem in $O(n^2)$ time as follows. Let a *type 1* line be one that passes through one point of P and this point is the midpoint of the line segment $\ell \cap U$. Let a *type 2* line be one that passes through two points of P. Note that there are only $O(n)$ type 1 lines, so we can easily compute the discrepancy of the halfplanes induced by these lines in $O(n^2)$ time.

For the type 2 lines, we use arrangements and levels. In particular, we know that any type 2 line will be a vertex in the arrangement in the dual of P^*, i.e. in $\mathcal{A}(\mathcal{P}^*)$. For each such vertex $p \in \mathcal{A}(\mathcal{P}^*)$, we can compute the area of the two halfplanes induced by the line p^* in the bounding box U in constant time. Thus we can compute μ_h in constant time per type-2 halfplane h.

Moreover, we can compute the levels of all the points in $\mathcal{A}(\mathcal{P}^*)$ in $O(n^2)$ time as described in the Section 5.1.1. This allows us to compute in constant time the value $\mu_P(h)$ in constant time per type-2 halfplane h.

Since the total number of type-2 halfplanes is $O(n^2)$ and we can compute the discrepancy induced by each in constant time, we can also find the maximum discrepancy of these type-2 halfplanes in $O(n^2)$.

6 Duality and Arrangements in Higher Dimensions

In $\mathbb{R}^d$, identify the y axis with the d-th coordinate. Then a point can be written as $p = (x_1, \ldots, x_{d-1}, y)$ and a $d-1$ dimensional hyperplane, h is $y = \sum_{i=1}^{d-1} a_i x_i - b$. The dual of the hyperplane is the point $h^* = (a_1, \ldots, a_{d-1}, b)$ and the dual of the point is the hyperplane $b = \sum_{i=1}^{d-1} x_i a_i - y$. All the properties general to point/hyperplane relationships under the assumption that the $y(b)$ axis is "vertical".

An arrangement of d dimensional hyperplanes has combinatorial complexity $O(n^d)$.

[3]Technically, we can say that as *phi* goes to 0, the area of the triangle is $r_1^2\phi/2 + o(r_1)$, since the error in approximating the shape as a sector angle versus a triangle is $o(r_1)$

References

[1] David Mount. Computational Geometry. http://www.cs.umd.edu/class/fall2016/cmsc754/Lects/cmsc754-fall16-lects.pdf, 2016.

[2] Alper Ungor. Lecture 11: Arrangements and Duality. https://www.cise.ufl.edu/class/cot5520sp15/CG_ArrangementsCh8.pdf, 2016. https://www.cise.ufl.edu/class/cot5520sp15/CG_ArrangementsCh8.pdf.

[3] Emo Welzl and Prof. Bernd Gartner. Chapter 11: Line Arrangements. https://www.ti.inf.ethz.ch/ew/Lehre/CG13/lecture/Chapter https://www.ti.inf.ethz.ch/ew/Lehre/CG13/lecture/Chapter%2011.pdf.