Projective Duality

Intuition: \[ \mathbb{C}^2 \]
Actual Computation: \[
\begin{vmatrix}
1 & x_i & y_i \\
1 & x_j & y_j \\
1 & x_k & y_k
\end{vmatrix} \geq 0?
\]

But pairs of numbers can also be interpreted differently:

Interval \[ \Rightarrow \] Point \( (a, b) \) \[ \Rightarrow \] Line \( y = ax - b \)

... but then what are the analogues of "left", "right", "ccw", etc?

Halfspace Intersection

Input: A set \( L \) of lines in the plane \( y = ax - b \)
Output: A description of their upper envelope

Intersection of upper halfplanes \( y \geq ax - bi \)

Sequence of supporting lines (indices) in order from left to right

We can infer vertex coordinates by solving linear system:
\[
\begin{align*}
y &= ax_i - bi \\
y &= ax_j - bj
\end{align*}
\]

Divide and Conquer ("Merge-envelope")

Partition input \( L \) into equal-sized subsets \( B \) and \( R \)
Recursively compute upper envelopes of \( B \) and \( R \)
Merge upper envelopes by sweeping
Merge using a sweep algorithm.

Sweep a vertical line \( l \) from left to right over \( \mathbb{R} \) and \( \hat{\mathbb{R}} \).

- Record intersections with \( l \):
  - which edge of \( \mathbb{R} \)
  - which edge of \( \hat{\mathbb{R}} \)
  - which intersection is higher.

Initially, \( r = 1 \), \( b = 1 \) whichever has smaller slope*  

* assume noties

Info changes at next:
- vertex of \( \mathbb{R} \)
- vertex of \( \hat{\mathbb{R}} \)
- intersection point of \( \mathbb{R} \cap \hat{\mathbb{R}} \)
whichever is furthest left.

We can compute these x-coordinates in \( O(1) \) time.

Repeat:
- Move \( l \) to next event
- if higher intersection changes
- record higher intersection
- until \( x_e = +\infty \)

Several cases to consider. For example:

Merge runs in \( O(n) \) time \( \Rightarrow \) Entire algorithm in \( O(n \log n) \) time.
Output-Sensitive algo:

Start with line \( l \) with min. slope
repeat
  Find next vertex \( \leftarrow O(n) \) time
  Update \( l \)
until \( l \) = line with max. slope

\[ h \text{ iterations} \]
\[ \Rightarrow O(nh) \text{ time} \]

Huh.

Have we seen this before?

This is Jarvis' March!

Duality

Define the dual of point \((a,b)\) as the line \( y = ax - b \)
\[ P \]

Dual of line \( y = a'x - b' \) as the point \((a', b')\)
\[ l \]

Incidence:

<table>
<thead>
<tr>
<th>Primal</th>
<th>Math</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P \in l )</td>
<td>( a_1 a' = b_1 + b' )</td>
<td>( P^* \in L^* )</td>
</tr>
<tr>
<td>( l = \overline{P_1P_2} )</td>
<td>( a_1 a' = b_1 + b' ); ( a_2 a' = b_2 + b' )</td>
<td>( L^* = P_1^* \cap P_2^* )</td>
</tr>
</tbody>
</table>

Order:

| \( P \) is above \( l \) | \( a_1' < b_1 + b' \) | \( L^* \) is above \( P^* \) |
| \( P \) is below \( l \) | \( a_1' > b_1 + b' \) | \( L^* \) is below \( P^* \) |
| \( P_1 \) left of \( P_2 \) | \( \hat{a}_1 < \hat{a}_2 \) | \( L_1 \) smaller slope than \( L_2 \) |
| \( P_1 \) above \( P_2 \) | \( b_1 > b_2 \) | \( L_1 \) intersects \( y = ax \) below \( L_2 \) |

Vertical Distance

\[ P_1 \]

\[ d = b_1 - b_2 \]

\[ P_2 \]
Orientation \( P_1 P_2 P_3 \text{ ccw} \quad \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} > 0 \quad P_1^* P_2^* P_3^* \text{ ccw} \\

Assume \( p_2^* \) furthest left \( \text{slope}(P_2 P_3) < \text{slope}(P_1 P_3) \)

Assume \( p_3^* \) smallest slope \( x(P_1 P_2) < x(P_1 P_3) \)

Convex stuff

Lower convex hull = all lines below \( P \)

Upper envelope = all points above \( P^* \)

Lower hull vertices from left to right

Lower hull edges from left to right = increasing slope

Upper envelope edges by increasing slope = left to right

Upper envelope vertices from left to right

It's the same algorithm!

Start with line \( l \) with min slope
Repeat
Find next vertex
Update \( l \)
Until \( l = \) line with max. slope

Start with leftmost point \( p \)
Repeat
Find next edge
Update \( p \)
Until \( p = \) rightmost point

Duality is not a transformation.
It's a type cast.
Dual of merge-envelope:

Split $P$ into equal-sized subsets $R$ and $B$
Recursively compute lower hulls $\text{conv}(R)$ and $\text{conv}(B)$

Merge by rotating line

Increase slope from $\infty$ to $\infty$
Maintain vertices whose tangent lines have current slope and which tangent line is lower

Change at
next red edge
next blue edge
or red-blue line whichever is first