Professor Diane Souvaine
Scribe: Ziqi (Stephanie) Xu

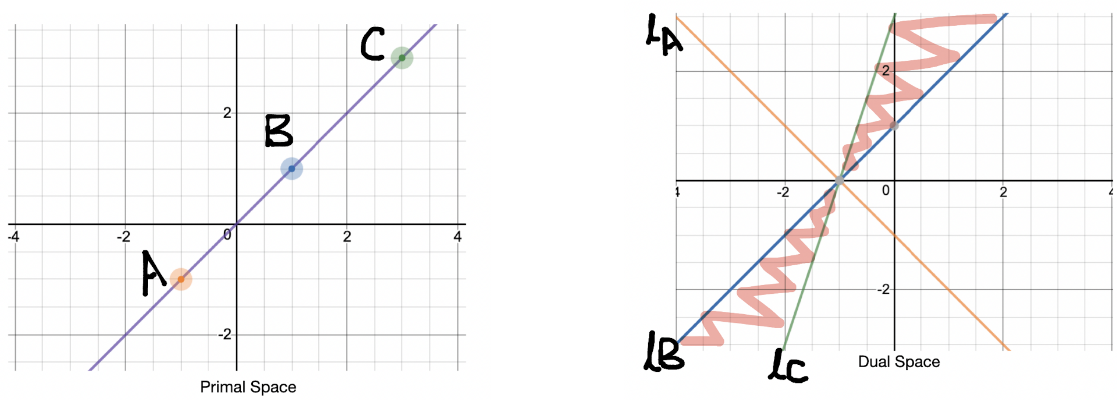
Double Wedges and LMS Regression

Oct 17, 2022

1. Double Wedge

According to the point-line duality that is introduced in last lecture, we can map a point in the primal space to a line in the dual, or a line in the primal to a point in the dual.

Moreover, we can also map a line segment in the primal space to a double wedge in the dual space. For example, as shown in Figure 1, points A, B, and C can be mapped to lines l_A, l_B, and l_C in the dual, and the line segment $\overline{BC}$ in the primal can be mapped to a range of slopes between line l_B and line l_C that are colored red in the dual space.



(a) Points in Primal Space

(b) Lines and Lines Segments in Dual Space

Figure 1

2. Relationships between two line segments

There are three positional relationships between two line segments, and each of them has a corresponding double wedge positional relationship in the dual.

Case 1 When two line segments do not intersect but their extended lines intersect in the primal, the center points of their double wedges in the dual would not be in each other's double wedge area.

Case 2 When two line segments do not intersect but l_2 intersect with l_1's extended line in the primal, the center point of l_1's double wedge would be in l_2's double wedge area in the dual.

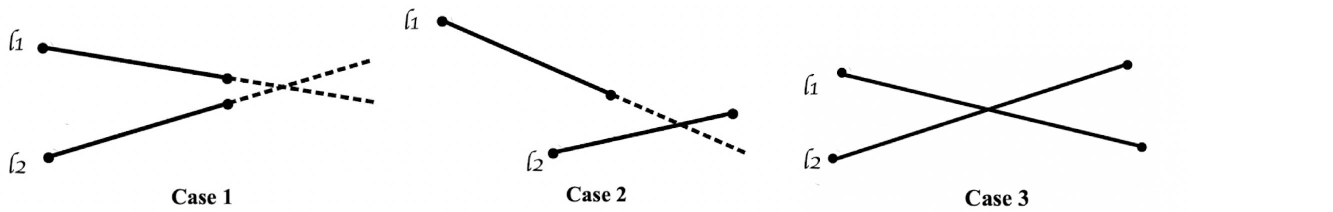


Figure 2: In the Primal

Case 3 When two line segments intersect in the primal, the center points of their double wedges would be in each other's double wedge area in the dual.

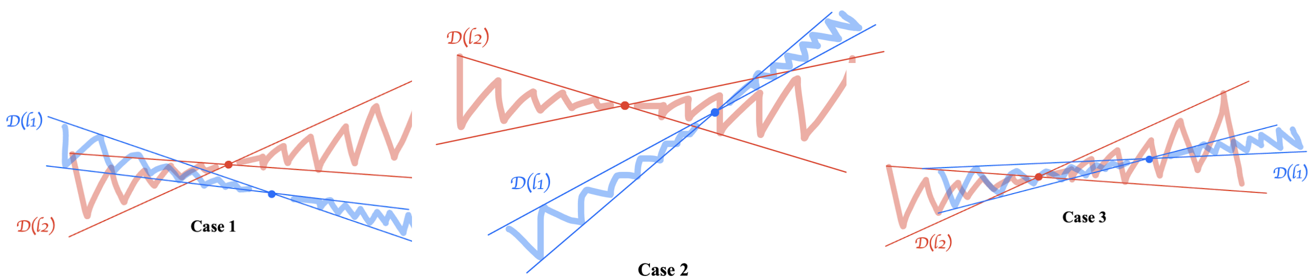


Figure 3: In the Dual

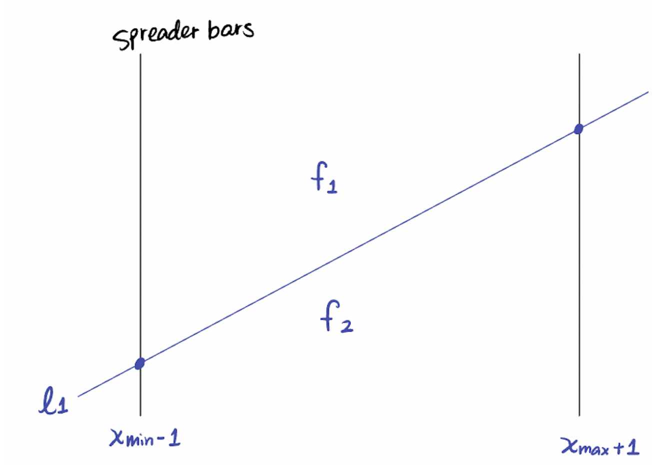
3. Walking through double wedges in the dual

If there are n line segments in the primal, there will be $2n$ lines in the dual and $O(n^2)$ intersection points among which there are n center points of double wedges.

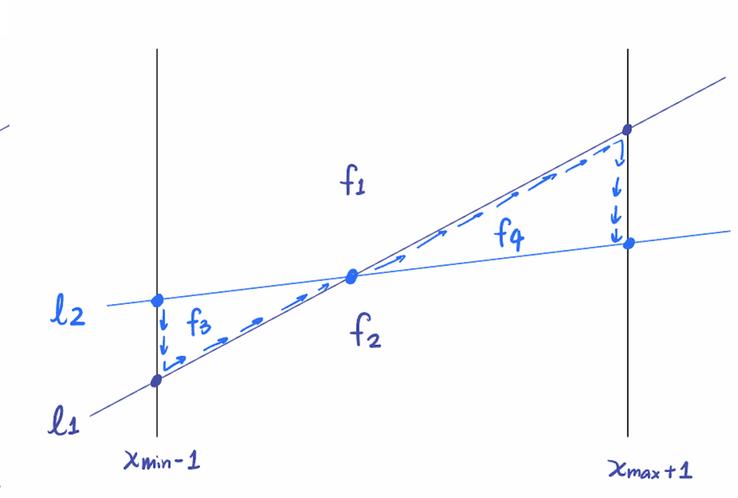
We know that if we use a sweepline algorithm, then the runtime is going to be $O(n^2 \log n)$ to walk through each double wedge in the dual, and the space is $O(n)$. But we can achieve better runtime by building a DCEl, turning the set of lines into a graph which Chazelle called a "hammock".

3.1 Building a DCEL

1. Presort the lines by slopes. Put a pair of spread bars at $x_{min} - 1$ and $x_{max} + 1$. Add the line with the highest slope into the empty DCEL.
2. Keep adding lines by their slope order. At each iteration, divide one face into two in constant time for each face the line passes through, by walking around the faces that the line crosses.



(a) Initial DCEL



(b) Adding the second line into the DCEL

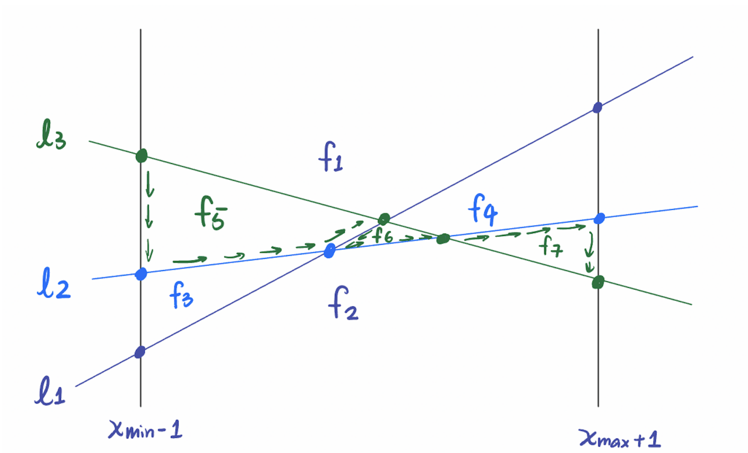


Figure 5: Adding the third line

The cost of putting a new line into the DCEL depends on the number of faces the line crosses and the number of edges in each face. We can use **amortization** analysis to prove that it takes $O(n)$ time to process each line so that overall it takes $O(n^2)$ to build the DCEL.

- For each face that's crossed by the current line, there are 2 edges that are directly connected to the intersection points. And we know that the number of intersectino points across one line is $O(n-1)$.
- For each edge not directly connected to an intersection point, follow the direction that's opposite to the "max" point of the face, turn the corner and deposit a charge. (See the orange and green edges in the figure.
- Each intersection point will be charged at most once from the same side of the face. This is because if one point gets 2 charges from the same side, then one of the faces

does not touch the line. Therefore, the number of edges that we are gonna walk through after adding one line will be at most $O(n-1)$, the same complexity as the number of intersection points across that line.

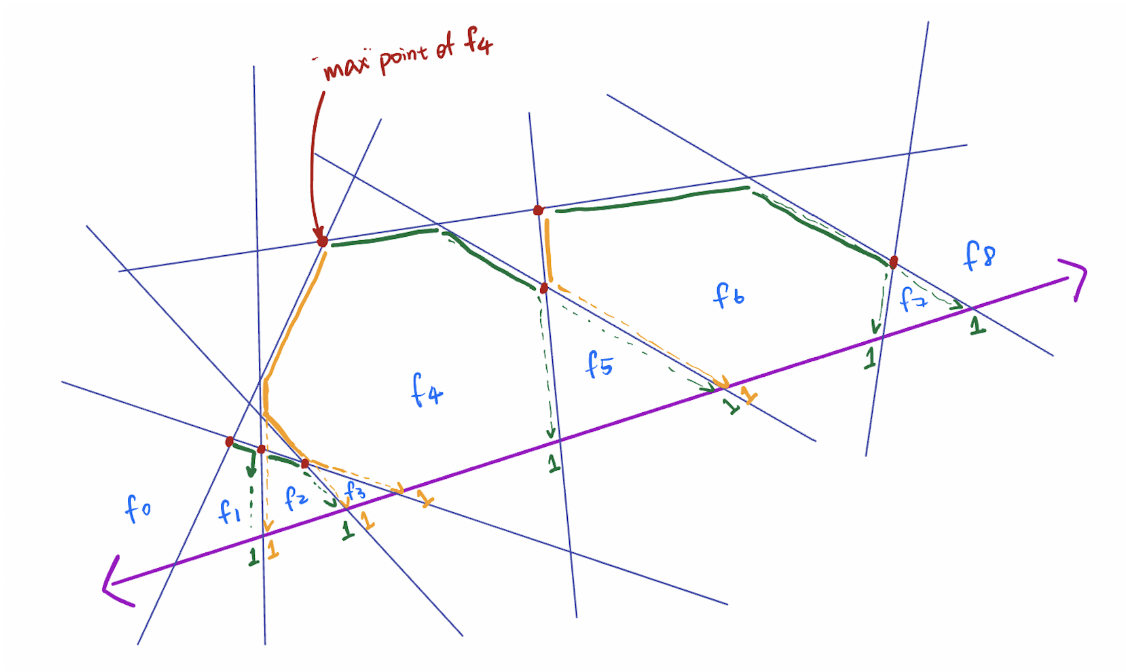


Figure 6: Amortization analysis

Therefore, building a DCEL of double wedges has a runtime of $O(n^2)$, which is less than the sweepline algorithm, and a space complexity of $O(n^2)$ which is the number of total intersection points.

4. LMS Regression

We can compute the Least Median of Squares of a set of points by computing the minimum distance between 2 lines that has $\lceil \frac{n}{2} \rceil$ lines in between in the dual space using DCEL.

This can be done in $\Theta(n^2)$ time by doing $\lceil \frac{n}{2} \rceil$ walks from one side of the bar to the other. The first walk can be initialized from line l_1 and $l_{\lceil n/2 \rceil}$, and the second walk starts form line l_2 and $l_{\lceil n/2 \rceil + 1}$. During one walk, always keeps $\lceil \frac{n}{2} \rceil$ lines in between, and calculate the distance between at every intersection point.