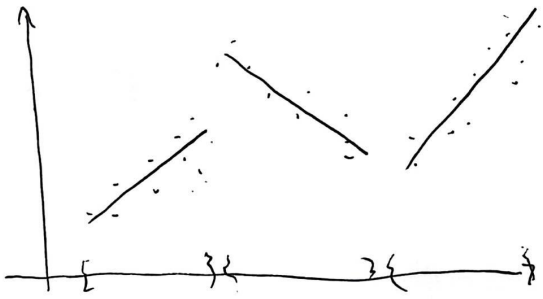
Segmented Line Fitting



input P = set of points $\{(x_i, y_i)\}_{i=1}^{n}$

s.t. $x_1 < x_2 < \ldots < x_n$

given some line L,

$y = ax + b$

$$\text{error} = \sum_{i=1}^{n} (y_i - ax_i - b)^2$$

↑ std. sum of squares error

Another parameter c = cost of introducing a new line.

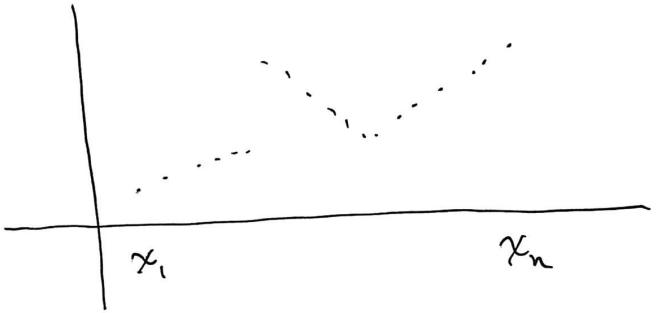
Using calculus,

$$a = \frac{n \sum x_i y_i - \left(\sum x_i\right)\left(\sum y_i\right)}{n \sum x_i^2 - \left(\sum x_i\right)^2}$$

↖ Minimize the Error ↙

$$b = \frac{\sum y_i - a \sum x_i}{n}$$

idea: use dynamic programming...



$OPT(j)$ = optimal fit for points $(x_1, y_1), \ldots, (x_j, y_j)$

Defn e_{ij} = best fit for points $(x_i, y_i) \ldots (x_j, y_j)$

$$= \sum_{k=i}^{j} (y_k - a_{ij} x_k - b_{ij})^2$$

$$a_{ij} = \frac{n \sum_{k=i}^{j} x_k y_k - \left(\sum_{i}^{j} x_k\right)\left(\sum_{i}^{j} y_k\right)}{n \sum_{i}^{j} x_k^2 - \left(\sum_{i}^{j} x_k\right)^2}$$

$$b_{ij} = \frac{\sum_{i}^{j} y_k - a \sum_{i}^{j} x_k}{n}$$

e_{ij} can be computed in $O(n^3)$ time.

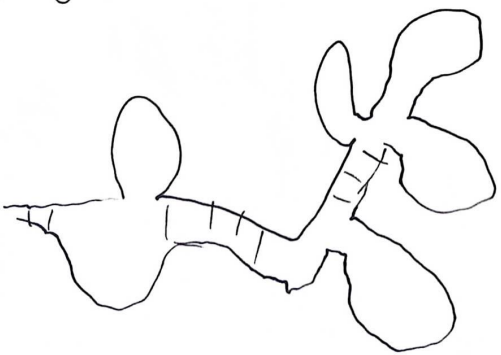
$$OPT(j) = \min_{1 \le i \le j} \left(e_{i,j} + c + OPT(i-1) \right)$$

j	OPT	i* // argmin
0	0	
1		
2		
⋮		
n		

$O(n^2)$ →

RNA Secondary Structure

String A C G U



"secondary structure":
defined by which bases form pairs.