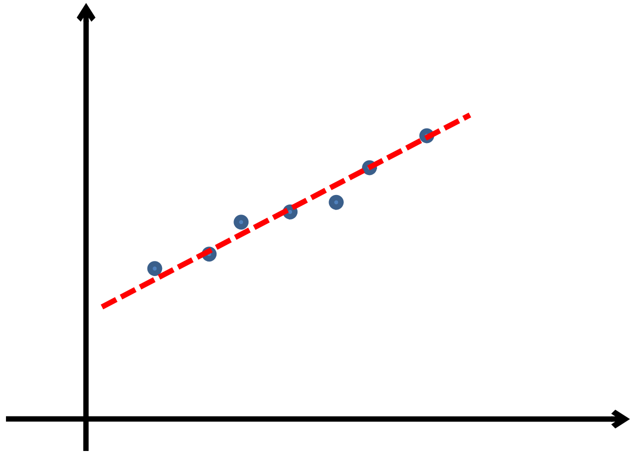
An Awesome Problem

Best Fit Line

- The problem is to find the best fit line
- Often appears in statistical data / scientific data.



A formal statement

- Given a set P of n points in the plane, denoted (x_1,y_1), (x_2,y_2), (x_3,y_3),…, (x_n,y_n); and suppose $x_1 < x_2 < \ldots < x_n$.
- Given a line L defined by the equation $y = ax + b$, we say that the error wrt P is sum of the its squared distances to the points in P:

Natural Goal for best fit line

- Find the line with minimum error

- Has a closed form solution

$$a = \frac{n\sum_i x_i y_i - \left(\sum_i x_i\right)\left(\sum_i y_i\right)}{n\sum_i x_i^2 - \left(\sum_i x_i\right)^2} \text{ and } b = \frac{\sum_i y_i - a\sum_i x_i}{n}$$

Can it handle these?

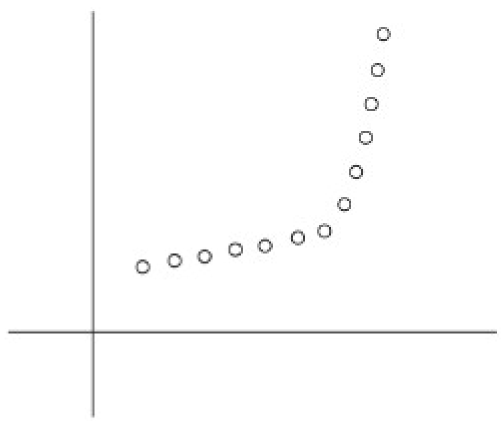


Figure 6.7 A set of points that lie approximately on two lines.

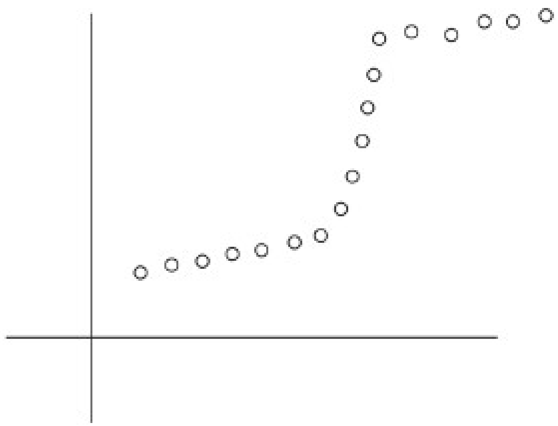
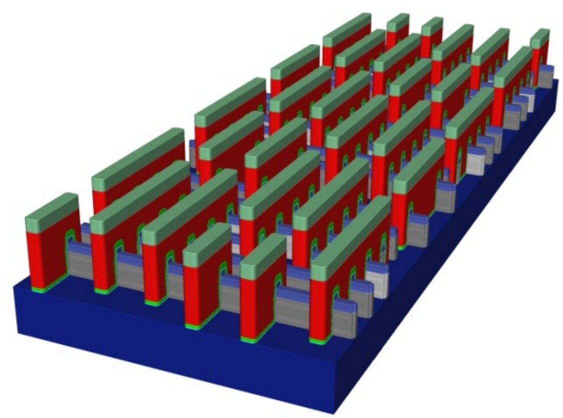


Figure 6.8 A set of points that lie approximately on three lines.

An interesting application

Fig.1 Contoured aerial photograph

A Novel Method for Extracting Building from LIDAR Data----------Fc-S method by Zizhen et al. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B1. Beijing 2008*



http://www.electronics-eetimes.com/en/3d-tool-aims-at-28nm-for-mems-and-new-structures.html?cmp_id=7&news_id=222912048#

Can it handle these?

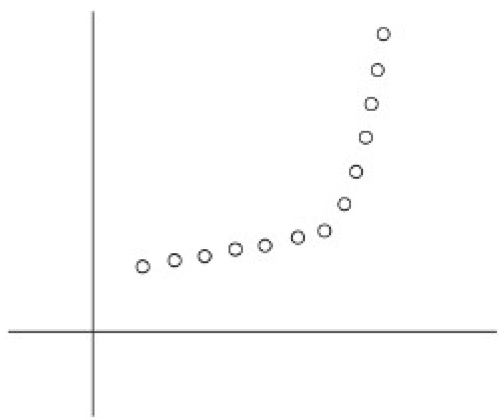


Figure 6.7 A set of points that lie approximately on two lines.

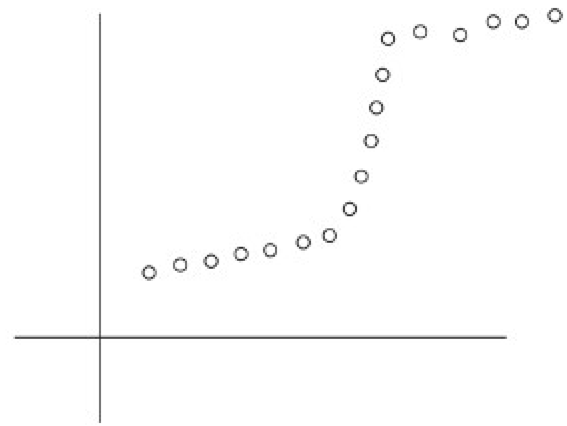


Figure 6.8 A set of points that lie approximately on three lines.

- We want to fit multiple lines through P.
- But how many?
- Also, Lines must minimize the error.

The Problem statement

- SEGMENTED LEAST SQUARES
- Given a set P of n points in the plane, denoted (x_1,y_1), (x_2,y_2), (x_3,y_3),…, (x_n,y_n); and suppose $x_1 < x_2 < \ldots < x_n$.
- Partition P into some *segments*.
- Segments is a subset of P that represents a contiguous set of x-coordinates.
- Compute the line minimizing the error with respect to the points in S.

A Penalty Function

- We want to penalize too many partitions as well as the error.
- So Penalty is sum of
  - The number of segments into which we partition P, times a fixed, given multiplier $C > 0$.
  - For each segment, the error value of the optimal line through that segment.
- **Objective: Find a partition with minimum penalty.**

Exploiting the subproblems

- We want a polynomial number of sub problems.
- Great Observation.
  - The last point p_n belongs to a single segment in the optimal partition, and that segment begins at some earlier point p_i.

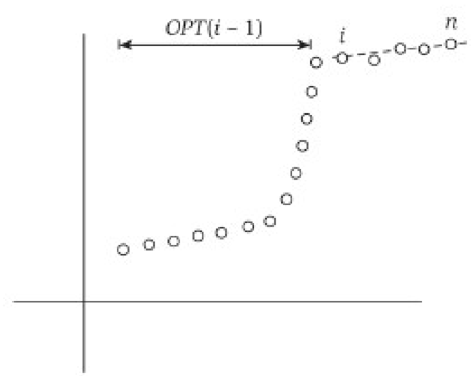


Figure 6.9 A possible solution: a single line segment fits points $p_i, p_{i+1}, \ldots, p_n$, and then an optimal solution is found for the remaining points $p_1, p_2, \ldots, p_{i-1}$.

Recurrence relation

- Let OPT(i) be the optimal value for the points $p_{1,}\ p_{2,}\ \ldots,\ p_i$.
- Let $e_{i,j}$ denote the minimum error of any line through $p_{i,}\ p_{i+1,}\ \ldots,\ p_j$.
- We want to compute OPT(n).
- So the Observation is

If the last segment of the optimal partition is $p_{i,}\ p_{i+1,}\ \ldots,\ p_n$, then the value of the optimal solution is $OPT(n) = e_{i,n} + C + OPT(i-1)$.

Completing the recurrence relation

Since i can take only j distinct values,

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

The segment p_i, p_{i+1}, …, p_j is used in an optimum solution for the subproblem if and only if the minimum is obtained using index i.

So the algorithm

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

`Segmented-Least-Squares(n)`

`Array` $M[0 \ldots n]$

`Set` $M[0] = 0$

`For all pairs` $i \le j$

O(n)

O(n^3)

`Compute the least squares error` $e_{i,j}$ `for the segment` $p_i, \ldots, p_j$

`Endfor`

`For` $j = 1, 2, \ldots, n$

`Use the recurrence (6.7) to compute` $M[j]$

`Endfor`

O(n) for each j

`Return` $M[n]$

Final solution

```
Find-Segment(j)
  If j== 0 then
     Output nothing
  Else
     Find an i that minimizes e_{i,j} + C + M[i-1]
     Output the segment {p_i, …,p_j} and the
      result of Find-Segment(i-1)
   Endif
```

Reference

- Algorithm Design by Kleinberg and Tardos
- courses.cs.vt.edu/~cs5114/.../lecture12-dynamic-programming.pdf