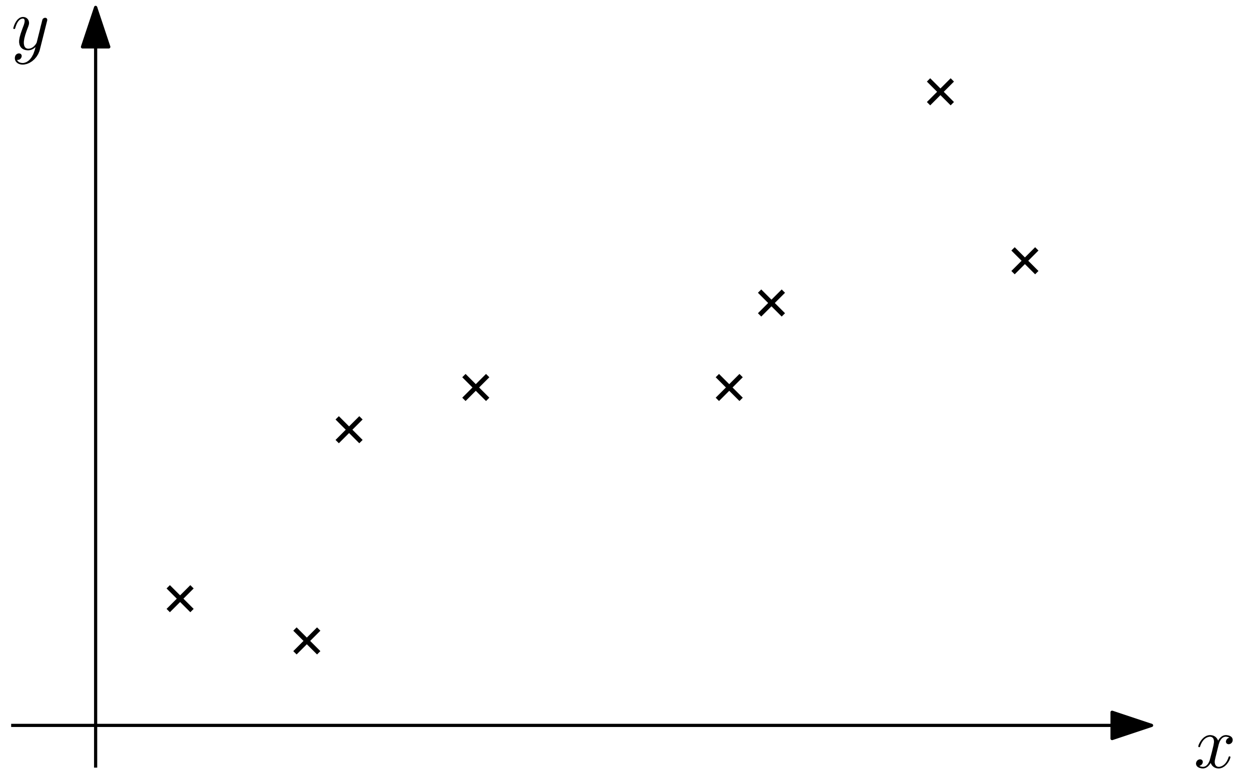
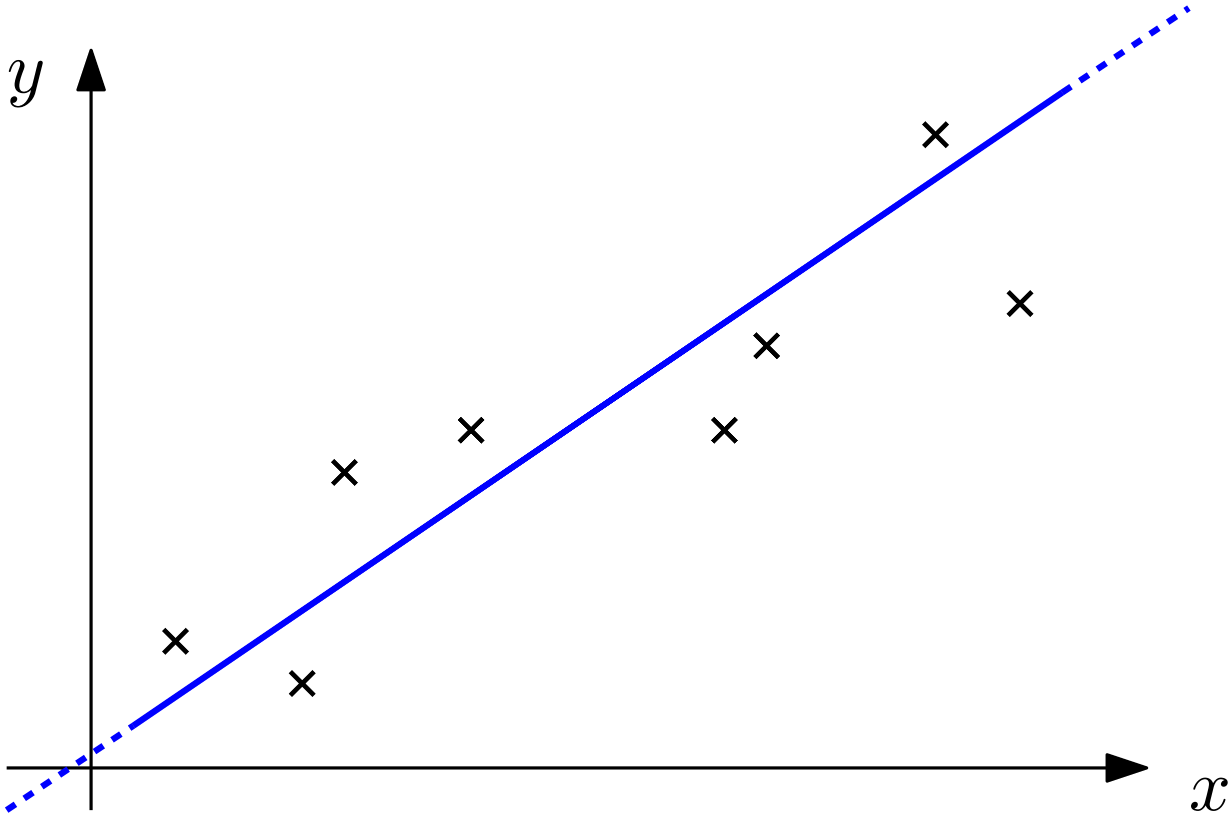
Segmented Least Squares

Linear Regression



Given a collection P of points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ ...

Linear Regression

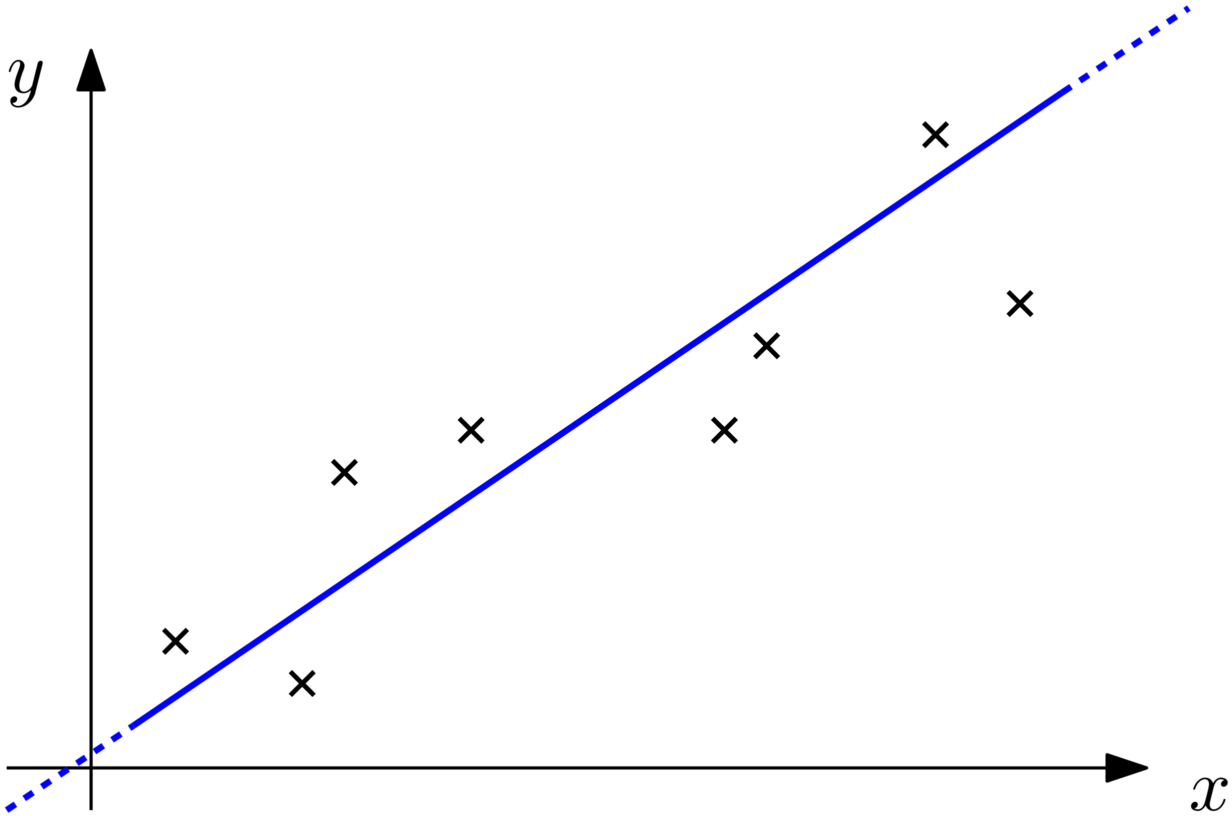


Given a collection P of points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ ...

... find the "line ℓ of best fit" through the points

$$\ell(x) = a \cdot x + b$$

Linear Regression



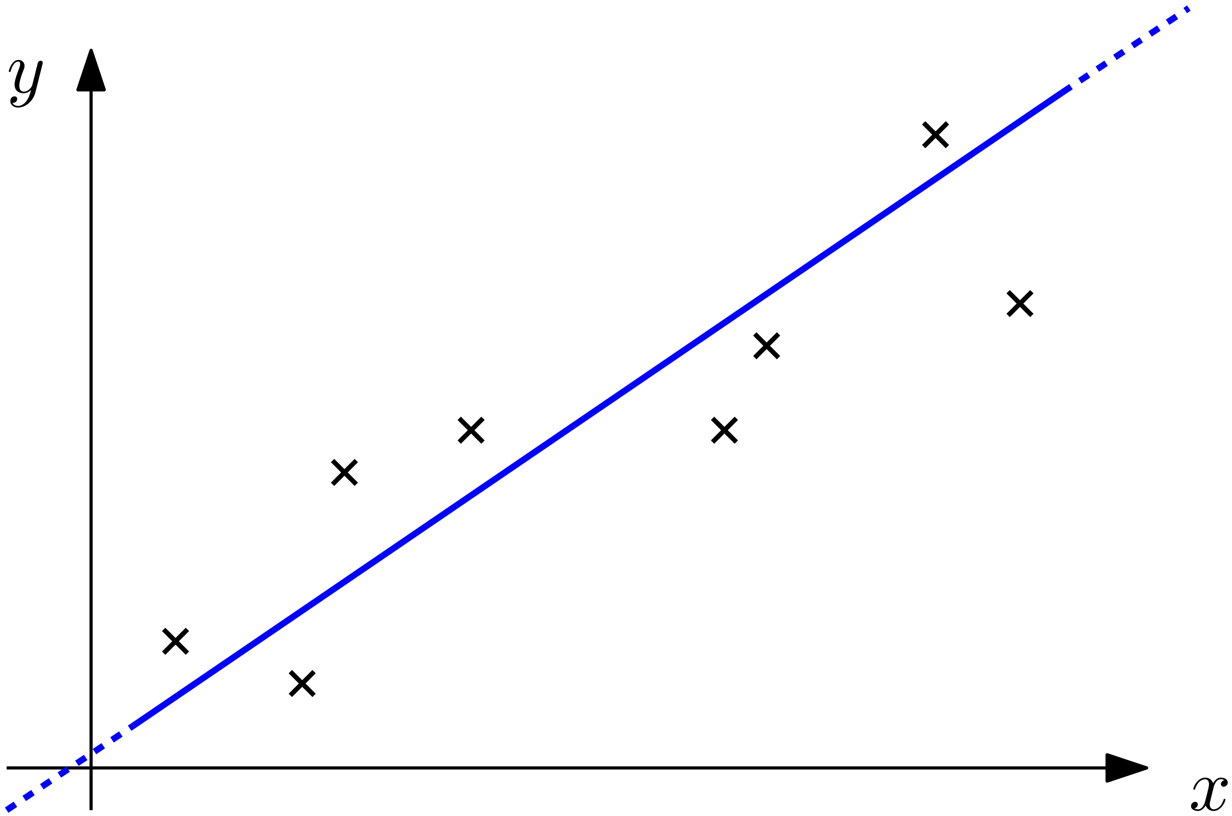
Given a collection P of points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ . . .

. . . find the "line ℓ of best fit" through the points

$$\ell(x) = \boxed{a} \cdot x + \boxed{b}$$

Need to find a and b

Linear Regression



Given a collection P of points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ ...

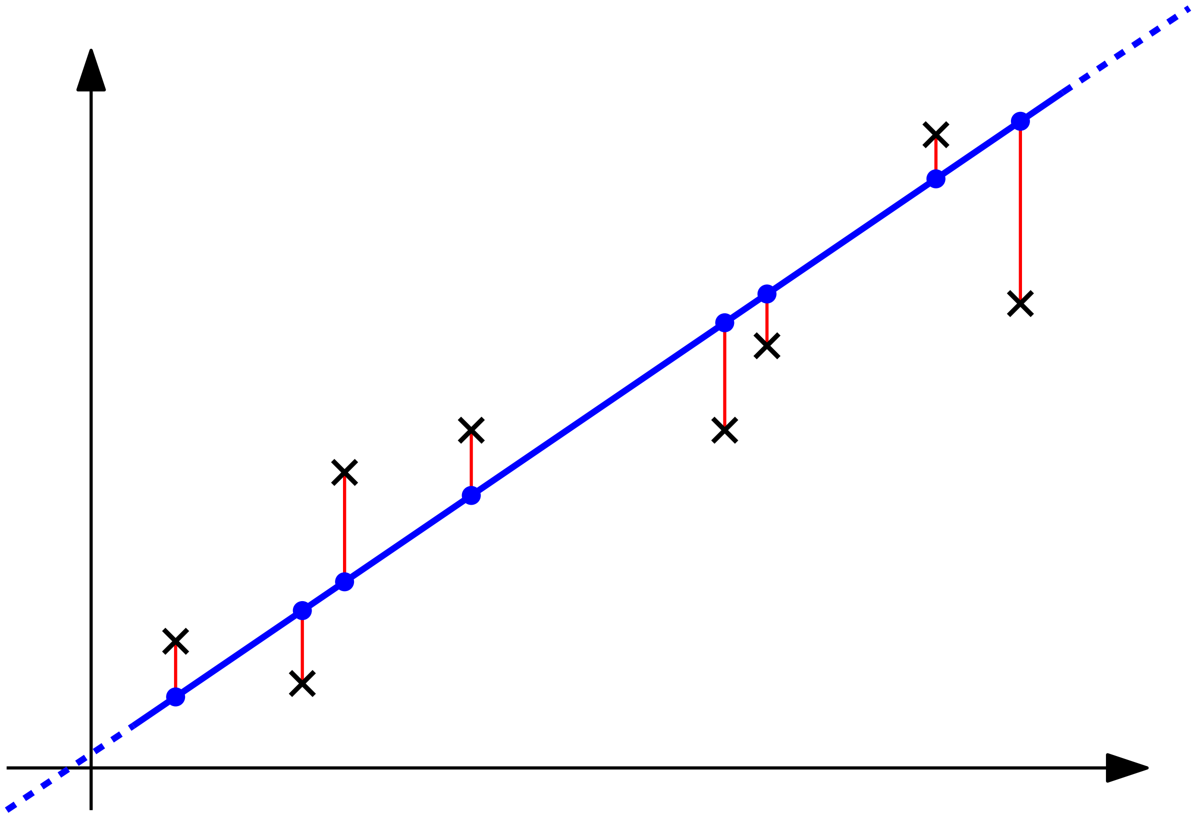
... find the "line ℓ of best fit" through the points

$$\ell(x) = \boxed{a} \cdot x + \boxed{b}$$

Need to find a and b

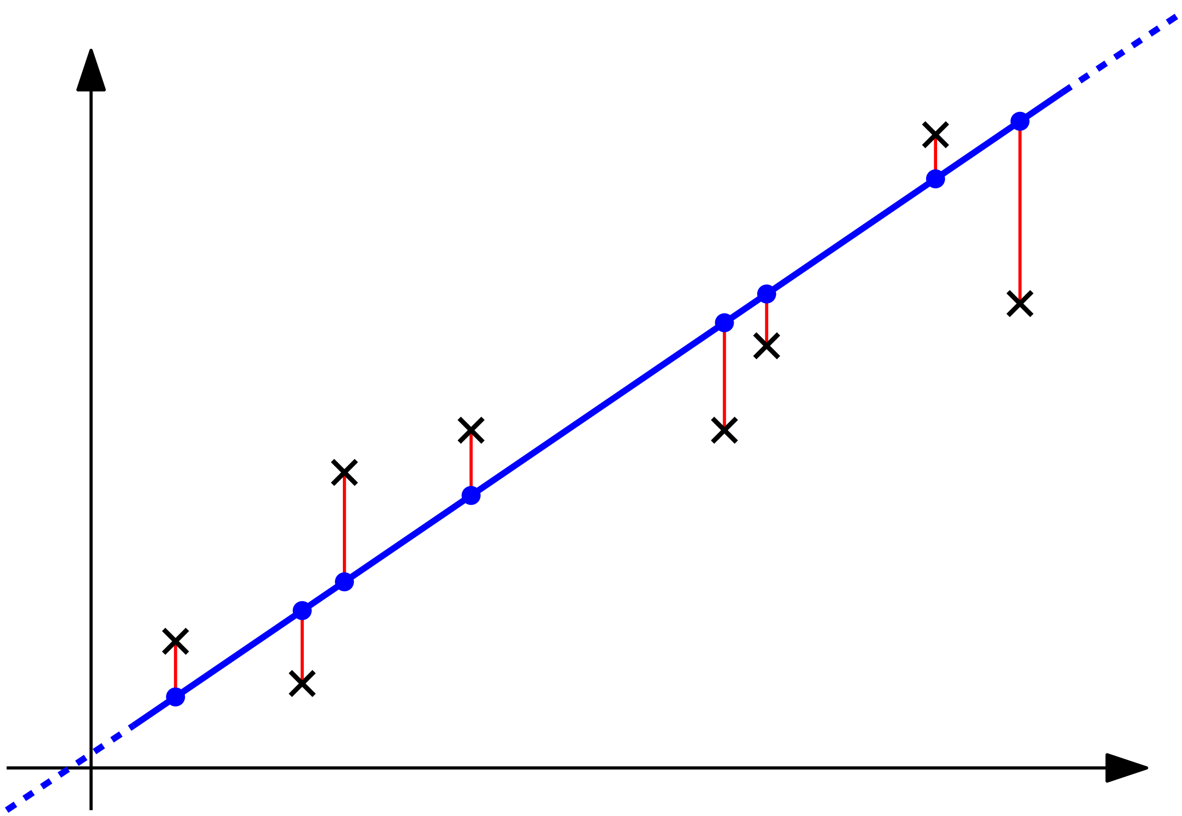
What does "best" mean? Minimizes some error measure

Least Squares Regression



Residual: difference between the y-coordinate of a point and the corresponding value of the fitted line

Least Squares Regression

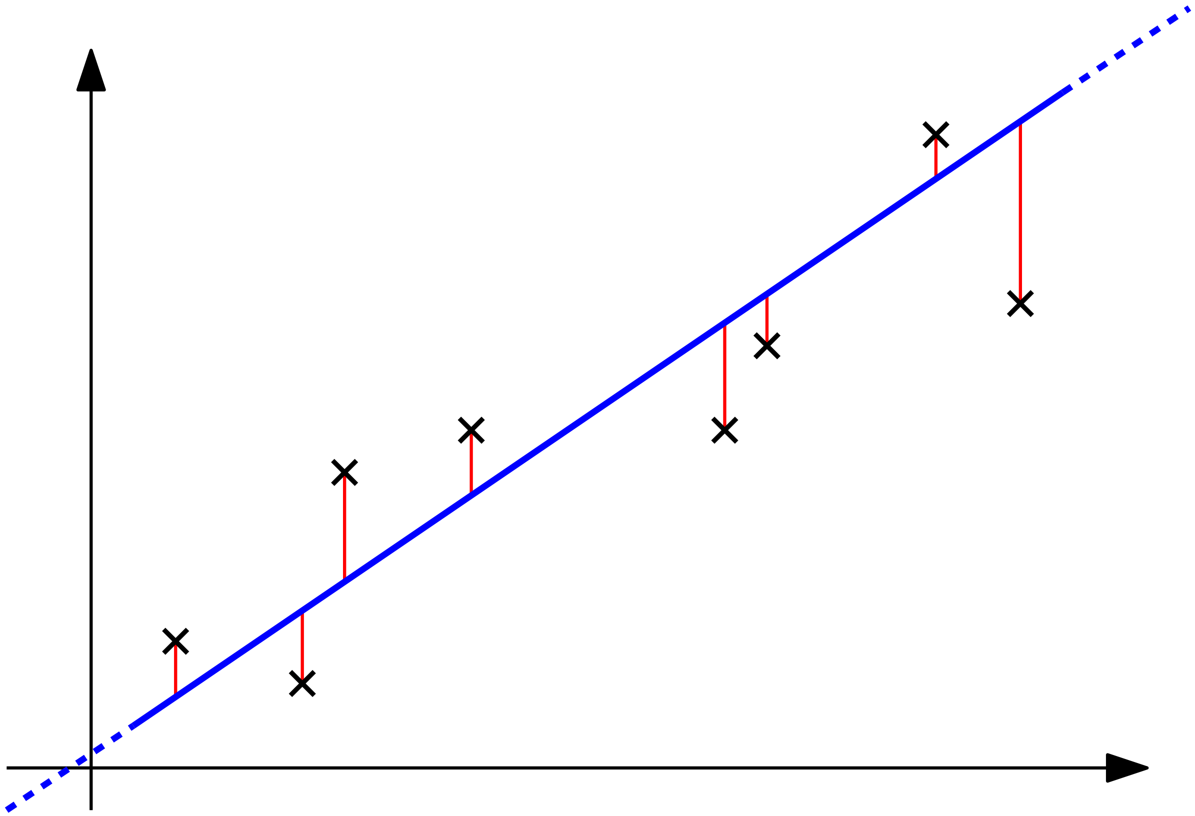


Residual: difference between the y-coordinate of a point and the corresponding value of the fitted line

Error: sum of the **squares** of the residuals

$$\mathrm{Err}(\ell, P) = \sum_{i=1}^{n} \Big(\ell(x_i) - y_i \Big)^2$$

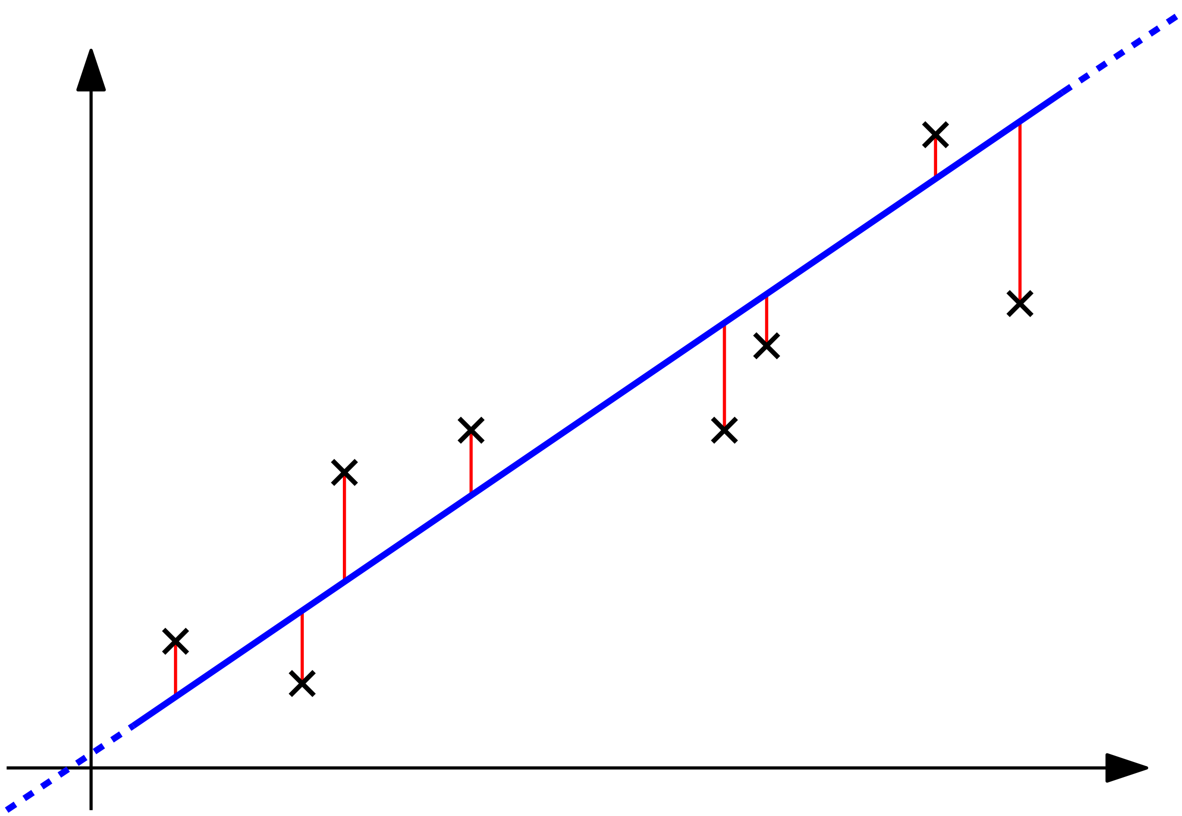
Least Squares Regression



The parameters of the line $\ell(x) = a \cdot x + b$ of best fit are:

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

Least Squares Regression



The parameters of the line $\ell(x) = a \cdot x + b$ of best fit are:

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

Can be found in time $O(n)$

Segmented Least Squares Regression

What if the points look like this?

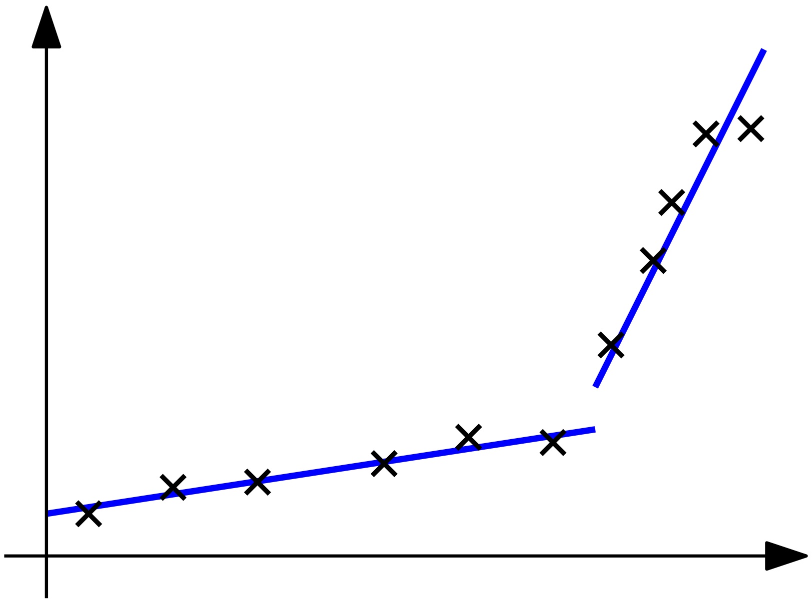
Segmented Least Squares Regression

What if the points look like this?

- No single line provides a good fit

Segmented Least Squares Regression

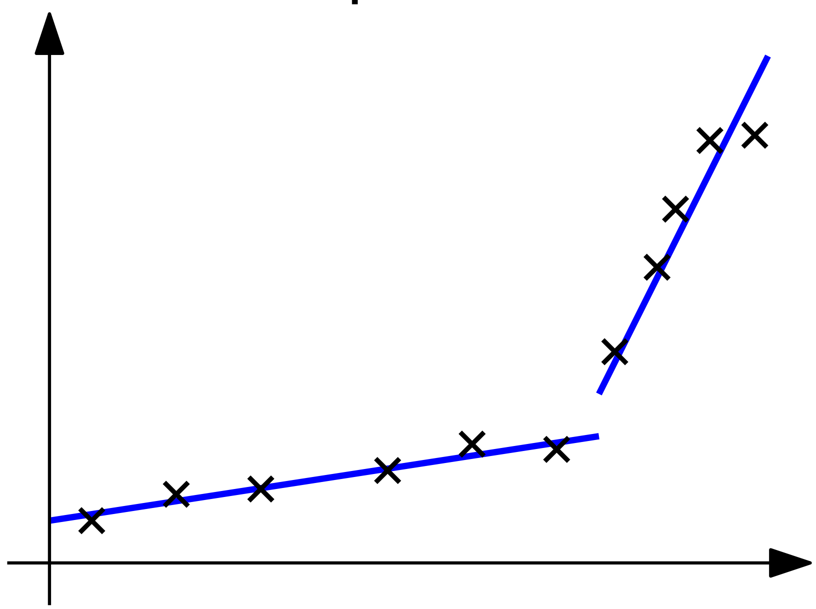
What if the points look like this?

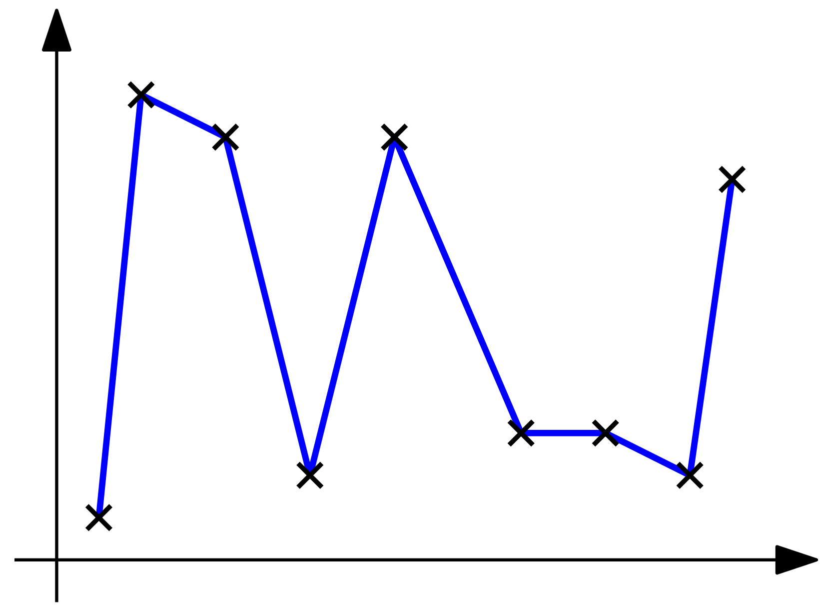


- No single line provides a good fit
- We can get a good fit with using **piecewise linear functions** instead of lines

Segmented Least Squares Regression

What if the points look like this?



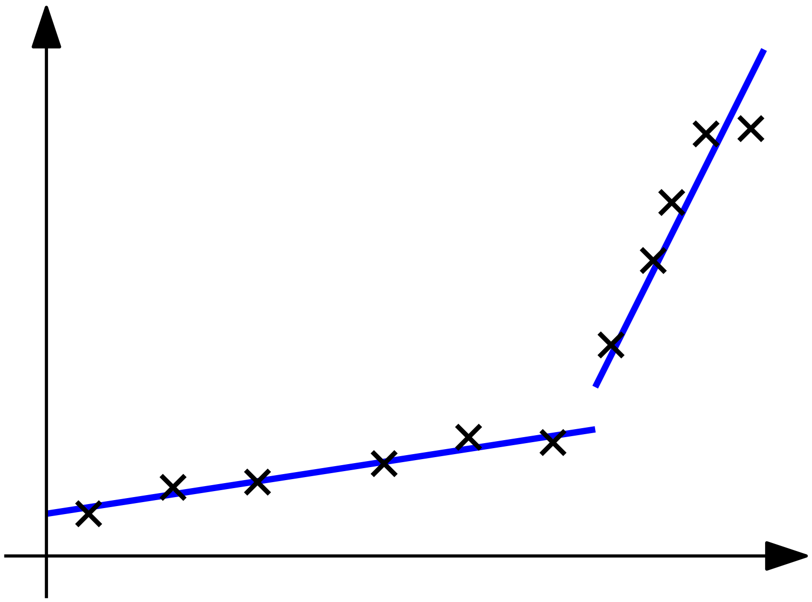


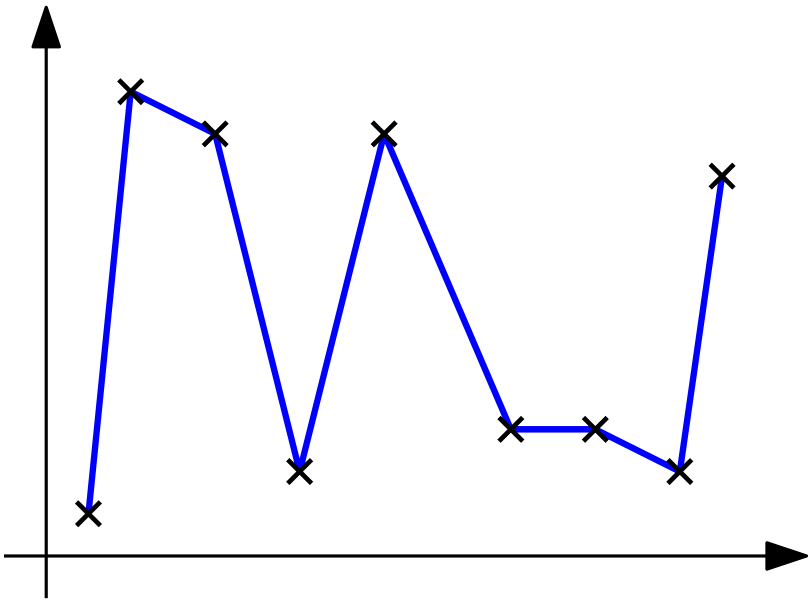
- No single line provides a good fit
- We can get a good fit with using **piecewise linear functions** instead of lines

Problem: if we use piecewise linear functions, then we can trivially fit all points

Segmented Least Squares Regression

What if the points look like this?





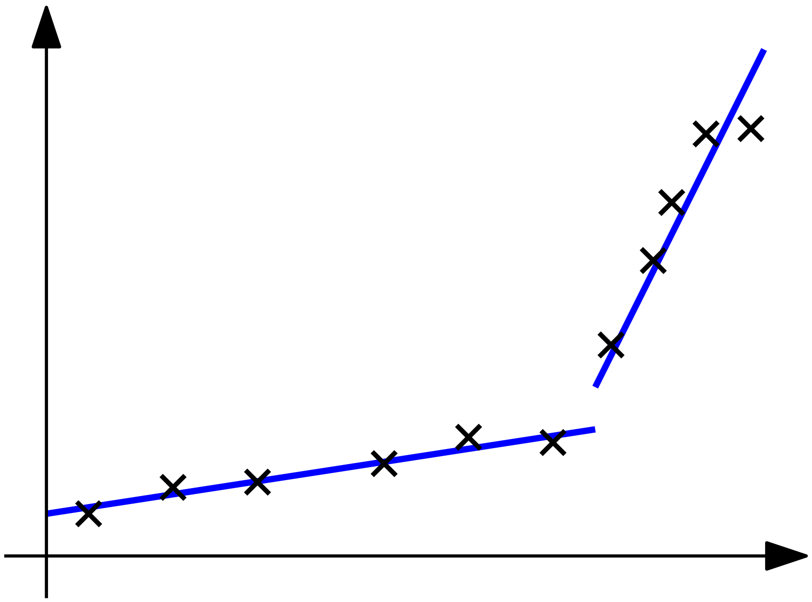
- No single line provides a good fit
- We can get a good fit with using **piecewise linear functions** instead of lines

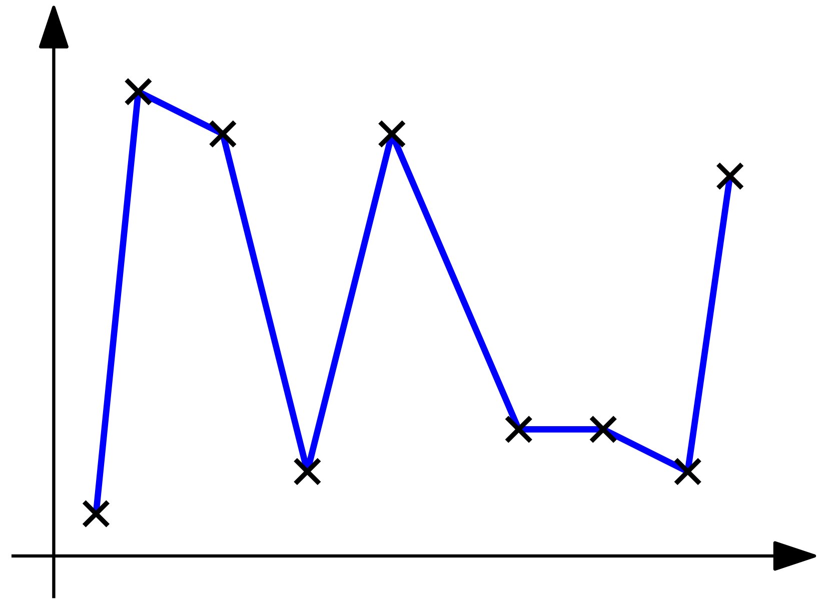
Problem: if we use piecewise linear functions, then we can trivially fit all points

Idea: each used segment incurs some cost C

Segmented Least Squares Regression

What if the points look like this?





- No single line provides a good fit
- We can get a good fit with using **piecewise linear functions** instead of lines

Problem: if we use piecewise linear funct
trivially fit all points

Balances the quality of the fit with the # of used segments

Idea: each used segment incurs some cost C

The Segmented Least Squares Problem

Input:

- A collection $P = \{p_1 = (x_1, x_2), \ldots, p_n = (x_n, y_n)\}$ of points with $x_1 < x_2 < \ldots < x_n$
- A cost $C \in \mathbb{R}^+$

The Segmented Least Squares Problem

Input:

- A collection $P = \{p_1 = (x_1, x_2), \ldots, p_n = (x_n, y_n)\}$ of points with $x_1 < x_2 < \ldots < x_n$
- A cost $C \in \mathbb{R}^+$

Output:

- A partition of P into some number k of *segments* $S_1, \ldots, S_k$
- where a segment is a subset of P containing a contiguous interval of points, i.e., $\{p_i, p_{i+1}, \ldots, p_j\}$ for some $i \leq j$,
- that minimizes the *total penalty* $C \cdot k + \sum_{i=1}^{k} \mathrm{Err}(\ell_i, S_i)$, where ℓ_i is the best fit line for S_i

The Segmented Least Squares Problem

Input:

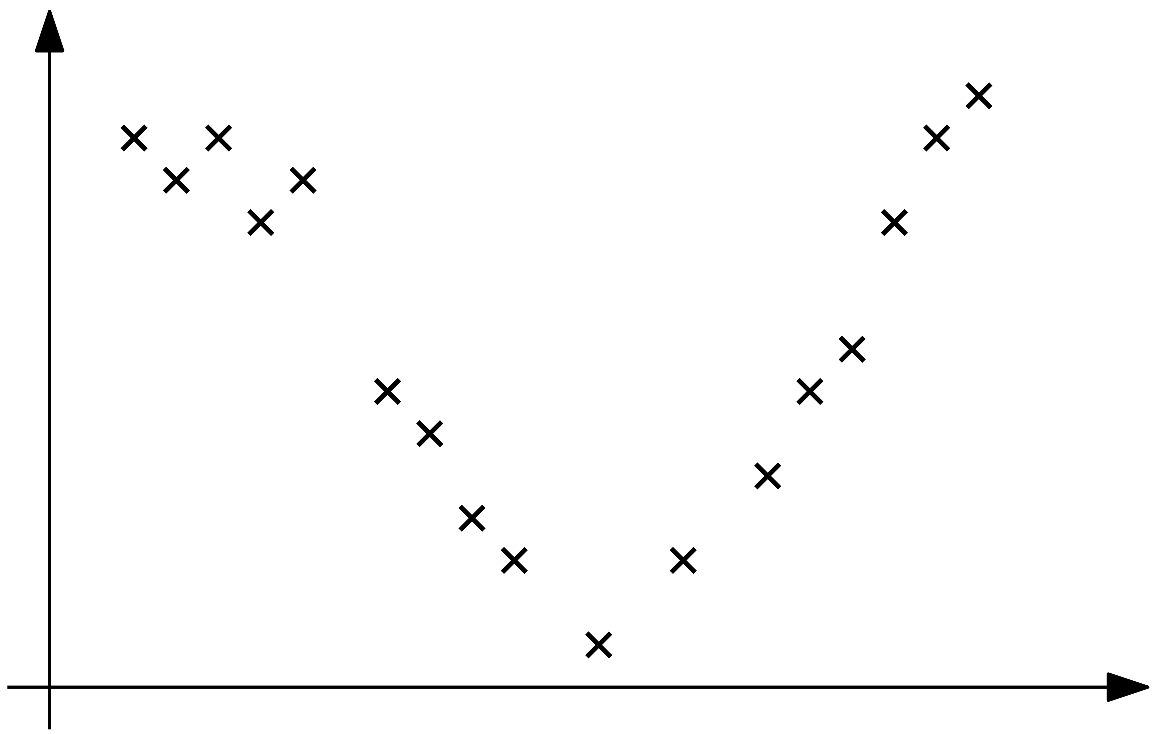
- A collection $P = \{p_1 = (x_1, x_2), \ldots, p_n = (x_n, y_n)\}$ of points with $x_1 < x_2 < \ldots < x_n$
- A cost $C \in \mathbb{R}^+$

Output:

- A partition of P into some number k of *segments* $S_1, \ldots, S_k$
- where a segment is a subset of P containing a contiguous interval of points, i.e., $\{p_i, p_{i+1}, \ldots, p_j\}$ for some $i \le j$,
- that minimizes the *total penalty* $C \cdot k + \sum_{i=1}^{k}$ Err(ℓ_i, S_i), where ℓ_i is the best fit line for S_i

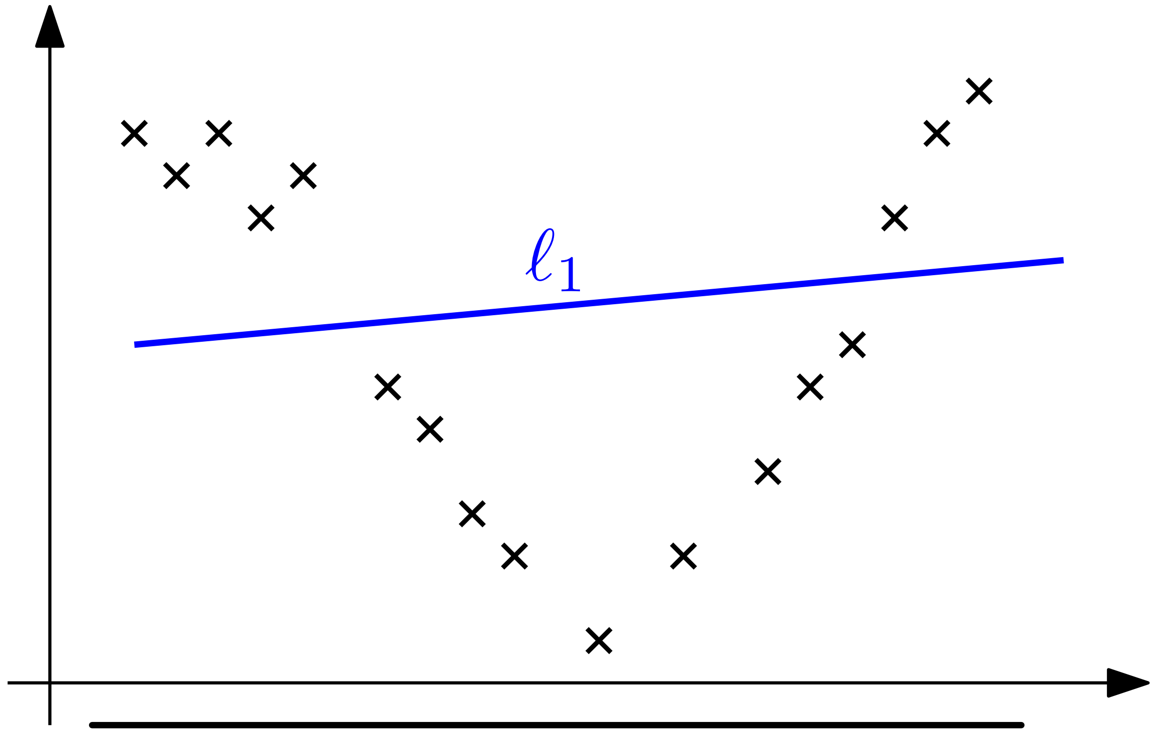
Err(S_i)

Example (qualitative)



$C = 5$

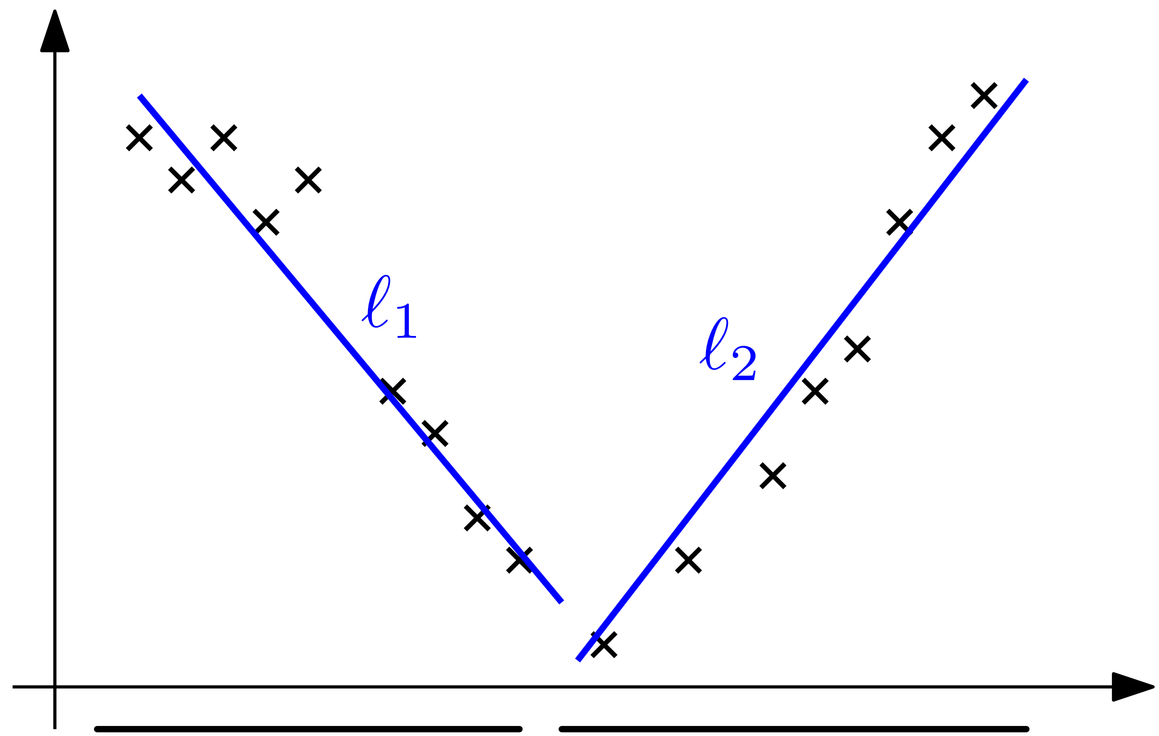
Example (qualitative)



$C = 5$

One segment: Total penalty: $1 \cdot 5 + 200 = 205$

Example (qualitative)

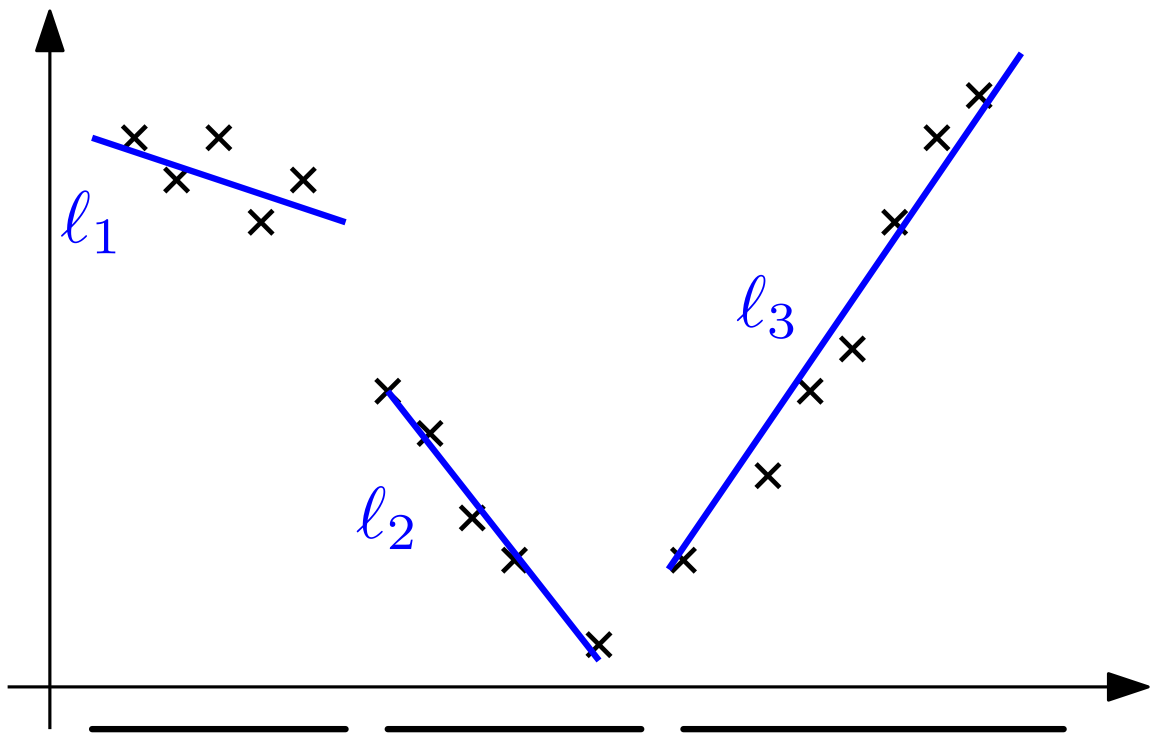


$C = 5$

One segment: Total penalty: $1 \cdot 5 + 200 = 205$

Two segments: Total penalty: $2 \cdot 5 + 50 = 60$

Example (qualitative)



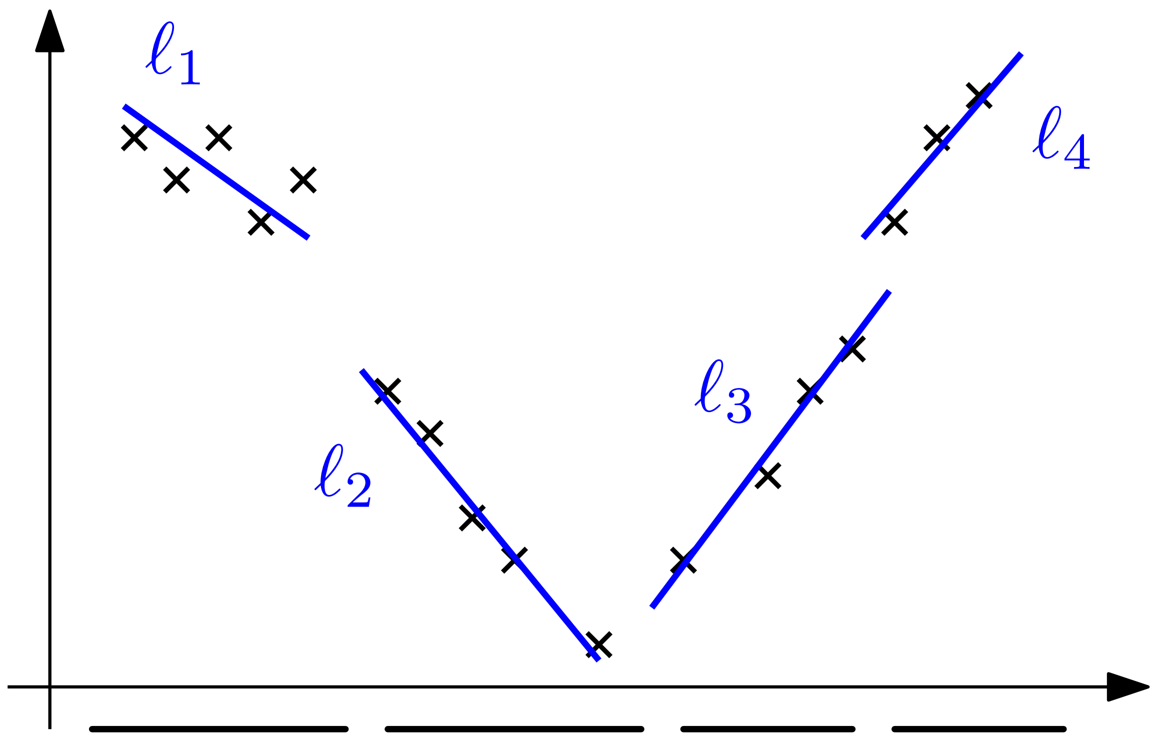
$C = 5$

One segment: Total penalty: $1 \cdot 5 + 200 = 205$

Two segments: Total penalty: $2 \cdot 5 + 50 = 60$

Three segments: Total penalty: $3 \cdot 5 + 40 = 55$

Example (qualitative)



$C = 5$

One segment: Total penalty: $1 \cdot 5 + 200 = 205$

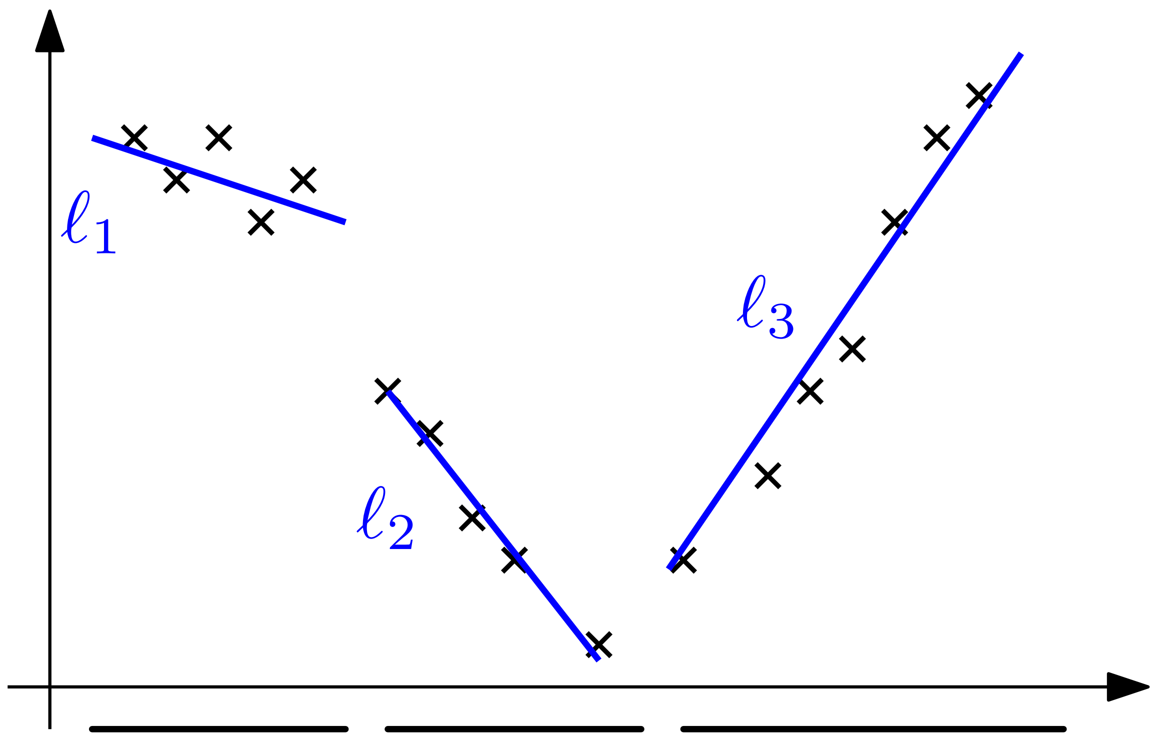
Two segments: Total penalty: $2 \cdot 5 + 50 = 60$

Three segments: Total penalty: $3 \cdot 5 + 40 = 55$

Four segments: Total penalty: $4 \cdot 5 + 37 = 57$

Example (qualitative)

$C = 5$



One segment: Total penalty: $1 \cdot 5 + 200 = 205$

Two segments: Total penalty: $2 \cdot 5 + 50 = 60$

Three segments: Total penalty: $3 \cdot 5 + 40 = 55$

Four segments: Total penalty: $4 \cdot 5 + 37 = 57$

A Key Observation

Let $\text{OPT}(i)$ denote the penalty incurred by an optimal solution for the instance that only considers the points in $\{p_1, p_2, \ldots, p_i\}$

A Key Observation

Let $\text{OPT}(i)$ denote the penalty incurred by an optimal solution for the instance that only considers the points in $\{p_1, p_2, \ldots, p_i\}$

If we know that the last segment in an optimal solution is $p_j, \ldots .p_n$, then the value of the optimal solution is

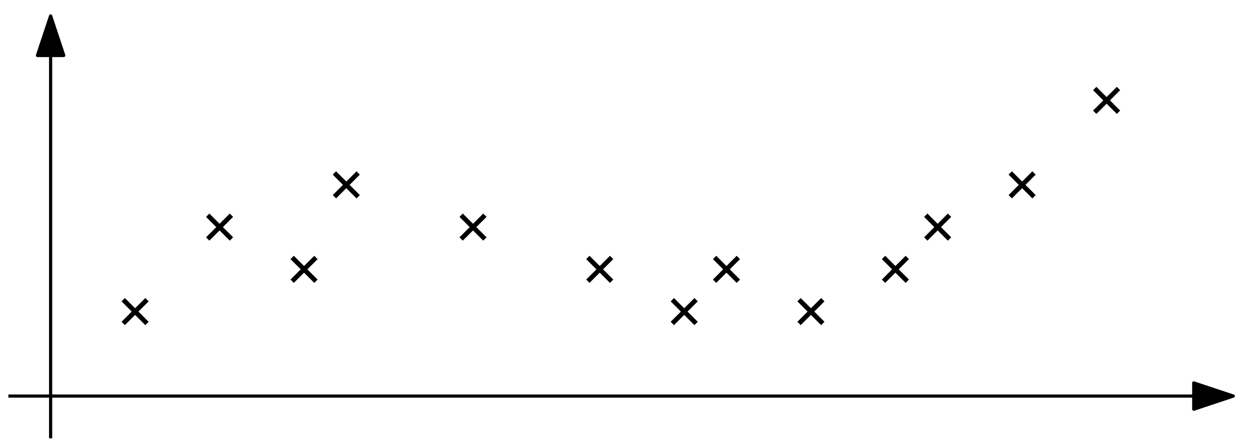
$$\text{OPT}(n) = C + \text{Err}(\{p_j, \ldots, p_n\}) + \text{OPT}(j-1).$$

A Key Observation

Let $\mathsf{OPT}(i)$ denote the penalty incurred by an optimal solution for the instance that only considers the points in $\{p_1, p_2, \ldots, p_i\}$

If we know that the last segment in an optimal solution is $p_j, \ldots .p_n$, then the value of the optimal solution is

$$\mathsf{OPT}(n) = C + \mathsf{Err}(\{p_j, \ldots, p_n\}) + \mathsf{OPT}(j-1).$$



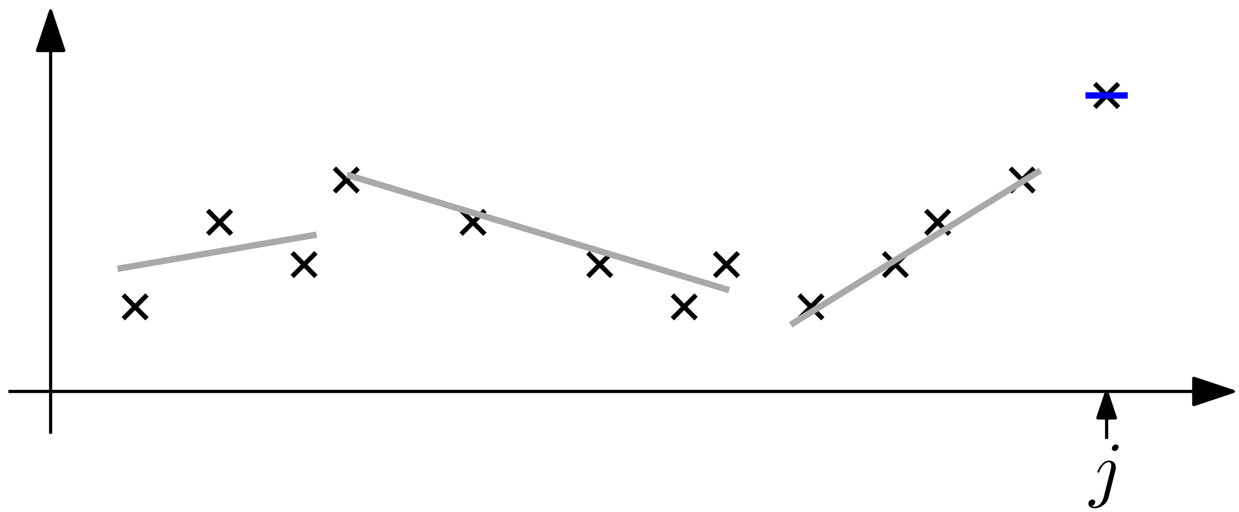
Idea: try all the possible choices for the last segment

A Key Observation

Let $\mathsf{OPT}(i)$ denote the penalty incurred by an optimal solution for the instance that only considers the points in $\{p_1, p_2, \ldots, p_i\}$

If we know that the last segment in an optimal solution is $p_j, \ldots . p_n$, then the value of the optimal solution is

$$\mathsf{OPT}(n) = C + \mathsf{Err}(\{p_j, \ldots, p_n\}) + \mathsf{OPT}(j-1).$$



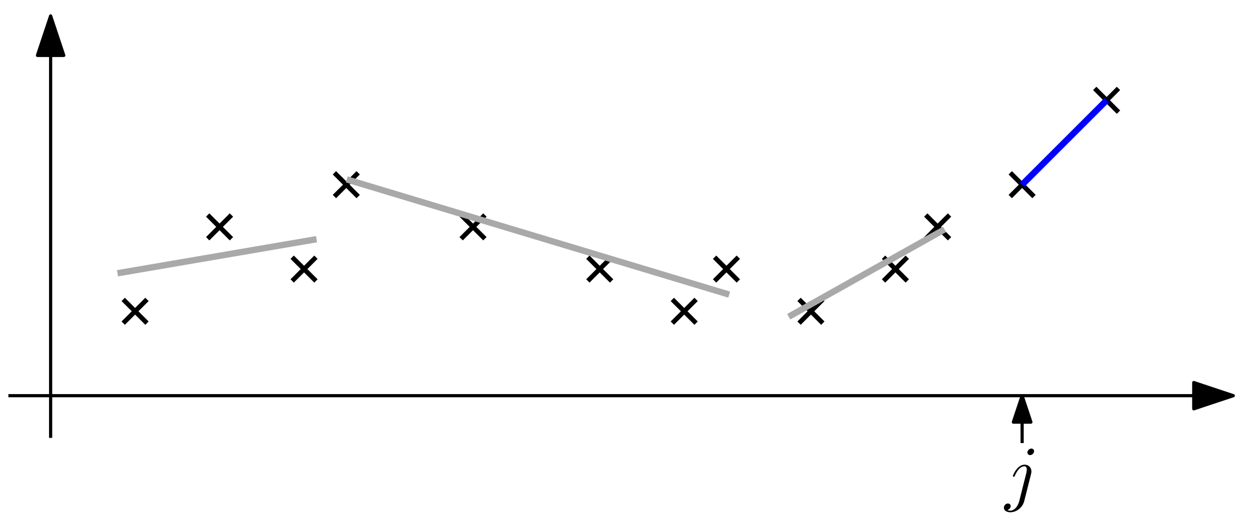
Idea: try all the possible choices for the last segment

A Key Observation

Let $\mathsf{OPT}(i)$ denote the penalty incurred by an optimal solution for the instance that only considers the points in $\{p_1, p_2, \ldots, p_i\}$

If we know that the last segment in an optimal solution is $p_j, \ldots . p_n$, then the value of the optimal solution is

$$\mathsf{OPT}(n) = C + \mathsf{Err}(\{p_j, \ldots, p_n\}) + \mathsf{OPT}(j-1).$$



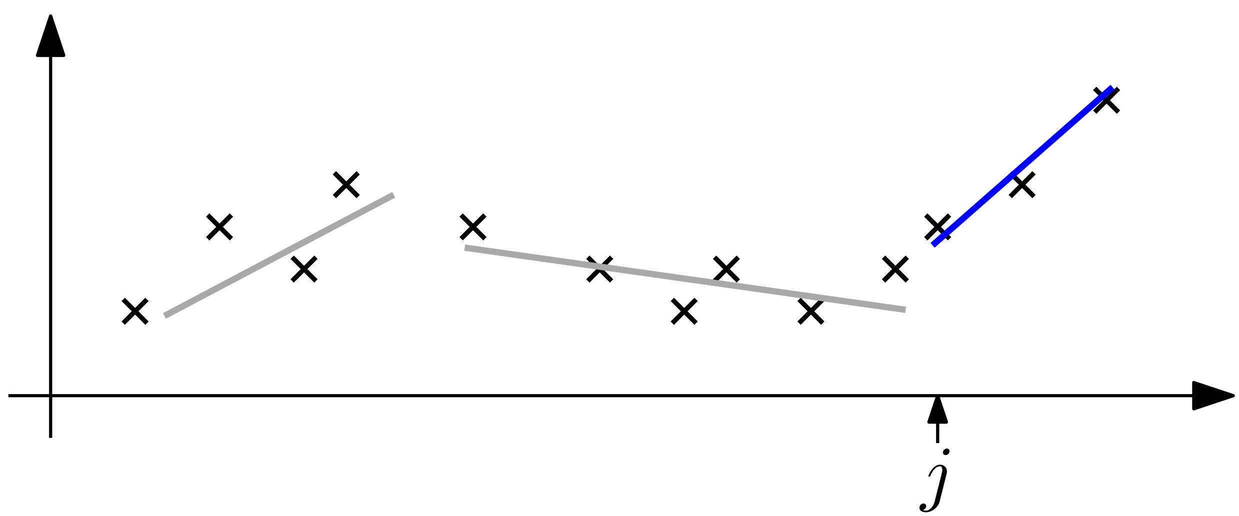
Idea: try all the possible choices for the last segment

A Key Observation

Let $\mathsf{OPT}(i)$ denote the penalty incurred by an optimal solution for the instance that only considers the points in $\{p_1, p_2, \ldots, p_i\}$

If we know that the last segment in an optimal solution is $p_j, \ldots .p_n$, then the value of the optimal solution is

$$\mathsf{OPT}(n) = C + \mathsf{Err}(\{p_j, \ldots, p_n\}) + \mathsf{OPT}(j-1).$$



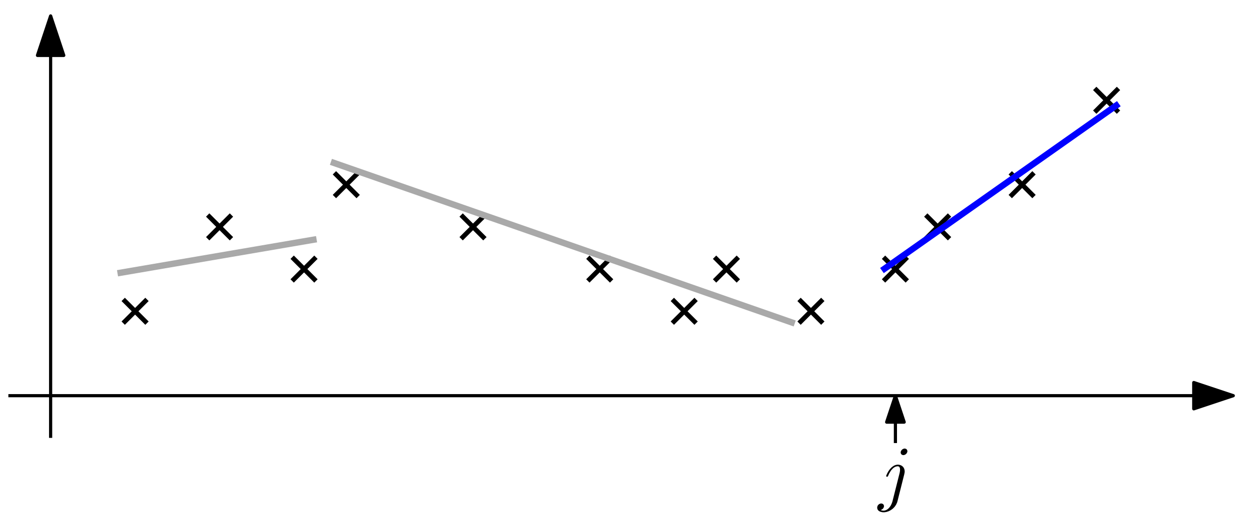
Idea: try all the possible choices for the last segment

A Key Observation

Let $\mathrm{OPT}(i)$ denote the penalty incurred by an optimal solution for the instance that only considers the points in $\{p_1, p_2, \ldots, p_i\}$

If we know that the last segment in an optimal solution is $p_j, \ldots . p_n$, then the value of the optimal solution is

$$\mathrm{OPT}(n) = C + \mathrm{Err}(\{p_j, \ldots, p_n\}) + \mathrm{OPT}(j-1).$$



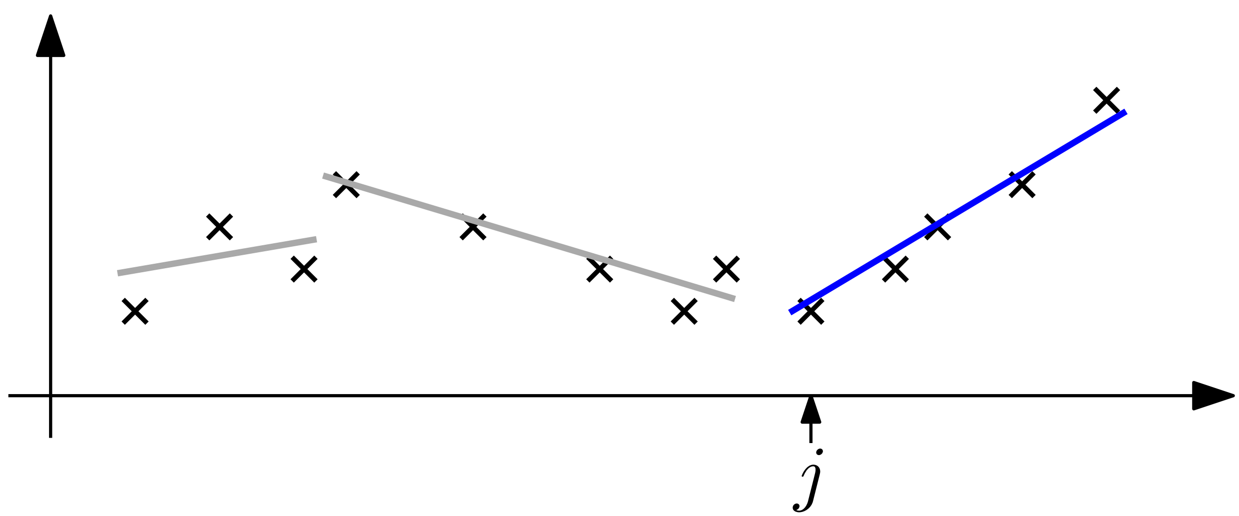
Idea: try all the possible choices for the last segment

A Key Observation

Let $\mathrm{OPT}(i)$ denote the penalty incurred by an optimal solution for the instance that only considers the points in $\{p_1, p_2, \ldots, p_i\}$

If we know that the last segment in an optimal solution is $p_j, \ldots .p_n$, then the value of the optimal solution is

$$\mathrm{OPT}(n) = C + \mathrm{Err}(\{p_j, \ldots, p_n\}) + \mathrm{OPT}(j-1).$$



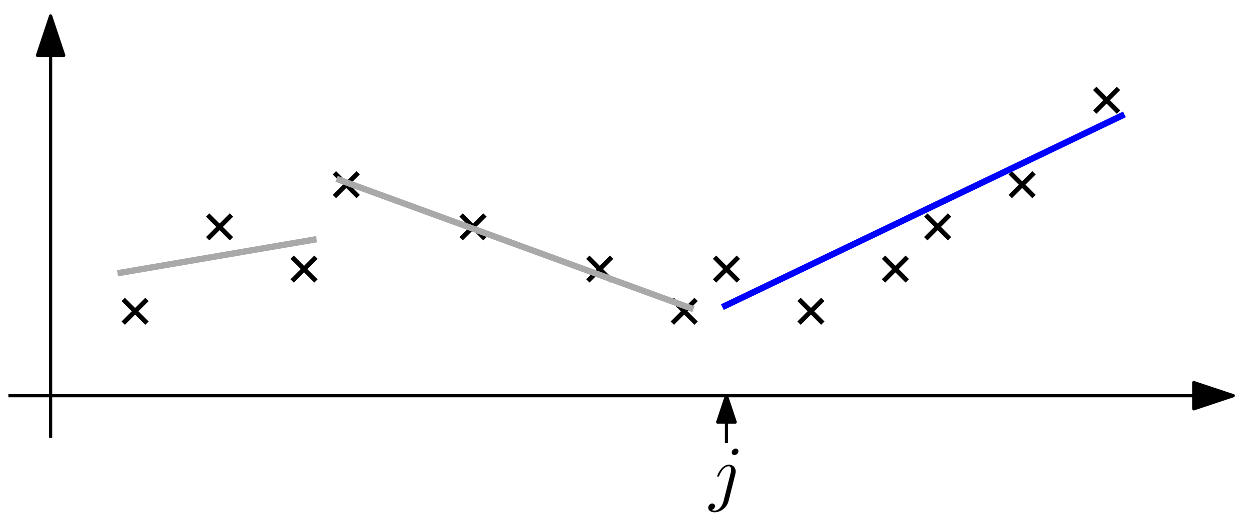
Idea: try all the possible choices for the last segment

A Key Observation

Let $\text{OPT}(i)$ denote the penalty incurred by an optimal solution for the instance that only considers the points in $\{p_1, p_2, \ldots, p_i\}$

If we know that the last segment in an optimal solution is $p_j, \ldots .p_n$, then the value of the optimal solution is

$$\text{OPT}(n) = C + \text{Err}(\{p_j, \ldots, p_n\}) + \text{OPT}(j-1).$$



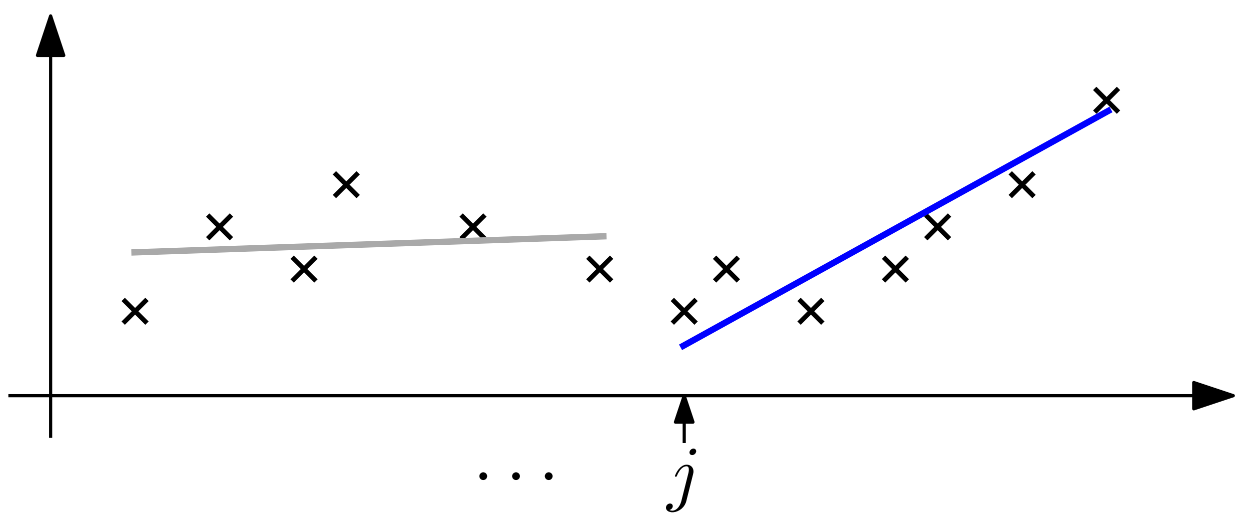
Idea: try all the possible choices for the last segment

A Key Observation

Let $\mathsf{OPT}(i)$ denote the penalty incurred by an optimal solution for the instance that only considers the points in $\{p_1, p_2, \ldots, p_i\}$

If we know that the last segment in an optimal solution is $p_j, \ldots .p_n$, then the value of the optimal solution is

$$\mathsf{OPT}(n) = C + \mathsf{Err}(\{p_j, \ldots, p_n\}) + \mathsf{OPT}(j-1).$$



Idea: try all the possible choices for the last segment

A Dynamic Programming Algorithm

Subproblem definition:

OPT(i) = penalty incurred by an optimal solution for the instance that only considers the points in $\{p_1, p_2, \ldots, p_i\}$

Base case: OPT$(0) = 0$

Recursive formula: (for $i \geq 1$)

$$\text{OPT}(i) = \min_{j=1,\ldots,i} \left\{ C + \text{Err}(\{p_j, \ldots, p_n\}) + \text{OPT}(j-1) \right\}.$$

Order of subproblems: OPT(1), OPT(2), . . . , OPT(n)

Solution: OPT(n)

Time Complexity

How many subproblems?

Time Complexity

How many subproblems? $O(n)$

Time Complexity

How many subproblems? $O(n)$

How much time per subproblem?

- We need to test $O(n)$ choices of j.
- How much time per choice of j?

Time Complexity

How many subproblems? $O(n)$

How much time per subproblem?

- We need to test $O(n)$ choices of j.
- How much time per choice of j?
  — Need to find the best fit line ℓ for $\{p_j, \ldots, p_i\}$ $O(n)$
  — Need to find the error $\mathsf{Err}(\ell, \{p_j, \ldots, p_i\})$ $O(n)$

Time Complexity

How many subproblems? $O(n)$

How much time per subproblem? $O(n^2)$

- We need to test $O(n)$ choices of j.
- How much time per choice of j?

— Need to find the best fit line ℓ for $\{p_j, \ldots, p_i\}$ $O(n)$

— Need to find the error $\text{Err}(\ell, \{p_j, \ldots, p_i\})$ $O(n)$

Overall time: $O(n^3)$

Time Complexity

How many subproblems? $O(n)$

How much time per subproblem? $O(n^2)$

- We need to test $O(n)$ choices of j.

- How much time per choice of j?

  — Need to find the best fit line ℓ for $\{p_j, \ldots, p_i\}$ $O(n)$

  — Need to find the error Err$(\ell, \{p_j, \ldots, p_i\})$ $O(n)$

Overall time: $O(n^3)$

Can we do better?

Improving the Time Complexity

Goal: find the best fit line $\ell(x) = a \cdot x + b$, and the error $\text{Err}(\ell, \{p_j, \ldots, p_i\})$ **quickly**

$$a = \frac{n \sum_{h=j}^{i} x_h y_h - \left(\sum_{h=j}^{i} x_h\right)\left(\sum_{h=j}^{i} y_h\right)}{n \sum_{h=j}^{i} x_h^2 - \left(\sum_{h=j}^{i} x_h\right)^2}$$

$$b = \frac{\sum_{h=j}^{i} y_h - a \sum_{h=j}^{i} x_h}{n}$$

$$\text{Err}(\ell, \{p_j, \ldots, p_i\}) = \sum_{h=j}^{i} \left(\ell(x_h) - y_h\right)^2$$

Finding ℓ quickly

$$a = \frac{n \sum_{h=j}^{i} x_h y_h - \left(\sum_{h=j}^{i} x_h \right) \left(\sum_{h=j}^{i} y_h \right)}{n \sum_{h=j}^{i} x_h^2 - \left(\sum_{h=j}^{i} x_h \right)^2}$$

$$b = \frac{\sum_{h=j}^{i} y_h - a \sum_{h=j}^{i} x_h}{n}$$

Finding ℓ quickly

Can we find this quantity quickly?

$$a = \frac{n\sum_{h=j}^{i} x_h y_h - \left(\sum_{h=j}^{i} x_h\right)\left(\sum_{h=j}^{i} y_h\right)}{n\sum_{h=j}^{i} x_h^2 - \left(\sum_{h=j}^{i} x_h\right)^2}$$

$$b = \frac{\sum_{h=j}^{i} y_h - a\sum_{h=j}^{i} x_h}{n}$$

Finding ℓ quickly

Can we find this quantity quickly?

$$a = \frac{n\sum_{h=j}^{i} x_h y_h - \left(\sum_{h=j}^{i} x_h\right)\left(\sum_{h=j}^{i} y_h\right)}{n\sum_{h=j}^{i} x_h^2 - \left(\sum_{h=j}^{i} x_h\right)^2}$$

What about this one?

$$b = \frac{\sum_{h=j}^{i} y_h - a\sum_{h=j}^{i} x_h}{n}$$

Finding ℓ quickly

Can we find this quantity quickly?

$$a = \frac{n\sum_{h=j}^{i} x_h y_h - \left(\boxed{\sum_{h=j}^{i} x_h}\right)\left(\boxed{\sum_{h=j}^{i} y_h}\right)}{n\boxed{\sum_{h=j}^{i} x_h^2} - \left(\boxed{\sum_{h=j}^{i} x_h}\right)^2}$$

What about this one?

What about this one?

$$b = \frac{\boxed{\sum_{h=j}^{i} y_h} - a\boxed{\sum_{h=j}^{i} x_h}}{n}$$

Finding ℓ quickly

Can we find this quantity quickly?

What about this one?

$$a = \frac{n \boxed{\sum_{h=j}^{i} x_h y_h} - \left(\boxed{\sum_{h=j}^{i} x_h}\right)\left(\boxed{\sum_{h=j}^{i} y_h}\right)}{n \boxed{\sum_{h=j}^{i} x_h^2} - \left(\boxed{\sum_{h=j}^{i} x_h}\right)^2}$$

What about this one?

What about this one?

$$b = \frac{\boxed{\sum_{h=j}^{i} y_h} - a \boxed{\sum_{h=j}^{i} x_h}}{n}$$

Finding ℓ quickly

Can we find this quantity quickly?

What about this one?

What about this one?

$$a = \frac{n \boxed{\sum_{h=j}^{i} x_h y_h} - \left(\boxed{\sum_{h=j}^{i} x_h}\right)\left(\boxed{\sum_{h=j}^{i} y_h}\right)}{n \boxed{\sum_{h=j}^{i} x_h^2} - \left(\boxed{\sum_{h=j}^{i} x_h}\right)^2}$$

What about this one?

$$b = \frac{\boxed{\sum_{h=j}^{i} y_h} - a \boxed{\sum_{h=j}^{i} x_h}}{n}$$

All the marked quantities can be found in **constant time** after a linear-time preprocessing using **prefix sums**.

Finding $\mathsf{Err}(\cdot, \cdot)$ quickly

$$\mathsf{Err}(\ell, \{p_j, \ldots, p_i\}) = \sum_{h=j}^{i} \Big(\ell(x_h) - y_h \Big)^2 \qquad \ell(x) = a \cdot x + b$$

Finding $\mathsf{Err}(\cdot, \cdot)$ quickly

$$\mathsf{Err}(\ell, \{p_j, \ldots, p_i\}) = \sum_{h=j}^{i} \Big(\ell(x_h) - y_h \Big)^2 \qquad \ell(x) = a \cdot x + b$$

$$= \sum_{h=j}^{i} \Big(\ell(x_h)^2 + y_h^2 - 2\ell(x_h) y_h \Big)$$

Finding $\mathsf{Err}(\cdot,\cdot)$ quickly

$$\mathsf{Err}(\ell, \{p_j, \ldots, p_i\}) = \sum_{h=j}^{i} \Big(\ell(x_h) - y_h\Big)^2 \qquad \ell(x) = a \cdot x + b$$

$$= \sum_{h=j}^{i} \Big(a^2 x_h^2 + b^2 + 2a x_h b + y_h^2 - 2a x_h y_h - 2b y_h\Big)$$

Finding $\mathsf{Err}(\cdot,\cdot)$ quickly

$$\mathsf{Err}(\ell, \{p_j, \ldots, p_i\}) = \sum_{h=j}^{i} \Big(\ell(x_h) - y_h\Big)^2 \qquad \ell(x) = a \cdot x + b$$

$$= a^2 \sum_{h=j}^{i} x_h^2 + (j - i + 1)b^2 + 2ab \sum_{h=j}^{i} x_h + \sum_{h=j}^{i} y_h^2 - 2a \sum_{h=j}^{i} x_h y_h - 2b \sum_{h=j}^{i} y_h$$

Finding $\mathrm{Err}(\cdot, \cdot)$ quickly

$$\mathrm{Err}(\ell, \{p_j, \ldots, p_i\}) = \sum_{h=j}^{i} \left(\ell(x_h) - y_h \right)^2 \qquad \ell(x) = a \cdot x + b$$

$$= a^2 \sum_{h=j}^{i} x_h^2 + (j - i + 1) b^2 + 2ab \boxed{\sum_{h=j}^{i} x_h} + \sum_{h=j}^{i} y_h^2$$

$$- 2a \sum_{h=j}^{i} x_h y_h - 2b \sum_{h=j}^{i} y_h$$

Finding $\mathrm{Err}(\cdot,\cdot)$ quickly

$$\mathrm{Err}(\ell, \{p_j, \ldots, p_i\}) = \sum_{h=j}^{i} \left(\ell(x_h) - y_h\right)^2 \qquad \ell(x) = a \cdot x + b$$

$$= a^2 \sum_{h=j}^{i} x_h^2 + (j - i + 1)b^2 + 2ab \boxed{\sum_{h=j}^{i} x_h} + \sum_{h=j}^{i} y_h^2 - 2a \sum_{h=j}^{i} x_h y_h - 2b \boxed{\sum_{h=j}^{i} y_h}$$

Finding Err$(\cdot, \cdot)$ quickly

$$\mathrm{Err}(\ell, \{p_j, \ldots, p_i\}) = \sum_{h=j}^{i} \Big(\ell(x_h) - y_h \Big)^2 \qquad \ell(x) = a \cdot x + b$$

$$= a^2 \boxed{\sum_{h=j}^{i} x_h^2} + (j - i + 1) b^2 + 2ab \boxed{\sum_{h=j}^{i} x_h} + \sum_{h=j}^{i} y_h^2 - 2a \sum_{h=j}^{i} x_h y_h - 2b \boxed{\sum_{h=j}^{i} y_h}$$

Finding $\mathrm{Err}(\cdot, \cdot)$ quickly

$$\mathrm{Err}(\ell, \{p_j, \ldots, p_i\}) = \sum_{h=j}^{i} \left(\ell(x_h) - y_h \right)^2 \qquad \ell(x) = a \cdot x + b$$

$$= a^2 \boxed{\sum_{h=j}^{i} x_h^2} + (j - i + 1) b^2 + 2ab \boxed{\sum_{h=j}^{i} x_h} + \sum_{h=j}^{i} y_h^2$$

$$- 2a \boxed{\sum_{h=j}^{i} x_h y_h} - 2b \boxed{\sum_{h=j}^{i} y_h}$$

Finding Err$(\cdot, \cdot)$ quickly

$$\text{Err}(\ell, \{p_j, \ldots, p_i\}) = \sum_{h=j}^{i} \Big(\ell(x_h) - y_h \Big)^2 \qquad \ell(x) = a \cdot x + b$$

$$= a^2 \boxed{\sum_{h=j}^{i} x_h^2} + (j - i + 1) b^2 + 2ab \boxed{\sum_{h=j}^{i} x_h} + \boxed{\sum_{h=j}^{i} y_h^2} - 2a \boxed{\sum_{h=j}^{i} x_h y_h} - 2b \boxed{\sum_{h=j}^{i} y_h}$$

All the marked quantities can be found in **constant time** after a linear-time preprocessing using **prefix sums**.

Time Complexity

How many subproblems? $O(n)$

How much time per subproblem? $O(n^2)$

- We need to test $O(n)$ choices of j.
- How much time per choice of j?

— Need to find the best fit line ℓ for $\{p_j, \ldots, p_i\}$ $O(n)$

— Need to find the error $\mathrm{Err}(\ell, \{p_j, \ldots, p_i\})$ $O(n)$

Time Complexity

How many subproblems? $O(n)$

How much time per subproblem? $O(n)$ $\cancel{O(n^2)}$

- We need to test $O(n)$ choices of j.

- How much time per choice of j?

$O(1)$

— Need to find the best fit line ℓ for $\{p_j, \ldots, p_i\}$ $\cancel{O(n)}$

— Need to find the error $\mathrm{Err}(\ell, \{p_j, \ldots, p_i\})$ $\cancel{O(n)}$

$O(1)$

Time Complexity

How many subproblems? $O(n)$

How much time per subproblem? $O(n)$ $\cancel{O(n^2)}$

- We need to test $O(n)$ choices of j.

- How much time per choice of j?

$O(1)$

— Need to find the best fit line ℓ for $\{p_j, \ldots, p_i\}$ $\cancel{O(n)}$

— Need to find the error $\text{Err}(\ell, \{p_j, \ldots, p_i\})$ $\cancel{O(n)}$

$O(1)$

$+\,O(n)$-**time preprocessing** (one-time only)

Time Complexity

How many subproblems? $O(n)$

How much time per subproblem? $O(n)$ $\cancel{O(n^2)}$

- We need to test $O(n)$ choices of j.

- How much time per choice of j?

$O(1)$

— Need to find the best fit line ℓ for $\{p_j, \ldots, p_i\}$ $\cancel{O(n)}$

— Need to find the error $\mathrm{Err}(\ell, \{p_j, \ldots, p_i\})$ $\cancel{O(n)}$

$O(1)$

$+\,O(n)$-**time preprocessing** (one-time only)

Overall time: $O(n^2)$