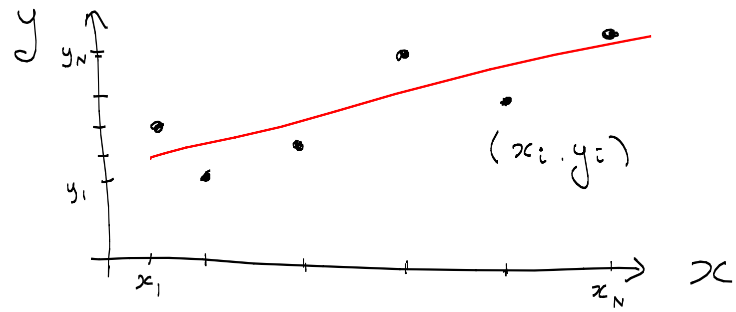
lecture 13

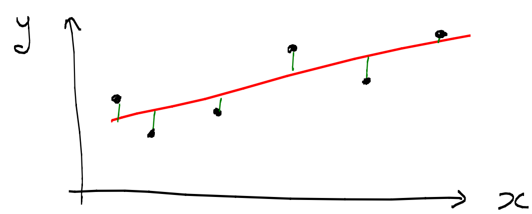
Segmented least squares

- least squares (introduction)
- segmentation
- dynamic programming solution

Least Squares



Fit a line to a set of points.



Model: $y_i = a x_i + b + \varepsilon_i$

error due to
e.g. – measurement
– wrong model
(ignoring non-linearity)

$$y_i = a x_i + b + \varepsilon_i$$

"Least squares" – find the model parameters a and b that minimize the sum of squared errors:

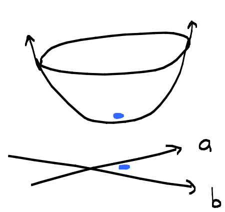
$$\sum_{i=1}^{N} \varepsilon_i^2 \equiv \sum_{i=1}^{N} (y_i - a x_i - b)^2$$

why squared? why not absolute value?

$$\sum_{i=1}^{N} \varepsilon_i^2 \equiv \sum_{i=1}^{N} (y_i - a x_i - b)^2$$

- a, b are the model parameters
- x_i y_i are input data (fixed)

Claim: If we vary the parameters of the line over all (a, b), there is a single minimum.



How to find the (a,b) that minimizes $\sum_i \varepsilon_i^2$?

Calculus III
$$\left\{ \begin{array}{l} \frac{\partial}{\partial a} \sum_i (y_i - a x_i - b)^2 = 0 \\ \frac{\partial}{\partial b} \sum_i (y_i - a x_i - b)^2 = 0 \end{array} \right.$$

$$\Rightarrow \left\{ \begin{array}{l} \sum_i 2 x_i (y_i - a x_i - b) = 0 \\ \sum_i 2 (y_i - a x_i - b) = 0 \end{array} \right.$$

Which we rewrite as two linear equations with variables (a, b). See next slide.

$$\left(\sum_{i=1}^{N} x_i^2\right) a + \left(\sum_{i=1}^{N} x_i\right) b = \sum_{i=1}^{N} x_i y_i$$

$$\left(\sum_{i=1}^{N} x_i\right) a + N b = \sum_{i=1}^{N} y_i$$

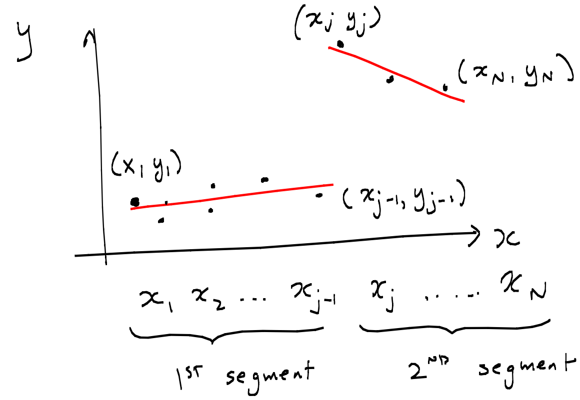
and solve :

$$a = \frac{\sum y_i \sum x_i - N \sum x_i y_i}{(\sum x_i)^2 - N \sum x_i^2}$$

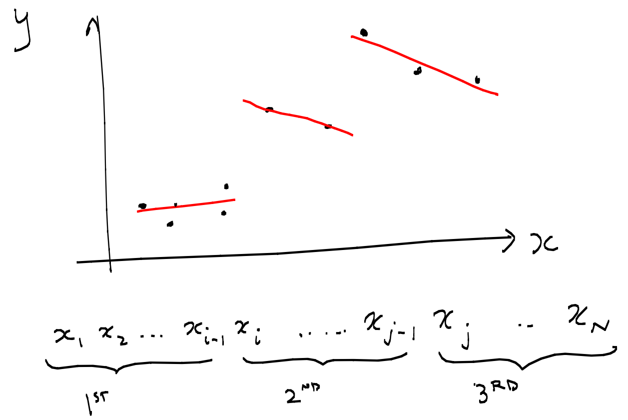
$$b = \frac{\sum x_i \sum x_i y_i - \sum x_i^2 \sum y_i}{(\sum x_i)^2 - N \sum x_i^2}$$

Segmented Least Squares

What if we allow ourselves to fit two lines ?



What if we allow ourselves to fit three lines (or k lines) ?



$x_1 \ x_2 \ x_3 \ \ldots \qquad x_{N-1} \ x_N$

How many ways can we partition this sequence into k segments ?

e.g. $(x_1 \ x_2 \ \ldots \ x_N)$ 1 segment

$(x_1 \ x_2 \ x_3)(x_4, \ldots \ x_N)$ 2 segments

$(x_1 \ x_2 \ x_3 \ x_4, \ldots)(x_N)$ 2 segments

$\vdots$

$(x_1, x_2)(x_3 \ x_4)(x_5 \ldots \ x_N)$ 3 segments

$\vdots$

$(x_1)(x_2)(x_3)(x_4) \ldots (x_N)$ N segments

Claim: There are 2^{N-1} ways can we segment $x_1 \ x_2 \ \ldots \ x_N$.

(I will return to this at end of lecture .)

Example (N=6)

```
['abcdef'],
['a', 'bcdef'],
['ab', 'cdef'],
['abc', 'def'],
['abcd', 'ef'],
['abcde', 'f'],
['a', 'b', 'cdef'],
['a', 'bc', 'def'],
['a', 'bcd', 'ef'],
['a', 'bcde', 'f'],
['ab', 'c', 'def'],
['ab', 'cd', 'ef'],
['ab', 'cde', 'f'],
['abc', 'd', 'ef'],
['abc', 'de', 'f'],
['abcd', 'e', 'f'],
['a', 'b', 'c', 'def'],
['a', 'b', 'cd', 'ef'],
['a', 'b', 'cde', 'f'],
['a', 'bc', 'd', 'ef'],
['a', 'bc', 'de', 'f'],
['a', 'bcd', 'e', 'f'],
['ab', 'c', 'd', 'ef'],
['ab', 'c', 'de', 'f'],
['ab', 'cd', 'e', 'f'],
['abc', 'd', 'e', 'f'],
['a', 'b', 'c', 'd', 'ef'],
['a', 'b', 'c', 'de', 'f'],
['a', 'b', 'cd', 'e', 'f'],
['a', 'bc', 'd', 'e', 'f'],
['ab', 'c', 'd', 'e', 'f'],
['a', 'b', 'c', 'd', 'e', 'f']
```

We would like to segment the data and fit lines such that

- the errors for each segment are small
- use as few lines as we can (to avoid "over fitting" e.g. if segments are two points each is, $N/2$ segments, then we will have zero fitting error)

Note there is a tradeoff here !

Cost of a segmentation

$(x_1 \ldots)(\ldots x_{i-1})\ (x_i \quad x_N)$

$\equiv \sum_{\text{segments}}$ least squares fitting error for each segment

$+ \ C \cdot (\text{number of segments})$

cost for each segment
(make this bigger if you want to guard against over fitting)

Approach

1.) $x_1\ x_2\ x_3 \ldots (x_i \ldots x_j) \ldots x_N$

Solve least squares fit for each (i, j) where $i < j$

2.) Solve the combinatorial problem of choosing a segmentation – using dynamic programming

1)

Suppose $(x_i, \ldots x_j)$ is a segment.
Let e_{ij} be the least squares fitting error for that segment.

Same as at beginning of today's lecture:

Fit line to $\{(x_\ell, y_\ell) : i \le \ell \le j\}$

$$e_{ij} \equiv \sum_{\ell=i}^{j} \varepsilon_\ell^2 \equiv \sum_{\ell=i}^{j} (y_\ell - a x_\ell - b)^2$$

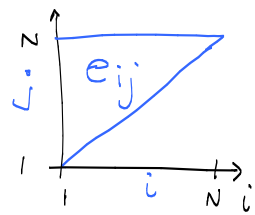
and solve for (a, b) using least squares.

Precompute e_{ij}

$e_{ii} = 0$

$e_{i,i+1} = 0$ since we can fit a line to two points

for $i = 1$ to N
  for $j = i+2$ to N
    compute e_{ij}



[Total time is $O(N^3)$ if each "compute e_{ij}" is $O(N)$, but its possible to do it in $O(N^2)$.]

Cost of a particular segmentation

$$\equiv \sum_{\substack{\text{segments} \\ (i,j)}} \text{fitting error } e_{ij} \text{ for segment } (x_i \ldots x_j) \ + \ C \cdot (\text{number of segments})$$

2

How to avoid the $O(2^N)$ possible segmentations?

Dynamic Programming: break problem into subproblems that share "sub-sub problems". Re-use the solutions of the sub-sub problems.

Let $Opt(j)$ be the minimum cost of segmentation for the problem with input $\{(x_1, y_1), (x_2, y_2), \ldots (x_j, y_j)\}$

i.e. how to segment $(x_1 \ldots x_j)$?

$Opt(0) = 0$

$Opt(1) = Opt(2) = C$.

$Opt(j) = \ ?$

$Opt(N) = \ ?$

Two Approaches

- Recursion
- Iteration

} both will use memoization to avoid combinatorial explosion

Recursion

The last segment of the optimal solution is $(x_i, x_{i+1}, \ldots x_N)$ for some unknown i.

Thus,

$$Opt(N) = \boxed{Opt(i-1)} + \boxed{e_{iN} + C}$$

Cost of segments that use $\boxed{\{x_1\ x_2 \ldots x_{i-1}\}}$

Cost of last segment $\boxed{(x_i \ldots x_N)}$

Thus, the lowest cost of segmention is

$$Opt(N) = \min_{1 \le i \le N} \left\{ \boxed{Opt(i-1)} + \boxed{e_{iN} + C} \right\}$$

N
$\{x_1 \quad \ldots \quad x_{N-1}\}\ (x_N)$

$\{x_1 \quad x_{N-2}\}\ (x_{N-1}, x_N)$

$\{x_1 \quad - \quad . \quad x_{N-3}\}\ (x_{N-2}, x_{N-1}, x_N)$

$\vdots$

i
$\{x_1\ x_2 \quad x_{i-1}\}\ (x_i, x_{i+1}, \ldots, x_N)$

$\vdots$

1
$(x_1, x_2, \quad \ldots \quad x_N)$

$$Opt(N) = \min_{1 \le i \le N} \left\{ \boxed{Opt(i-1)} + \boxed{e_{iN} + C} \right\}$$

$\Downarrow$ recursion with memoization

$$= \boxed{\sum_{\substack{\text{segments} \\ (i,j)}} e_{ij} + C \cdot (\text{number of segments})}$$

Iteration

Solve for $Opt(j)$, $j = 1$ to N

$Opt[0] = 0$ $\quad \{\}$

$Opt[1] = c$ $\quad \{x_1\}$

$Opt[2] = c$ $\quad \{x_1\ x_2\}$

$Opt[3] = ?$ $\quad \{x_1\ x_2\ x_3\}$

$\vdots$

$Opt[j] = ?$ $\quad \{x_1\ x_2\ x_3 \ldots x_j\}$

$Opt[N] = ?$ $\quad \{x_1\ x_2\ x_3 \ldots x_j \ldots x_N\}$

$Opt[0] = 0$

for $j = 1$ to N

$$Opt[j] = \min_{1 \le i \le j} \{ Opt[i-1] + e_{ij} + c \}$$

Time required for above is $O(N^2)$.

The methods described above tell us how to compute the lowest total cost. But they don't keep track of the segmentation and models.

You could modify the pseudocode to do this. Or you could do afterwards (Similar to what we saw with weighted interval scheduling.)

```
findSegments(j) {
   if j>0 {                    // find (x_i ... x_j)
      find i such that i ≤ j and
            Opt[j] == Opt[i-1] + e_ij + c

      print (i,j)
      findSegments(i-1)
   }
}
```

Note: if solution is not unique, then we might find a different segmentation here than we found before.

Recall from earlier in the lecture:

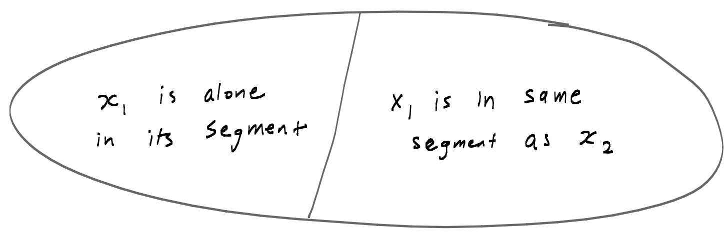
Claim: There are 2^{N-1} ways can we segment $x_1\ x_2\ \ldots\ x_N$.

Example (N=6)

```
['abcdef'],
['a', 'bcdef'],
['ab', 'cdef'],
['abc', 'def'],
['abcd', 'ef'],
['abcde', 'f'],
['a', 'b', 'cdef'],
['a', 'bc', 'def'],
['a', 'bcd', 'ef'],
['a', 'bcde', 'f'],
['ab', 'c', 'def'],
['ab', 'cd', 'ef'],
['ab', 'cde', 'f'],
['abc', 'd', 'ef'],
['abc', 'de', 'f'],
['abcd', 'e', 'f'],
['a', 'b', 'c', 'def'],
['a', 'b', 'cd', 'ef'],
['a', 'b', 'cde', 'f'],
['a', 'bc', 'd', 'ef'],
['a', 'bc', 'de', 'f'],
['a', 'bcd', 'e', 'f'],
['ab', 'c', 'd', 'ef'],
['ab', 'c', 'de', 'f'],
['ab', 'cd', 'e', 'f'],
['abc', 'd', 'e', 'f'],
['a', 'b', 'c', 'd', 'ef'],
['a', 'b', 'c', 'de', 'f'],
['a', 'b', 'cd', 'e', 'f'],
['a', 'bc', 'd', 'e', 'f'],
['ab', 'c', 'd', 'e', 'f'],
['a', 'b', 'c', 'd', 'e', 'f']
```

$x_1\ x_2\ x_3\ \ldots\ x_{N-1}\ x_N$

How many ways can we partition this sequence into k segments of consecutive elements?



Let $f(N,k)$ be the number of ways we can partition $x_1\ x_2\ \ldots\ x_N$ into k segments.

$$f(N,k) = \underbrace{f(N-1,k-1)}_{x_1 \text{ is alone in its segment}} + \underbrace{f(N-1,k)}_{x_1 \text{ is in same segment as } x_2}$$

Q: How many segmentations are there in total for all $k=1,\ldots N$?

A: $\sum_{k=1}^{N} f(N,k)$

Exercise:

Show $\sum_{k=1}^{N} f(N,k) = 2^{N-1}$