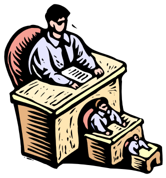
Algorithm Design and Analysis

Victor Adamchik
CS 15-451 Spring 2014

Lecture 6 Jan 27, 2014 Carnegie Mellon University

Dynamic programming - II



Outline

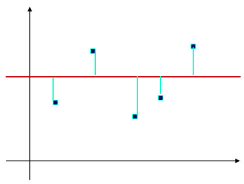
Fitting piecewise functions to data

Simple Case

Given a set of points $p_1, p_2, \ldots, p_n$ on a plane

Find a constant function $f(x) = c$, s.t.

$$\underset{c}{\mathrm{MIN}} \sum_{k=1}^{n} (c - p_k)^2$$



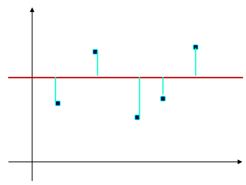
Claim: $c = \frac{1}{n}\sum_{k=1}^{n} p_k$ This is an average

Simple Case (Least Squares)

Proof.

Take a derivative and set it to 0:

$$\frac{d}{dc}\sum_{k=1}^{n} (c - p_k)^2 = 0$$



$$\frac{d}{dc}\sum_{k=1}^{n} (c - p_k)^2 = \sum_{k=1}^{n} 2(c - p_k) = 2cn - 2\sum_{k=1}^{n} p_k = 0$$

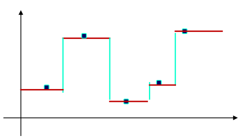
It follows, $c = \frac{1}{n}\sum_{k=1}^{n} p_k$

Piecewise Constant Segments

Rather than seek a single line, we allow a set of lines.

But this problem has a trivial solution:

Clearly we do not want to allow too many lines as well as too few.



Goal: Fund a set of constant functions $f_1, \ldots, f_n$ s.t.

$$\underset{f_1,\ldots,f_n}{\mathrm{MIN}} \left[\underbrace{\sum_{k=1}^{n} (f_k - p_k)^2}_{\text{fidelity term}} + \alpha \underbrace{\sum_{k=1}^{n-1} \delta(f_k, f_{k+1})}_{\text{smoothness term}} \right]$$

$$\delta(a,b) = \begin{cases} 0, & \text{if } a = b \\ 1, & \text{o.w.} \end{cases}$$

α is a penalty

Fitting with penalty α (given)

$$\underset{f_1,\ldots,f_n}{\mathrm{MIN}} \left[\underbrace{\sum_{k=1}^{n} (f_k - p_k)^2}_{\text{fidelity term}} + \alpha \underbrace{\sum_{k=1}^{n-1} \delta(f_k, f_{k+1})}_{\text{smoothness term}} \right]$$

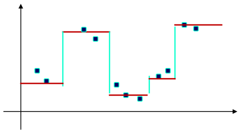
If we have too few lines f_k, we increase the fidelity term

As we increase the number of lines f_k, we increase the penalty

There are exponentially many partitions (sets of lines) , so we will be using DP.

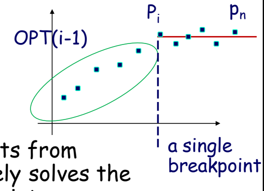
Optimal Substructure in DP

Show that a solution to a problem consists of making a choice, which results in one or more subproblems.
Suppose that you are given the last choice that leads to an optimal solution.
Given this choice, determine which subproblems are to be solved.
Show that the solutions to the subproblems are optimal.



Subproblems: an intuition

Consider the optimal solution. The last point p_n belongs to a line that starts at some p_i.



Then we remove these points from consideration and recursively solves the problem on the remaining points.

$$OPT(n) = OPT(i-1) + \left[\sum_{k=i}^{n} (c - p_k)^2 + \alpha \right]$$

Recurrence

Let $C(j)$ a subproblem for $p_1, p_2, \ldots, p_j$, $1 \le j \le n$.

We will fit these points either with a single line, or the last break is at some point p_i.

$$C(j) = \min \begin{cases} P(p_1, p_2, \ldots, p_j) \\ \min_{i<j} C(i) + \alpha + \underbrace{P(p_{i+1}, \ldots, p_j)} \end{cases}$$

$C(1) = 0$

where

$$P(p_1, \ldots, p_j) = \sum_{k=1}^{j} (c - p_k)^2$$

fitting with a constant function

Correctness

We need to prove that $C(j) = OPT(j)$.

Clearly, $OPT(j) \le C(j)$.

Proof of $C(j) \le OPT(j)$ by strong induction

Base case $j = 1$. $C(1) = 0 = OPT(1)$

IH: true for j-1 points

We add j-th point.
Case 1). all points fit by a single line

Correctness

Case 2). The last breakpoint is at p_i.

which means that $f_i \ne f_{i+1}$ and $f_{i+1} = f_{i+2}, \ldots, f_j$.

$OPT(j) = OPT(i) + \alpha + P(p_{i+1}, \ldots, p_j) \ge$

by IH

$\ge C(j-1) + \alpha + P(p_{i+1}, \ldots, p_j) = C(j)$

Runtime

$$C(j) = \min \begin{cases} P(p_1, p_2, \ldots, p_j) \\ \min_{i<j} C(i) + \alpha + P(p_{i+1}, \ldots, p_j) \end{cases}$$

$$P(p_i, \ldots, p_j) = \sum_{k=i}^{j} (c_{i,j} - p_k)^2$$

$P(p_i, \ldots, p_j) = O(i)$

For all i's $P(p_i, \ldots, p_j) = \Sigma O(i) = O(j^2)$

$C(j) = \Sigma O(j^2) = O(n^3)$

Can we do better?? Precompute $P(p_i, \ldots, p_j)$

Precompute all $P(p_i,\dots,p_j)=\sum_{k=i}^{j}(c_{i,j}-p_k)^2$

$1 \le i < j \le n$

What is the runtime complexity of precomputing?

$O(n^3)$, hmm, this does not speed up the algorithm

How about using DP?

The main problem is how to compute P for j+1 points knowing the result for j points.

Precompute all $P(p_i,\dots,p_j)=\sum_{k=i}^{j}(c-p_k)^2$

Claim1: It takes $O(n^2)$ to precompute all $P(p_i,\dots,p_j)$.

What would be the new complexity of fitting?

$$C(j)=\min\begin{cases} P(p_1,p_2,\dots,p_j) \\ \min_{i<j} C(i)+\alpha+P(p_{i+1},\dots,p_j) \end{cases}$$

We can compute each $C(j)$ in $O(j)$

Thus, the total is $O(n^2)$

Precompute all $P(p_i,\dots,p_j)=\sum_{k=i}^{j}(c-p_k)^2$

Claim2: $P = M_2 - M_1^2 / M_0$

Def: The k-th moment M_k is defined by $M_k=\sum_{j=1}^{n}p_j^k$

$M_0 = n$, $M_1 = p_1 + \dots + p_n$, $M_2 = p_1^2 + \dots + p_n^2$

Observe, if we know M_K for n points, we can compute M_k for (n+1) points in O(1).

Proof of Claim2 $\quad P = M_2 - M_1^2 / M_0$

Let us recall the first slide ("simple case") in which we showed $c=\frac{1}{n}\sum_{k=1}^{n}p_k$

This can be rewritten through moments $c=\frac{M_1}{M_0}$

The next step is all math

$$P=\sum(c-p_k)^2=\sum\left(\frac{M_1}{M_0}-p_k\right)^2=\sum\left(\frac{M_1}{M_0}\right)^2-2\frac{M_1}{M_0}\sum p_i+\sum p_i^2$$

$$P=\left(\frac{M_1}{M_0}\right)^2 M_0-2\frac{M_1}{M_0}M_1+M_2=M_2-\frac{M_1^2}{M_0}$$

Proof of Claim1

Claim1: It takes $O(n^2)$ to compute all

$$P(p_i,\dots,p_j)=M_2-M_1^2/M_0$$

$$M_k(p_i,\dots,p_j)=p_i^k+p_{i+1}^k+\dots+p_j^k$$

We use DP to compute moments!!

$$M_k(p_i,\dots,p_j)=\begin{cases} p_j^k, & \text{if } i=j \\ M_k(p_i,\dots,p_{j-1})+p_j^k, & \text{o.w.} \end{cases}$$

It has $O(n^2)$ time complexity.

Fitting in L_1

We considered fitting using least squares (L_2)

$$\underset{f_1,\dots,f_n}{\mathrm{MIN}}\left[\underbrace{\sum_{k=1}^{n}(f_k-p_k)^2}_{L_2}+\alpha\sum_{k=1}^{n-1}\delta(f_k,f_{k+1})\right]$$

However in many practical cases a variation of the above achieves a better result, namely

$$\underset{f_1,\dots,f_n}{\mathrm{MIN}}\left[\underbrace{\sum_{k=1}^{n}|f_k-p_k|}_{L_1}+\alpha\sum_{k=1}^{n-1}\delta(f_k,f_{k+1})\right]$$

Fitting in L_1

$$\underset{f_1,\ldots,f_n}{\mathrm{MIN}}\left[\sum_{k=1}^{n}|f_k-p_k|+\alpha\sum_{k=1}^{n-1}\delta(f_k,f_{k+1})\right]$$

Let us start with the simplest case – a single line

$$\underset{c}{\mathrm{MIN}}\sum_{k=1}^{n}|c-p_k|$$

How do we find c?