斜率优化基本原理

在线性规划的过程中，往往我们会确定一个方程，求其的最值。而在这个过程当中，往往会进行一些多余的，不必要的计算增加了事件复杂度，对这一类问题的优化解决，被称为斜率优化。

在求最值得问题当中，我们一般会进行平面上有限个数的有序数对的枚举：

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_n, y_n) \tag{1}$$

现引入一个一般形式的方程，设这个方程为我们要计算的方程：

$$f(i) = \min_{0 \le j < i} -K_i X_j + Y_j \tag{2}$$

其中 (X_j, Y_j) 为平面上的点，且这个方程需要满足两个条件：**K_i与X_j是递增的**。此时我们会将（1）当中的所有i所对应的j代入（2）求值，求其最小值。

设一线性函数过两点，(X_j, Y_j)，$(0, -K_i X_j + Y_j)$，则设其解析式为 $y = kx + b$，分别代入，可得以下关系：

$$0 + b = -K_i X_j + Y_j \tag{3}$$

$$b + kX_j = Y_j \tag{4}$$

联立上述两式，可得：

$$Y_j - kX_j = -K_i Y_j + Y_j \tag{5}$$

即 $k = K_i$。也就是说，$y = kx + b$ 这个线性方程的斜率与 $y = kx + b$ 在 $0 \le j < i$ 情况下的斜率相同，同样地，可以将这条直线代入表达出来，即：

$$y = K_i x + (-K_i X_j + Y_j) \tag{6}$$

问题所需求解的函数值 $f(i)$ 即为方才所求直线在y轴上的截距。

也就是说，对于固定的i，斜率 K_i 已经确定，所以直线 $y = K_i x + b$ 在不断靠近所对应的确定点集：

$$(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), ..., (X_n, Y_n) \tag{7}$$

的过程当中，这条直线所最先撞到的点，所对应的在y轴上的截距 $-K_i X_j + Y_j$ 在点集内所有值当中最小，即此时的 (X_j, Y_j) 对应了 $\min_{0 \le j < i} -K_i X_j + Y_j$，即求得此时j所对应的$f(i)$。

所以说为了确定每个 (X_j, Y_j) 所分别对应的 $\min_{0\le j<i} -K_iX_j + Y_j$ ，通常会进行遍历，随后去最小值。但是这种做法确实非常不高效的，会重复计算。此时我们尝试构建一个模型，减少所需尝试的点，使得算法最优。由于是求最小的截距，所以可以形象地理解为将直线：

$$y = K_ix + b \tag{8}$$

从负无穷远处向上不断移动，直至直线遇到点集中的任意一点，而对于最先遇到的点，必定是点集中部分点构成的下凸曲线，将所有的点处于此下图曲线线上或其上方（如下图）

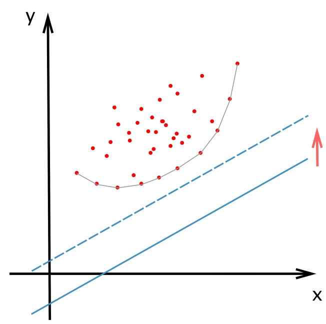

Fig.1 斜率优化

Fig 1. 斜率优化

由此，引入了凸包的概念，并将求最小值问题转换为对凸包的维护：

- 凸包：点集Q的凸包（Convex Hull）是指一个最小的凸多边形，满足Q中的点在多边形上或在其内。

引入凸包后可以发现需要判断的点都处在凸包上，处理量大大减少。在图中，凸包即为部分的下图曲线上的点集，同时，凸包满足斜率不断增加的条件，换而言之，其各个部分导数不断增加，若使用光滑的曲线连接各点使其连续且可导，则其二阶导数始终大于0（函数下凸）。

对凸包的维护：

（1）对于一个凸包，每当i增加时，会有一个新的点，而由于 X_j 递增，所以新点 (X_{i+1}, Y_{i+1}) 必定在最右侧，一定在凸包上，需将此点加入凸包，但是加入的同时必须要使其符合斜率不断增加的定义，即 $\frac{y_{i+1}-y_i}{x_{i+1}-x_i}$ 是递增的，所以可能会需要删除部分点，使其满足条件。进行判断：当

$$\frac{y_t - y_{t-1}}{x_t - x_{t-1}} \ge \frac{y_i - y_t}{x_i - x_t} \tag{9}$$

时，舍去第 t 个数对（点），t为每次点集的最后一项。重复上述过程直至（9）的条件不再成立。如下图：

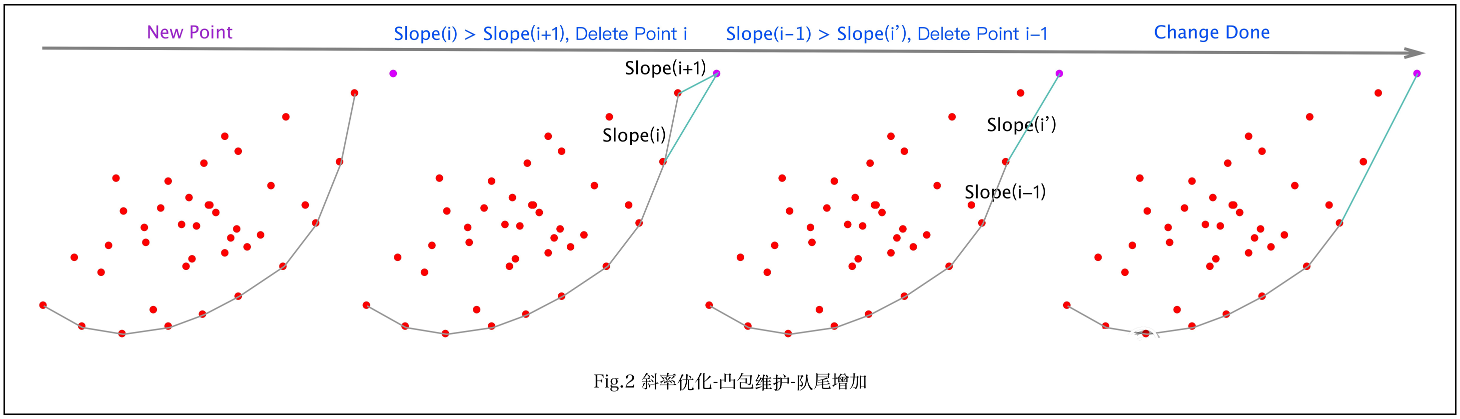

Fig 2. 斜率优化-凸包维护-队尾增加

（2）首先进行以下推导：

由于对于凸包的队首而言，如果两点 $(X_1, Y_1), (X_2, Y_2)$，过前者的直线与过后者的平行直线（$k = K_i$）前者截y轴的截距大于后者截y轴的截距，即：

$$-K_i \times X_1 + Y_1 > -K_i \times X_2 + Y_2 \tag{10}$$

若（10）式成立，则可以推导出下式：

$$K_i > \frac{Y_2 - Y_1}{X_2 - X_1} \tag{11}$$

由于 X_j 始终递增，所以分母不为0，又因为 K_i 是递增的（开头提到）所以有：

$$K_{i+1} > K_i > \frac{Y_2 - Y_1}{X_2 - X_1} \tag{12}$$

$$\Rightarrow -K_{i+1} \times X_1 + Y_1 > -K_{i+1} \times X_2 + Y_2 \tag{13}$$

可以发现由（10）式可以推出（13）式，即（10）为（13）的充分条件，即如果对于i=k，(X_j, Y_j) 所对应的函数值大于 (X_{j+1}, Y_{j+1}) 所对应的函数值，即后者更优时，在所有 $i \geq k$，(X_j, Y_j) 所对应的函数值都大于 (X_{j+1}, Y_{j+1}) 所对应的函数值，即后者都更优。所以通过这个结论，每当计算开始之前，就对凸包的前部进行维护，可以直接通过判断（10）式，确定对凸包的第一个元素的取舍，并重复上述过程直至（10）不再成立。

从上述两点，可以发现维护这个凸包，基本相当于维护一个队列。并且上述的两个过程是针对固定的i中的过程，对于不同的i值，进行同样地循环。而对于方程f(i)的计算，则在单个循环的两次维护之间进行。