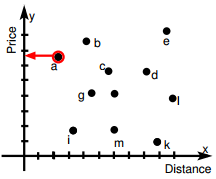
[1] Skyline queries and its variations: An Optimal and Progressive Algorithm for Skyline Queries, D. Papadias, Y. Tao, G.Fu et al., Janpan Sankaranarayanan.

**Skyline Queries**:定义为给定一系列点,…,通过Skyline Queries可以返回一组点P（指skyline 点）。在P中的任意点都不由数据集中的任意点dominated。

**Point domination**:定义为一个点 dominates其它的点当且仅当的坐标在任意方向上都不大于相应的坐标。通俗来说，选择具有较大的转换作为偏好函数，该函数对所有属性而言是单调的。

示例：

* 假设一个数据集包含酒店的所有信息，即到达沙滩的距离以及每个酒店的价格。
* 考虑数据集的二维坐标图，其中X轴表示距离，Y轴表示价格。
* 目标是找到一个酒店，该酒店距离沙滩的距离最近且价格最低（在使用Skyline queries时并不仅限于最小，其它的函数例如最大，联合等都可以使用）。
* 在这一例子中的偏好函数是最小化价格和距离，这个数据集中可能不存在一个单独的点满足这两个条件。
* 用户提供了一组有趣的点，满足了部分所施加的约束。



如左图所示，数据集中的有趣的点为{a,i,k}，a具有到沙滩的最小距离，k具有最低的价格，i既没有最小距离也没有最低价格。i相较于k具有较小的距离，相较于a有较小的价格。因此，i没有被其它任意点dominated。剩余的所有点都由这三个点{a,i,k}所dominated，因为距离或价格都要高于这三个点。

相关算法：

* Convex hulls:包含了Skyline queries的子集；
* Top-K queries：如果偏好函数定义为最小化花费函数，可以采用该算法；
* 与多元优化，最大向量有关的问题。

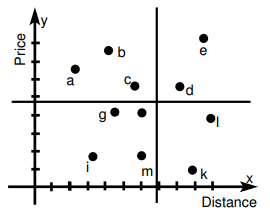
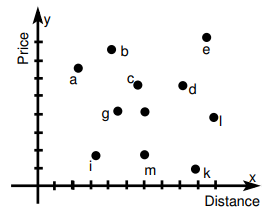
Skyline queries评估技术：

* 环算法；
* 分而治之；
* 平面扫描；
* 最近邻搜索；
* 分支定界Skyline。

环算法：

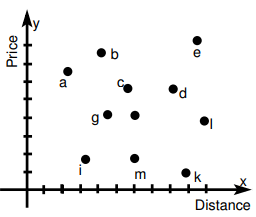
1. 扫描点的列表，测试每个点的优势标准；
2. 包含找到的潜在的skyline点，再将每个访问的点与列表中的元素进行对比，再更新列表；
3. 该方法不需要预先计算索引是独立于空间维度进行的。

分而治之：



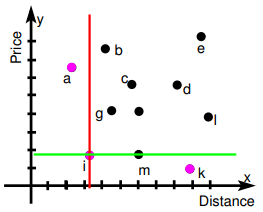
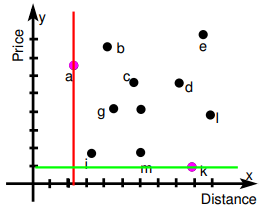
1. 递归地将大型数据集分解为更小的分区。继续，直到数据集的每个较小的分区适合于主存储器中。
2. 用in-memory方法计算每个分区内的部分skyline，将每个分区内的skyline点合并得到最终的skyline query。

平面扫描：

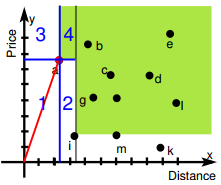
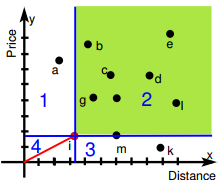
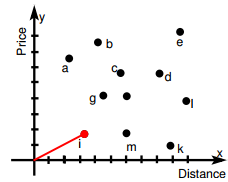
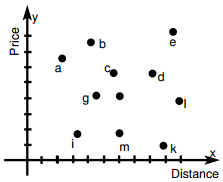
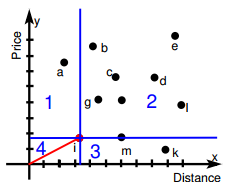


1：平面扫描d维中的每一维；

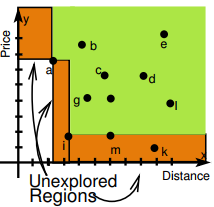
2：找到终止条件。



最近邻搜索：



假设数据点可以使用空间索引结构；使用合适的定义的距离标准，重复使用最近邻算法找到数据点上的skyline queries。数据点中的最近邻居可以看作是在假设测量值之后最接近于原点的数据点。数据点i将整个区域划分为个不相交的区域，可以通过递归找到更多的水平线。

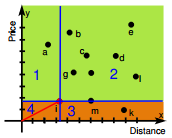
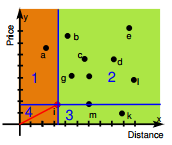


区域2和区域4均没有不需要搜索，剩余的个区域均需要被搜索，没有比节点i更加原点的数据点了，因此在区域i中的任意点都dominate by i。在区域1中重复这一过程，在区域1中的最近邻数据点应该为a。区域4不需要被搜索

重叠搜索区域：

1. 放宽区域限制，使区域之间是不相互重叠的；点将区域减少至2d个区域，而不是划分为个区域；
2. 这样可以花费较少的时间复杂度去搜索整个数据集区域以找到兴趣点。
3. 可以利用任一重复删除技术来恢复被重复删除的数据点。

* Laisser-Faire：维护一个已经嵌入内存的哈希表，将每个存在于哈希表中的数据和flag都进行复制；
* Propagate：当已经找到了数据点p，删除所有未访问点中的与数据点p相关的实例。
* Merge：合并分区以形成非重叠区域。



Skyline queries的变化：

* 排序Skyline queries查询：使用偏好函数来替代最小准测；优先级队列使用偏好函数来计算队列中的元素的最小距离；
* 约束Skyline queries查询：这种Skyline queries仅从约束定义的数据空间返回skyline点。当将对象插入优先级队列时，修剪对象完全位于约束区域之外。
* 枚举查询：对于数据集中的Skyline point，找到数据集中的点的数量。确定Skyline点，定义区域的空间边界，即Skyline占主导地位的点。扫描数据集中的所有点，并针对每一个Skyline点检查其空间范围。点区域交点的总数给出了每个Skyline点所需的计数。
* K-支配查询检索：找到K个点能够最大化dominate数据集中的点。