UNIVERSITY OF TEXAS ARLINGTON

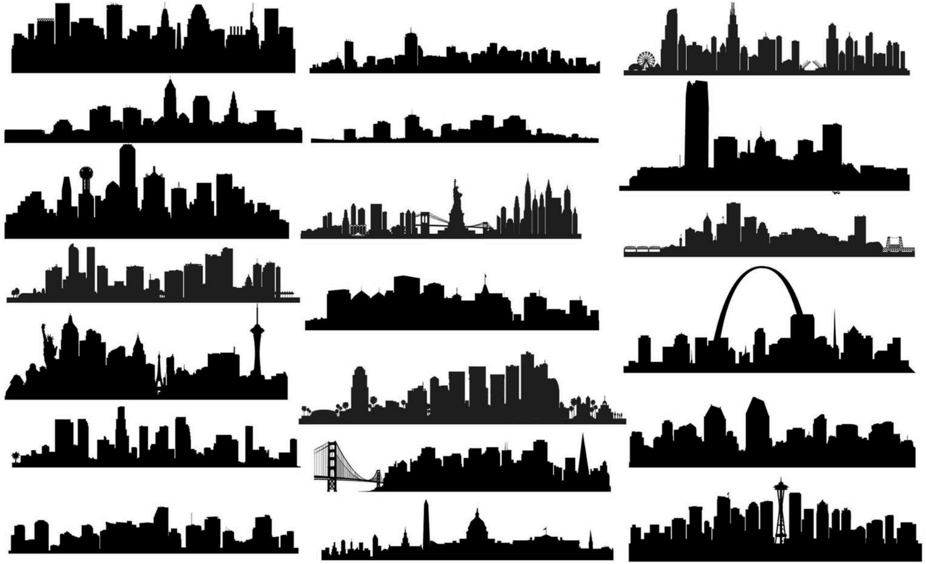
CSE4334/5334 Data Mining

Multi-dimensional Data Analytics: Skyline

Fall 2020

Chengkai Li

What is a skyline?



www.rtkl.com
www.etsy.com

For this lecture, use Google Colab at

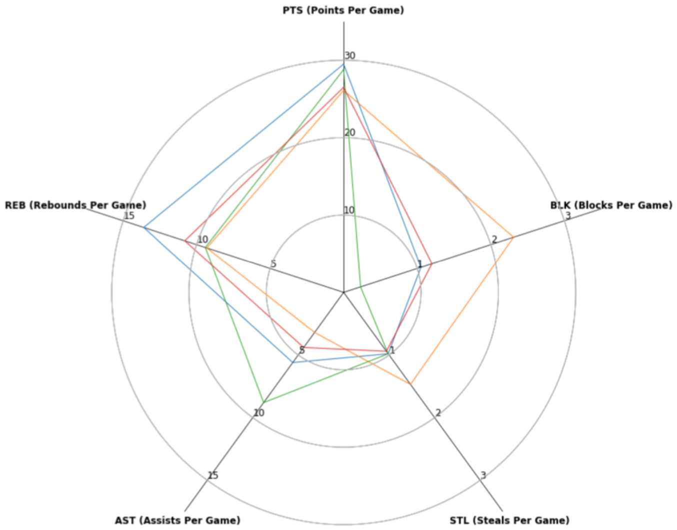
https://colab.research.google.com/drive/1-9OCldZ2YH5iuv1zoA6g6UMb04604QD3

What is a skyline in multi-dimensional data analytics?

	Name	Team	GP (Games Played)	MPG (Minutes Per Game)	PTS (Points Per Game)	REB (Rebounds Per Game)	AST (Assists Per Game)	STL (Steals Per Game)	BLK (Blocks Per Game)	TO (Turnovers Per Game)
0	Steven Adams	OKC	63	26.7	10.9	9.3	2.3	0.81	1.06	1.51
1	Bam Adebayo	MIA	72	33.6	15.9	10.2	5.1	1.14	1.29	2.82
2	LaMarcus Aldridge	SAN	53	33.1	18.9	7.4	2.4	0.68	1.64	1.40
3	Kyle Alexander	MIA	2	6.7	1.0	1.5	0.0	0.00	0.00	0.50
4	Nickeil Alexander-Walker	NOR	47	12.6	5.7	1.8	1.9	0.36	0.17	1.15

Representation of Multi-Dimensional Data Records

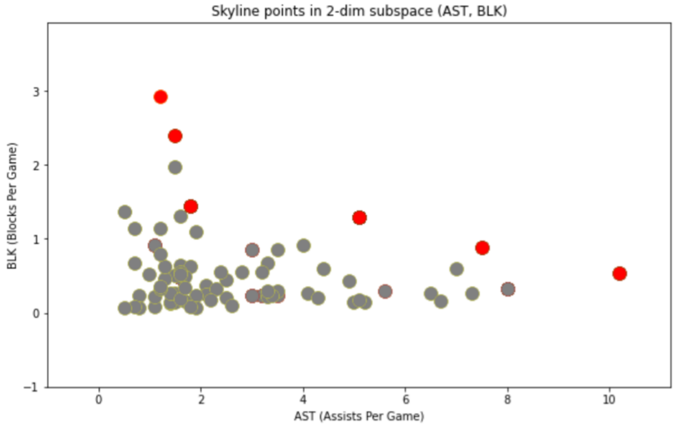
	Name	Team	GP (Games Played)	MPG (Minutes Per Game)	PTS (Points Per Game)	REB (Rebounds Per Game)	AST (Assists Per Game)	STL (Steals Per Game)	BLK (Blocks Per Game)	TO (Turnovers Per Game)
0	Steven Adams	OKC	63	26.7	10.9	9.3	2.3	0.81	1.06	1.51
1	Bam Adebayo	MIA	72	33.6	15.9	10.2	5.1	1.14	1.29	2.82
2	LaMarcus Aldridge	SAN	53	33.1	18.9	7.4	2.4	0.68	1.64	1.40
3	Kyle Alexander	MIA	2	6.7	1.0	1.5	0.0	0.00	0.00	0.50
4	Nickeil Alexander-Walker	NOR	47	12.6	5.7	1.8	1.9	0.36	0.17	1.15



Each data record represented as a polygon

Representation of Multi-Dimensional Data Records

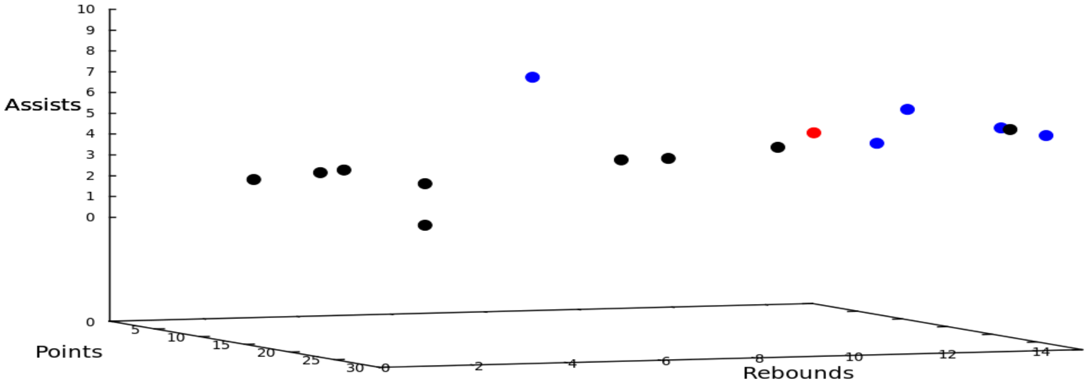
	Name	Team	GP (Games Played)	MPG (Minutes Per Game)	PTS (Points Per Game)	REB (Rebounds Per Game)	AST (Assists Per Game)	STL (Steals Per Game)	BLK (Blocks Per Game)	TO (Turnovers Per Game)
0	Steven Adams	OKC	63	26.7	10.9	9.3	2.3	0.81	1.06	1.51
1	Bam Adebayo	MIA	72	33.6	15.9	10.2	5.1	1.14	1.29	2.82
2	LaMarcus Aldridge	SAN	53	33.1	18.9	7.4	2.4	0.68	1.64	1.40
3	Kyle Alexander	MIA	2	6.7	1.0	1.5	0.0	0.00	0.00	0.50
4	Nickeil Alexander-Walker	NOR	47	12.6	5.7	1.8	1.9	0.36	0.17	1.15



Each data record represented as a point (in a 2-dimensional space in this example)

Representation of Multi-Dimensional Data Records

id	player	day	month	season	team	opp_team	pts	ast	reb
t_1	Bogues	11	Feb.	1991-92	Hornets	Hawks	4	12	5
t_2	Seikaly	13	Feb.	1991-92	Heat	Hawks	24	5	15
t_3	Sherman	7	Dec.	1993-94	Celtics	Nets	13	13	5
t_4	Wesley	4	Feb.	1994-95	Celtics	Nets	2	5	2
t_5	Wesley	5	Feb.	1994-95	Celtics	Timberwolves	3	5	3
t_6	Strictland	3	Jan.	1995-96	Blazers	Celtics	27	18	8
t_7	Wesley	25	Feb.	1995-96	Celtics	Nets	12	13	5



Each data record represented as a point (in a 3-dimensional space in this example)

Skyline

- In a multi-dimensional data space, given a set of data records, the skyline is composed of the subset of data records that are **not dominated by any other data record**.
- Dominance relation: Given two points $p=<p_1, \ldots, p_n>$ and $q = <q_1, \ldots, q_n>$, p dominates q (denoted as $p \succ q$) if and only if (1) p_i "is stronger than or equal to" q_i for any i, and (2) there exists at least one such i that p_i "is stronger than" q_i .
- The notion of a value being "stronger" is application domain dependent: it could mean greater, smaller, cheaper, faster, lower quality, …
- For example, in one application, the definition of dominance relation can be: p dominates q (i.e., $p \succ q$) if and only if (1) $p_i >= q_i$ for any i, and (2) there exists at least one such i that $p_i > q_i$.

Example

Data Record	Attribute A	Attribute B	Attribute C
P1	4	2	3
P2	1	4	2
P3	2	1	3
P4	1	1	3

- First, assuming "the greater, the better" for all attributes
  - Which data records (i.e., data points) dominate which records?
  - Which records are in the skyline?
  - Does a record necessarily belong to the skyline if it dominates some other records?
  - Can a record belong to the skyline even if it doesn't dominate any other record?
  - If a record has the "best" value on some attribute, does it necessarily make the record a skyline point?
- Now, assuming "the smaller, the better" for all attributes. What are the skyline points?

Example

Data Record	Attribute A	Attribute B	Attribute C
P1	4	2	3
P2	1	4	2
P3	2	1	3
P4	1	1	3

- First, assuming "the greater, the better" for all attributes
  - Which data records (i.e., data points) dominate which records? (P1 ≻ P3, P1 ≻ P4, P3 ≻ P4)
  - Which records are in the skyline? (P1, P2)
  - Does a record necessarily belong to the skyline if it dominates some other records? No
  - Can a record belong to the skyline even if it doesn't dominate any other record? Yes
  - If a record has the "best" value on some attribute, does it necessarily make the record a skyline point? No
- Now, assuming "the smaller, the better" for all attributes. What are the skyline points? P2, P4

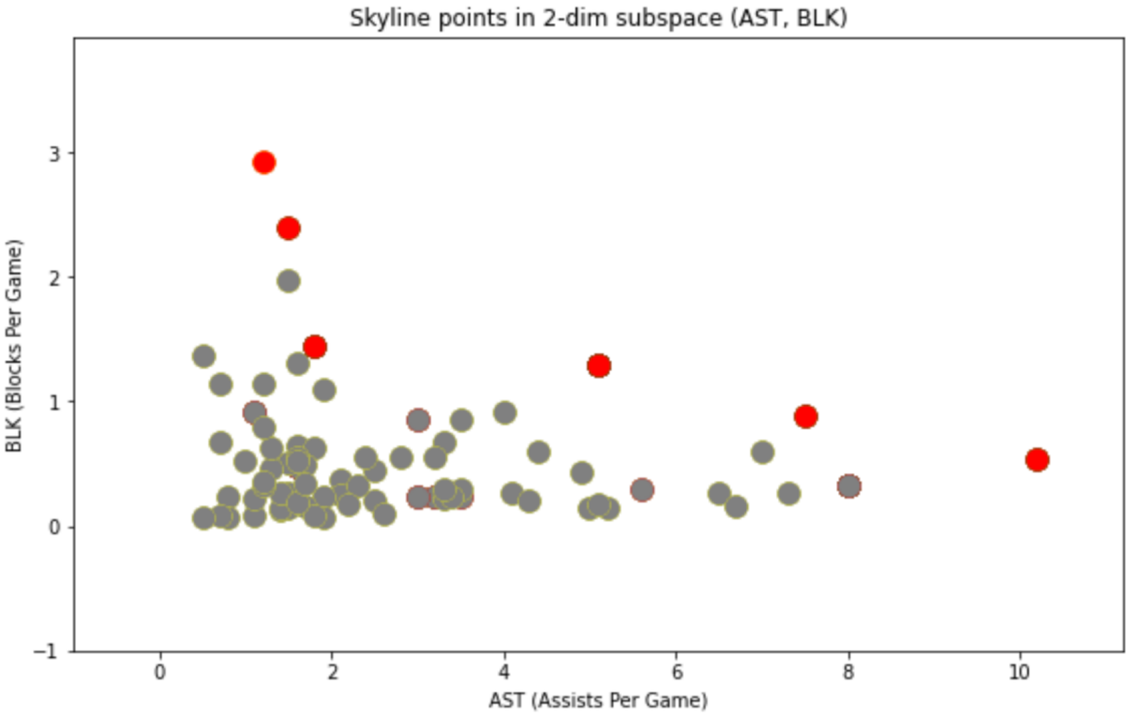
Dominance Test Function

```
def compare1(p, q):
  if len(p) != len(q): return Outcomes.INCOMPARABLE

  p_greater = False
  q_greater = False
  for i in range(len(p)):
    if p[i] < q[i]: q_greater = True
    if p[i] > q[i]: p_greater = True

  if p_greater and not q_greater: return Outcomes.DOMINATE
  if not p_greater and q_greater: return Outcomes.DOMINATED
  return Outcomes.NO_DOMINANCE
```

Which are the skyline points?



Straightforward Algorithm

```python
def straightforward(records):
  skyline = []
  comparisons = 0

  for p in records:
    outcome = None

    for q in records:
      outcome = compare1(p[1:], q[1:])
      comparisons += 1
      if outcome == Outcomes.DOMINATED: break

    if outcome != Outcomes.DOMINATED:
      skyline.append(p)

  print("The algorithm compared " + str(comparisons) + " pairs of data points.")
  return skyline
```

```python
print("There are " + str(len(pts_ast)) + " data points.")
straightforward(pts_ast)
```

```
There are 591 data points.
The algorithm compared 7553 pairs of data points.
[array(['James Harden', 34.3, 7.5], dtype=object),
 array(['LeBron James', 25.3, 10.2], dtype=object),
 array(['Damian Lillard', 30.0, 8.0], dtype=object),
 array(['Trae Young', 29.6, 9.3], dtype=object)]
```

- This straightforward algorithm has a time complexity of $O(n\text{^}2)$, which is apparent from the nested loops. When there are many points to consider, it is inefficient.
- When the straightforward algorithm is applied on the 591 data points in the 2-dimension space (PTS, AST), it took 7,553 comparisons.

A More Efficient Algorithm

This algorithm works by continuously updating the skyline while iterating through all data points. Given each data point `p`, the algorithm compares it with points in the current skyline and updates the skyline as follows:

- If `p` dominates an existing point `q` in the current skyline, `q` is kicked out of the current skyline. In fact, it is discarded forever since we can be sure it cannot be part of the final skyline.
- If `p` is dominated by an existing point `q` in the current skyline, then `p` can be discarded and it is unnecessary to further compare it with the rest of the current skyline points. In fact, it is also unnecessary to further compare it with the rest of the points in the dataset.
- If `p` is not dominated by any existing point in the current skyline, the current skyline is updated to include `p`.

```python
def alg1(records):
  skyline = []
  comparisons = 0

  for p in records:
    updated = []
    outcome = None

    for q in skyline:
      outcome = compare1(p[1:], q[1:])
      comparisons += 1
      if outcome == Outcomes.DOMINATED: break
      if outcome != Outcomes.DOMINATE: updated.append(q)

    if outcome != Outcomes.DOMINATED:
      skyline = updated
      skyline.append(p)

  print("The algorithm compared " + str(comparisons) + " pairs of data points.")
  return skyline
```

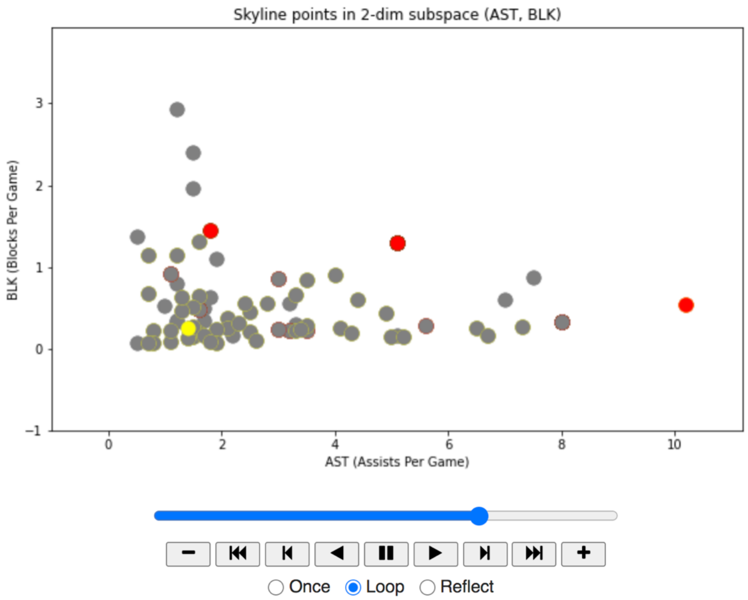
A More Efficient Algorithm

```
alg1(pts_ast)

The algorithm compared 620 pairs of data points.
[array(['James Harden', 34.3, 7.5], dtype=object),
 array(['LeBron James', 25.3, 10.2], dtype=object),
 array(['Damian Lillard', 30.0, 8.0], dtype=object),
 array(['Trae Young', 29.6, 9.3], dtype=object)]
```

- This algorithm only required 620 comparisons, in contrast with the 7,553 comparisons taken by the straightforward algorithm.
- Its worst-case time-complexity is no better than the straightforward algorithm. It could end up comparing every pair of data points. (Question: in what scenario such worst-case will happen?) In practice, though, it is very efficient.

Visualization of the Algorithm Execution



- Visualization at https://colab.research.google.com/drive/1-9OCldZ2YH5iuv1zoA6g6UMb04604QD3
- The number of red points, i.e., the size of the skyline that is being continuously updated, has always been about a handful. Let's denote by s the maximal size of the skyline during algorithm execution. If the number of data points is n, there have been at most $s \times n$ comparisons. Since s in practice is small, this algorithm is much more efficient than the $O(n^2)$ straightforward algorithm.

The algorithm compared 600 pairs of data points.

There are 3 skyline points in subspace ['PTS (Points Per Game)', 'REB (Rebounds Per Game)']

The algorithm compared 670 pairs of data points.

There are 8 skyline points in subspace ['PTS (Points Per Game)', 'REB (Rebounds Per Game)', 'AST (Assists Per Game)']

The algorithm compared 1084 pairs of data points.

There are 16 skyline points in subspace ['PTS (Points Per Game)', 'REB (Rebounds Per Game)', 'AST (Assists Per Game)', 'STL (Steals Per Game)']

The algorithm compared 1691 pairs of data points.

There are 29 skyline points in subspace ['PTS (Points Per Game)', 'REB (Rebounds Per Game)', 'AST (Assists Per Game)', 'STL (Steals Per Game)', 'BLK (Blocks Per Game)']

The algorithm compared 58548 pairs of data points.

There are 262 skyline points in subspace ['PTS (Points Per Game)', 'REB (Rebounds Per Game)', 'AST (Assists Per Game)', 'STL (Steals Per Game)', 'BLK (Blocks Per Game)', 'TO (Turnovers Per Game)']

We start with the 2-dim space (PTS, REB) and include other dimensions one by one. What do you observe from the execution results? The larger the dimensionality, the more skyline points and the more comparisons required to find the skyline points. Usually the purpose of skyline analysis is to locate a small number of extraordinary records. A large skyline is not very useful in practice. In other words, skyline analysis is less useful in a subspace with high dimensionality.

Particularly, including the negative performance category TO (Turnovers Per Game) drastically increased the size of the skyline. Why? It is because this dimension forms clear anti-correlation with other performance categories: It is difficult to achieve outstanding performance stats, which oftentimes means a player needs to log a lot of minutes, while maintaining a small number of turnovers.

Discussion

How is skyline analysis related to and different from OLAP/data cube?