# OPTIMAL SETS OF PROJECTIONS OF HIGH-DIMENSIONAL DATA

DIRK J. LEHMANN, HOLGER THEISEL. TVCG 22(1) 2016
PRESENTED BY JASON HARTFORD

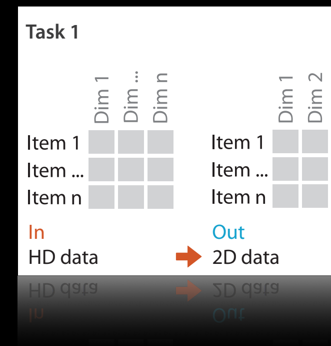
## WHAT?

Recall **REDUCE** task:

- **In:** HD Data
- **Out:** 2D projection

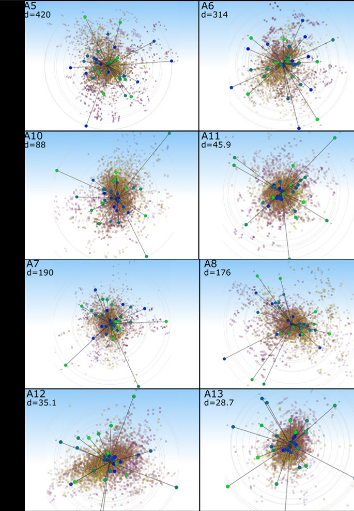
Today's paper:

- **In:** HD Data
- **Out:** "optimal" set of **2D projections**




## WHY?

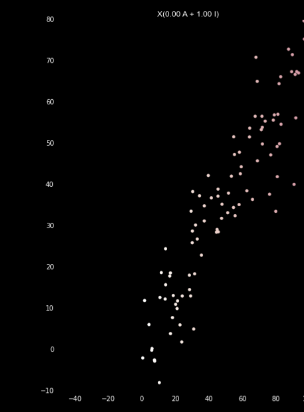
- Large space of potential projections
- Would like to find a minimal set of "interesting" projections to describe our dataset



## HOW?

**Core assumption:**

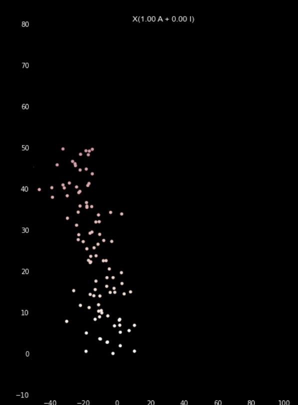
- Assume projections only provide **insight** if they're not equivalent up to an **affine** map.



## HOW?

**Core assumption:**

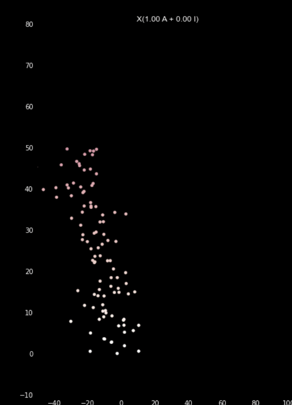
- Assume projections only provide **insight** if they're not equivalent up to an **affine** map.



## HOW?

**Core assumption:**

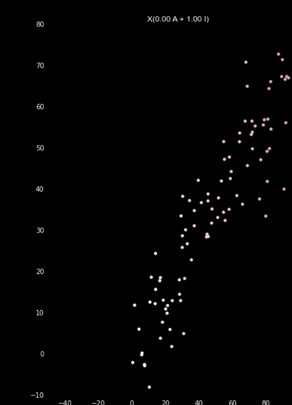
- Assume projections only provide **insight** if they're not equivalent up to an **affine** map.



## HOW?

**Core assumption:**

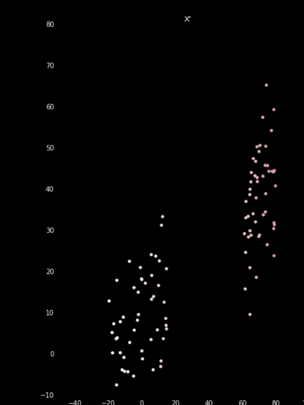
- Assume projections only provide **insight** if they're not equivalent up to an **affine** map.

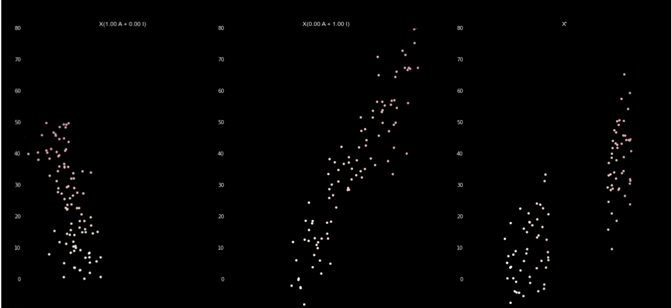


## HOW?

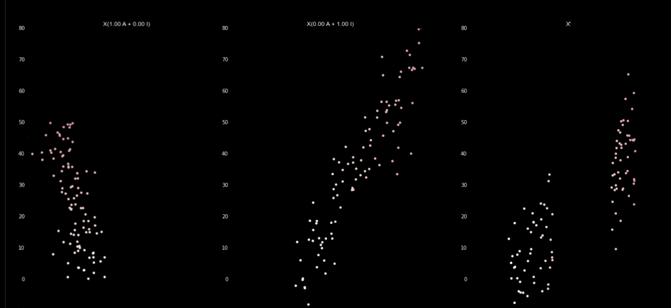
**Core assumption:**

- Assume projections only provide **insight** if they're not equivalent up to an **affine** map.

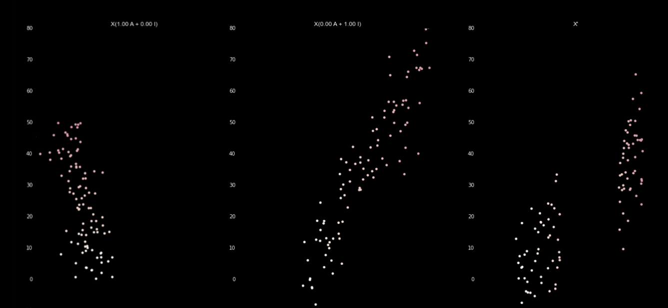




$A_1 \approx A_2 \neq A_3$



$A_1 \approx A_2 \neq A_3$



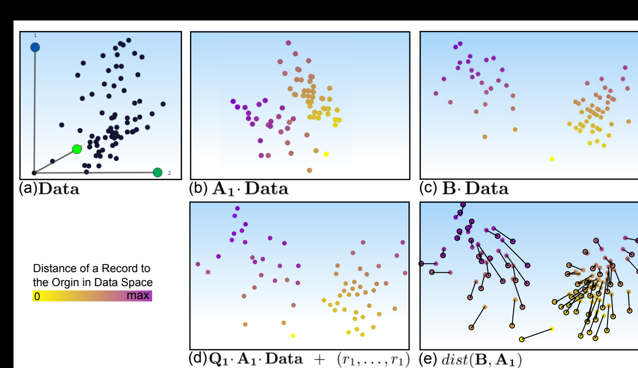
$d(A_1, A_2)=0$ $d(A_1, A_3)>0$

## ALGORITHM - HIGH LEVEL

- At iteration i, given set of projections $\mathbf{A} = \{A_0, \ldots, A_{i-1}\}$
- Greedily find linear projection B that is most dissimilar from the projections in **A**
- Add $A_i$ = B to our set of projections
- Repeat until the best new projection gives no new insight (equivalent up to an affine transformation)

The devil's in the details….

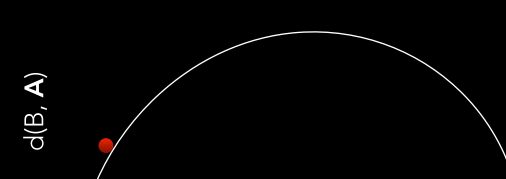
## MEASURING DISSIMILARITY




(a) Data (b) $A_1 \cdot$ Data (c) $B \cdot$ Data (d) $Q_1 \cdot A_1 \cdot$ Data + $(r_1, \ldots, r_1)$ (e) $dist(B, A_1)$

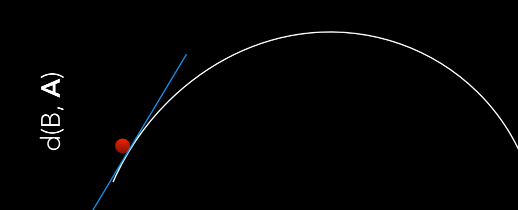
## FINDING THE "MOST DISSIMILAR" PROJECTION

- Given $\mathbf{A} = \{A_0, \ldots, A_{i-1}\}$ start by setting $B = A_{i-1}$.
- Apply gradient ascent to increase the dissimilarity
- Stop when B converges and it to **A**



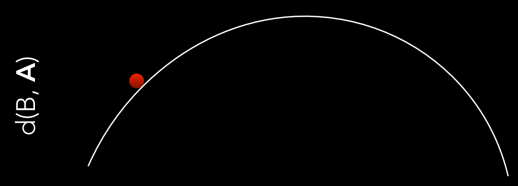

## FINDING THE "MOST DISSIMILAR" PROJECTION

- Given $\mathbf{A} = \{A_0, \ldots, A_{i-1}\}$ start by setting $B = A_{i-1}$.
- Apply gradient ascent to increase the dissimilarity
- Stop when B converges and it to **A**



## FINDING THE "MOST DISSIMILAR" PROJECTION

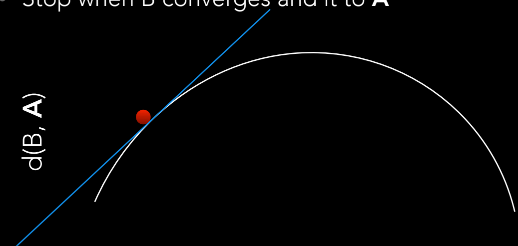
- Given **A** = $\{A_0, \ldots, A_{i-1}\}$ start by setting $B = A_{i-1}$.
- Apply gradient ascent to increase the dissimilarity
- Stop when B converges and it to **A**





## FINDING THE "MOST DISSIMILAR" PROJECTION

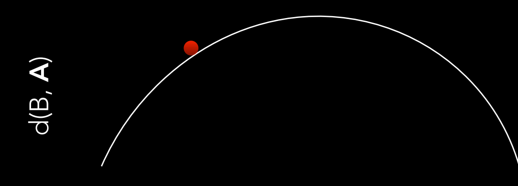
- Given **A** = $\{A_0, \ldots, A_{i-1}\}$ start by setting $B = A_{i-1}$.
- Apply gradient ascent to increase the dissimilarity
- Stop when B converges and it to **A**



## FINDING THE "MOST DISSIMILAR" PROJECTION

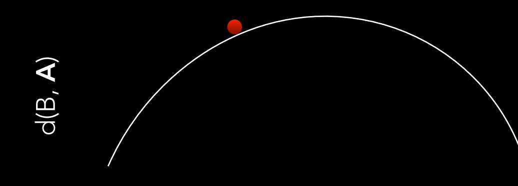
- Given **A** = $\{A_0, \ldots, A_{i-1}\}$ start by setting $B = A_{i-1}$.
- Apply gradient ascent to increase the dissimilarity
- Stop when B converges and it to **A**



## FINDING THE "MOST DISSIMILAR" PROJECTION

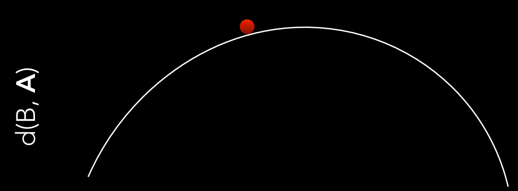
- Given **A** = $\{A_0, \ldots, A_{i-1}\}$ start by setting $B = A_{i-1}$.
- Apply gradient ascent to increase the dissimilarity
- Stop when B converges and it to **A**



## FINDING THE "MOST DISSIMILAR" PROJECTION

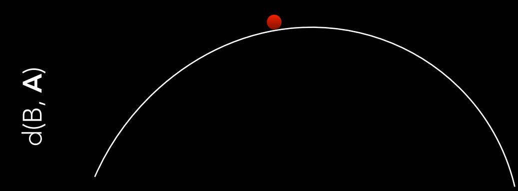
- Given **A** = $\{A_0, \ldots, A_{i-1}\}$ start by setting $B = A_{i-1}$.
- Apply gradient ascent to increase the dissimilarity
- Stop when B converges and it to **A**



## FINDING THE "MOST DISSIMILAR" PROJECTION

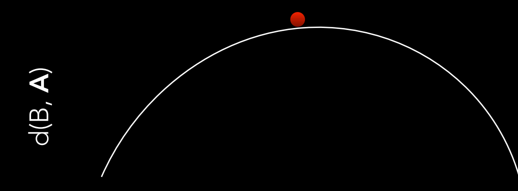
- Given **A** = $\{A_0, \ldots, A_{i-1}\}$ start by setting $B = A_{i-1}$.
- Apply gradient ascent to increase the dissimilarity
- Stop when B converges and it to **A**



## FINDING THE "MOST DISSIMILAR" PROJECTION

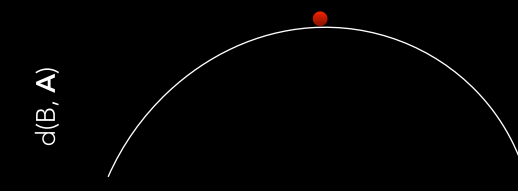
- Given **A** = $\{A_0, \ldots, A_{i-1}\}$ start by setting $B = A_{i-1}$.
- Apply gradient ascent to increase the dissimilarity
- Stop when B converges and it to **A**



## FINDING THE "MOST DISSIMILAR" PROJECTION

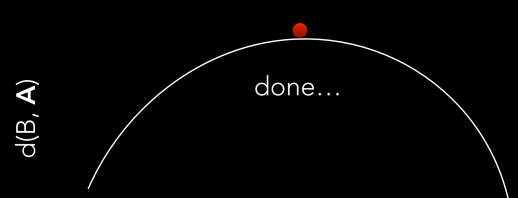
- Given **A** = $\{A_0, \ldots, A_{i-1}\}$ start by setting $B = A_{i-1}$.
- Apply gradient ascent to increase the dissimilarity
- Stop when B converges and it to **A**



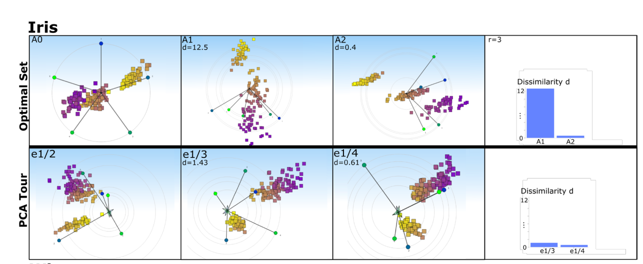
## FINDING THE "MOST DISSIMILAR" PROJECTION

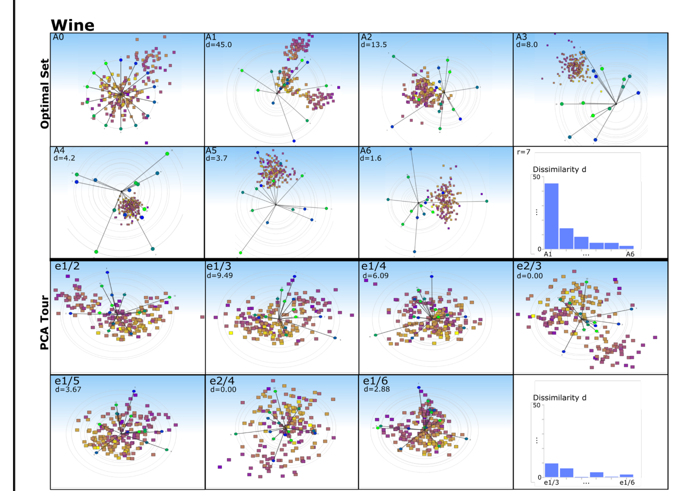
- Given **A** = $\{A_0, \ldots, A_{i-1}\}$ start by setting $B = A_{i-1}$.
- Apply gradient ascent to increase the dissimilarity
- Stop when B converges and it to **A**

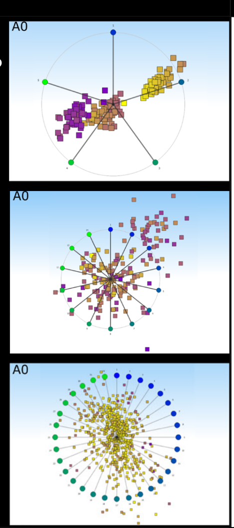

## TERMINATING THE ALGORITHM

- Terminate when $d(B, A_0, \ldots, A_{i-1}) = 0$.
- i.e. We have a complete set of linear projects up to affine transforms.
- This occurs after at most n/2 projections.

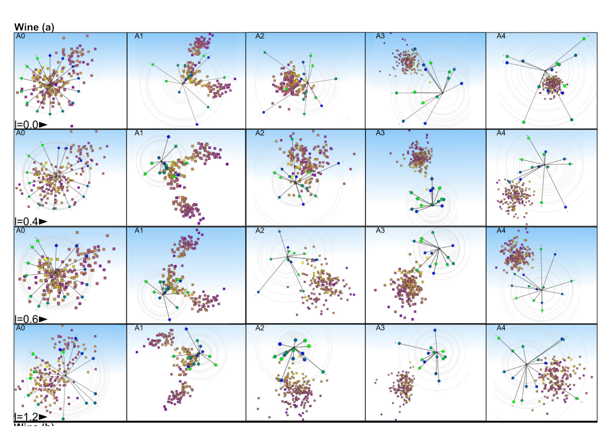



## HOW DO WE CHOOSE $\{A_0\}$?




- Default choice: radial layout.
- Stable to alternative choices - the data patterns remain visible even if the projections change.




## SUMMARY

- The algorithm produces the optimal set of *linear* projections up to affine transforms.
- Produces < n/2 independent projections.
- Relatively robust to initialisation and convergence parameters.
- Scalability could be an issue? Distance is expensive.
- Needs testing to see if the affine assumption reasonable