A Near-linear Time Approximation Algorithm for Angle-based Outlier Detection in High-dimensional Data

Ninh Pham
IT University of Copenhagen
ndap@itu.dk

Rasmus Pagh
IT University of Copenhagen
pagh@itu.dk

Outline

- Introduction
- Proposed Approach
- Experiments
- Conclusion

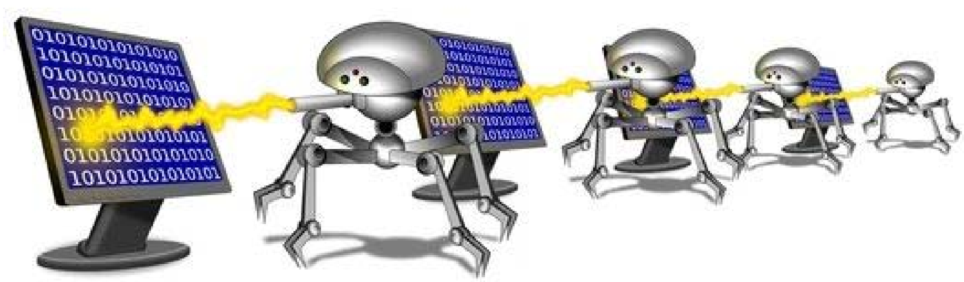
What is an outlier?

- Definition

  An object deviates significantly from normal objects.

- Applications

  - Credit Card Fraud **- Network Intrusion**



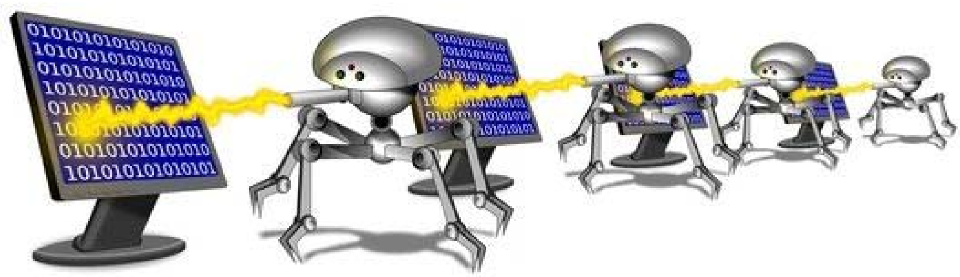
What is an outlier?

- Definition

  An object deviates significantly from normal objects.

- Applications

  - Credit Card Fraud **- Network Intrusion**



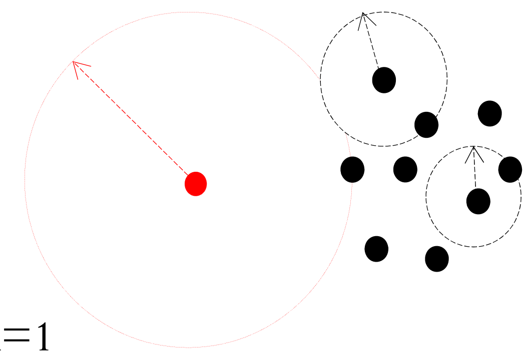
Outlier factor

- Measure the degree of outlier-ness

Outlier factor

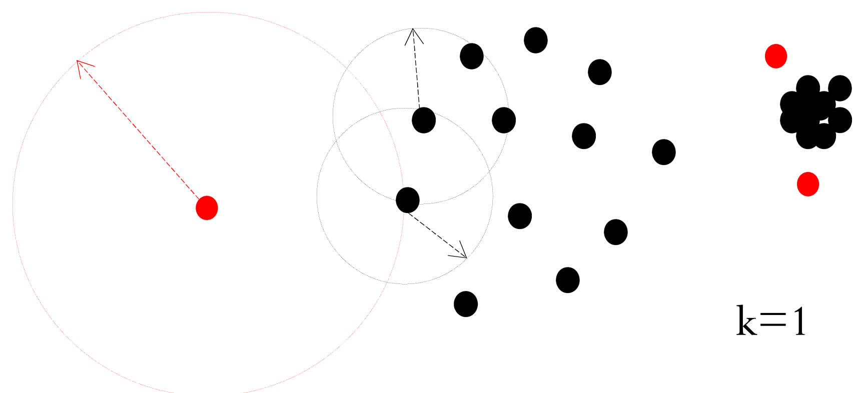
- Measure the degree of outlier-ness
- Traditional outlier factors

- kNN distance [RRS'00]



k=1

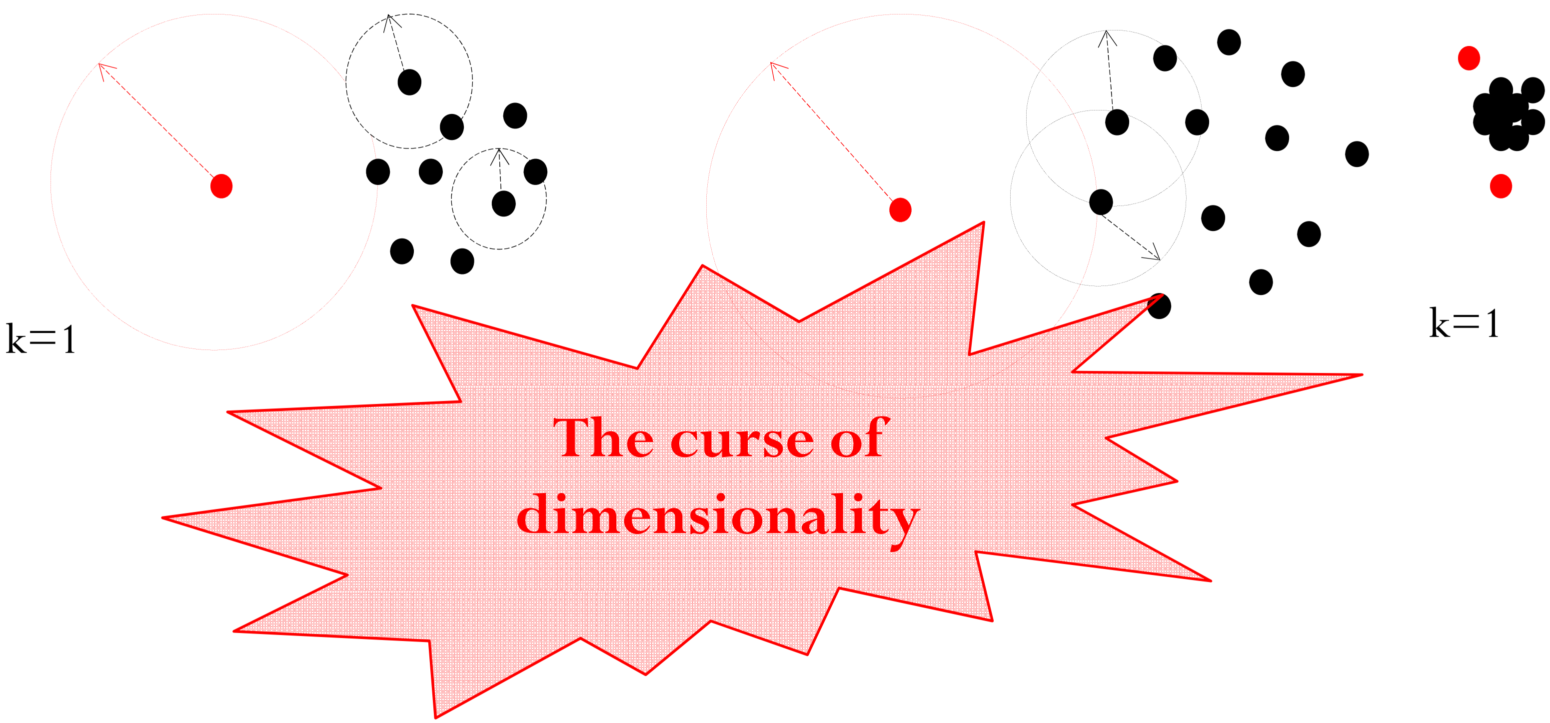
- Local density [BKNS'00]



k=1

Outlier factor

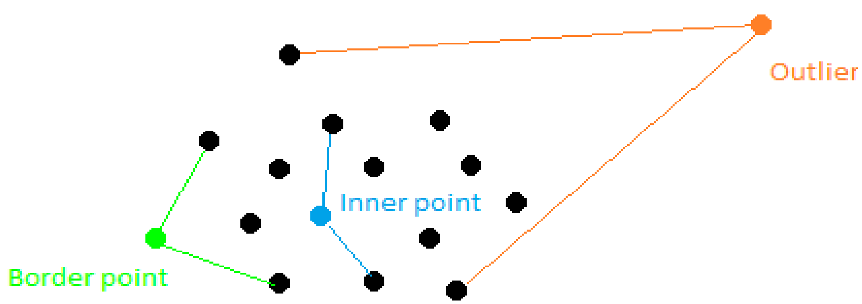
- Measure the degree of outlier-ness
- Traditional outlier factors
  - **kNN distance** [RRS'00]
  - **Local density** [BKNS'00]

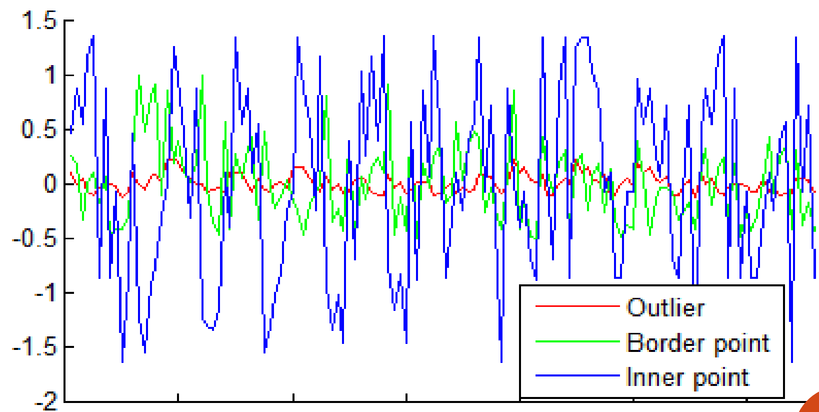


Outlier factor

- Measure the degree of outlier-ness
- Traditional outlier factors

  - kNN distance [RRS'00] **- Local density** [BKNS'00]





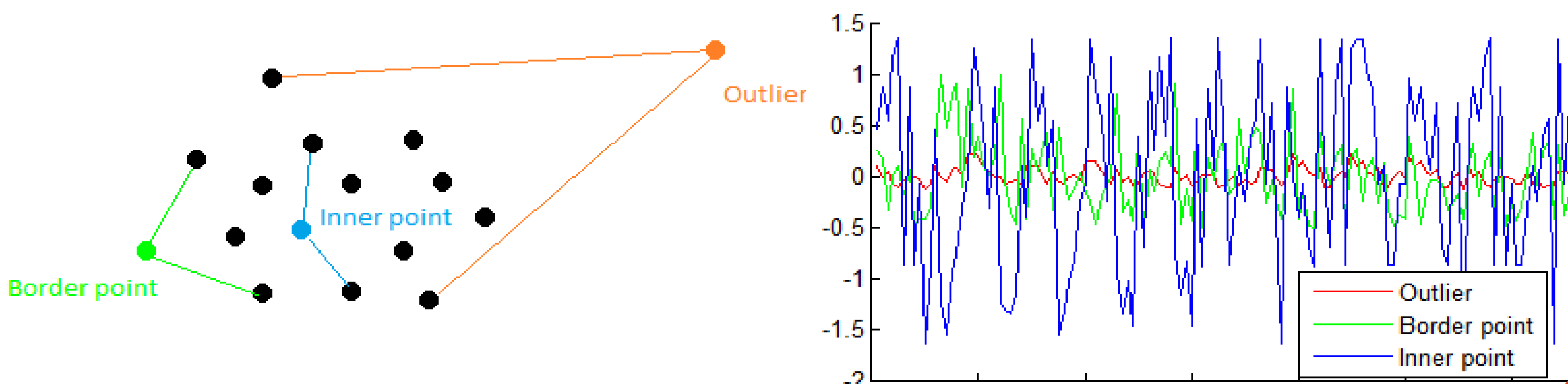
The variance of angles for different kinds of points

Outlier factor

- Measure the degree of outlier-ness
- Traditional outlier factors

  - kNN distance [RRS'00] **- Local density** [BKNS'00]
- **Angle-based outlier factor** [KSZ'08]

  The smaller the variance of angle between a point to other pairs of points is, the more likely it is an outlier.



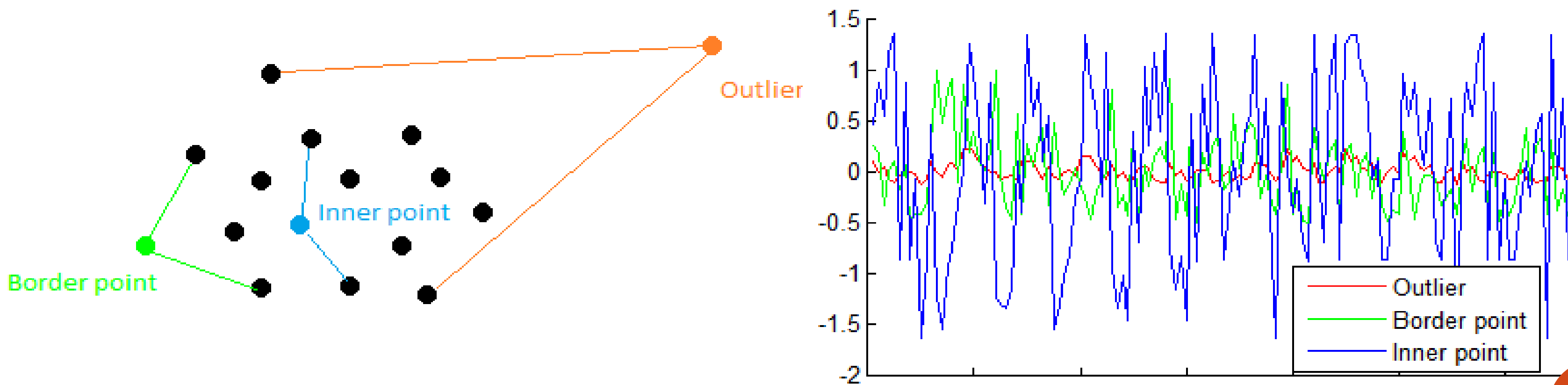
The variance of angles for different kinds of points

Angle-based Outlier Factor

- Outlier factor of point p:

$$VOA(p) = \mathrm{Var}[\Theta_{apb}] = MOA_2(\Theta_{apb}) - MOA_1^2(\Theta_{apb})$$

where Θ_{apb} is a random angle between p and random pair (a, b)



The variance of angles for different kinds of points

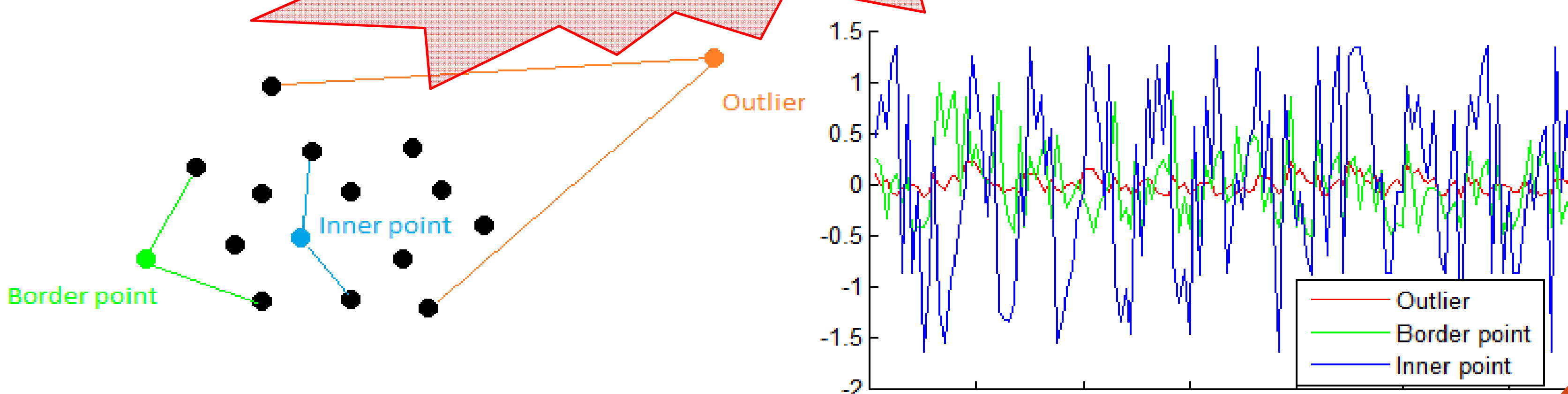
Angle-based Outlier Factor

- Outlier factor of point p:

$$VOA(p) = \mathrm{Var}[\Theta_{apb}] = MOA_2(\Theta_{apb}) - MOA_1^2(\Theta_{apb})$$

where Θ_{apb} is a random angle between p and random pair (a, b)

Naïve approach runs in $O(n^3)$

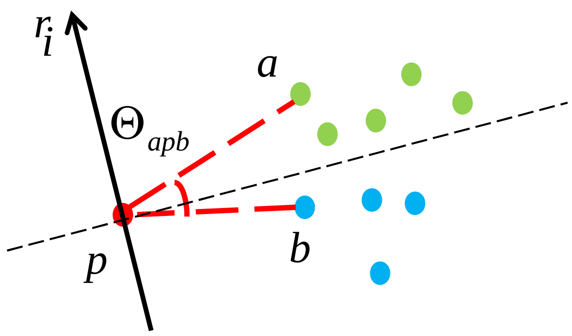


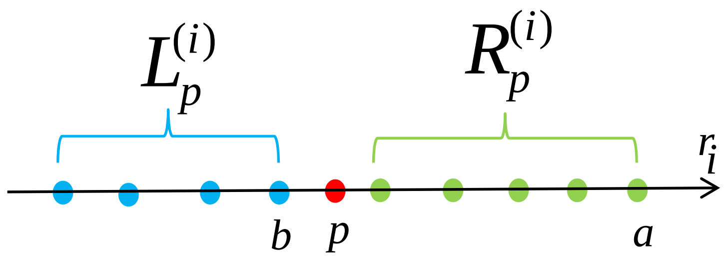
The variance of angles for different kinds of points

Near-linear Time Approximation for VOA

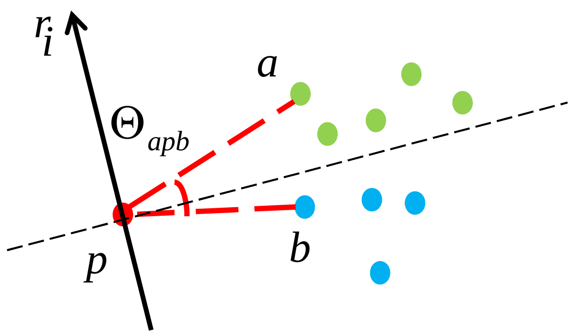
- Project the data set on random hyperplanes
- Sort the data set by their inner products.
- Apply efficient approaches (AMS Sketch) to approximate variance of angle (VOA)

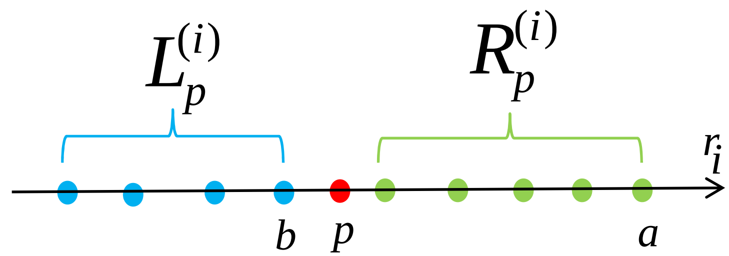
Random Hyperplane Projection





Random Hyperplane Projection

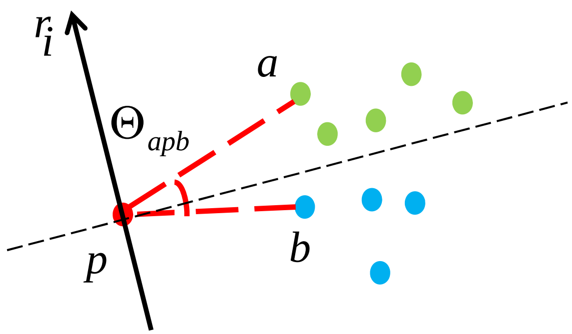


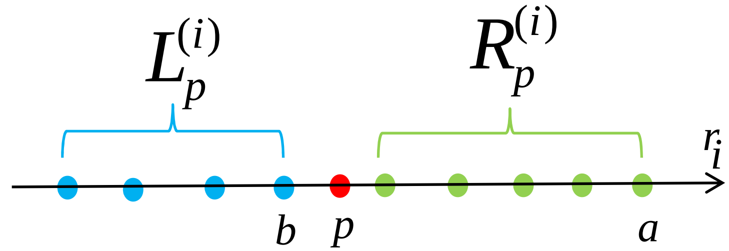


$$X_{apb}^{(i)} = \begin{cases} 1 & \text{if } a \cdot r_i < p \cdot r_i < b \cdot r_i \\ 0 & \text{otherwise} \end{cases} \Rightarrow \boxed{\mathbf{E}[X_{apb}^{(i)}] = \frac{\Theta_{apb}}{2\pi}}$$

[Goemans & Williamson'95]

Random Hyperplane Projection





$u_i = \langle 1,1,1,1,0,0,0,0,0 \rangle \qquad v_i = \langle 0,0,0,0,1,1,1,1,1 \rangle$

$$X_{apb}^{(i)} = \begin{cases} 1 & \text{if } a \cdot r_i < p \cdot r_i < b \cdot r_i \\ 0 & \text{otherwise} \end{cases} \Rightarrow \boxed{\mathbf{E}[X_{apb}^{(i)}] = \frac{\Theta_{apb}}{2\pi}}$$

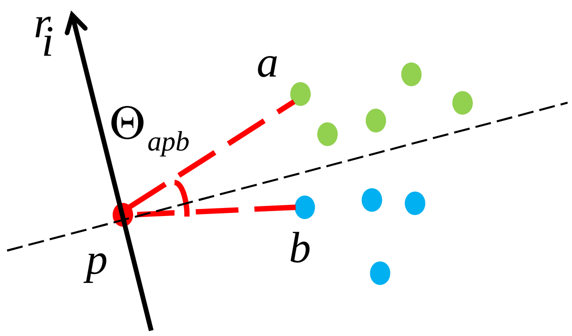
[Goemans & Williamson'95]

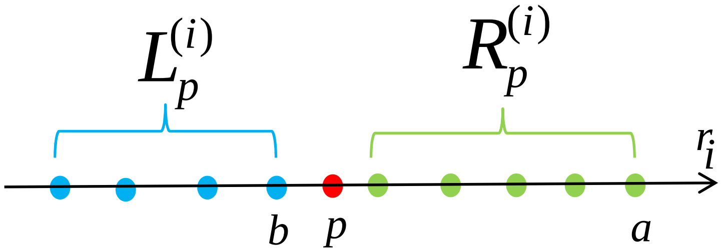
$$P = \sum_{i=1}^{t} (u_i \otimes v_i)$$

$$\boxed{\Theta_{apb}^2 = \frac{(2\pi)^2}{t(t-1)} \left(\mathbf{E}[P_{ij}^2] - \frac{t}{2\pi} \Theta_{apb} \right)}$$

P_{ij} is the number of times that a locates on the left side and b locates on the right side after t projections

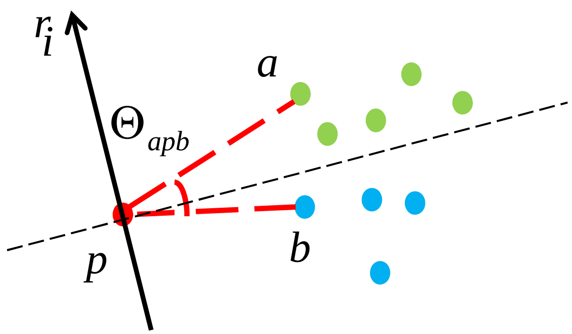
First moment estimation

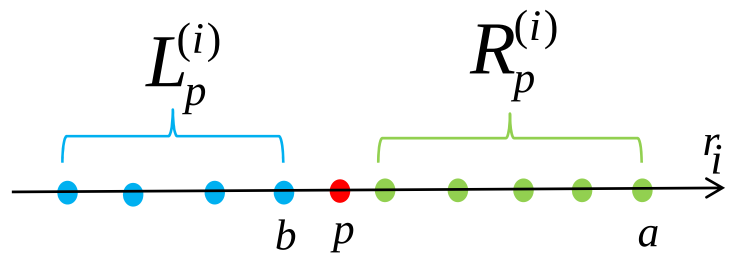




$$X_{apb}^{(i)} = \begin{cases} 1 & \text{if } a \cdot r_i < p \cdot r_i < b \cdot r_i \\ 0 & \text{otherwise} \end{cases} \Rightarrow \boxed{\mathbf{E}[X_{apb}^{(i)}] = \frac{\Theta_{apb}}{2\pi}}$$

First moment estimation

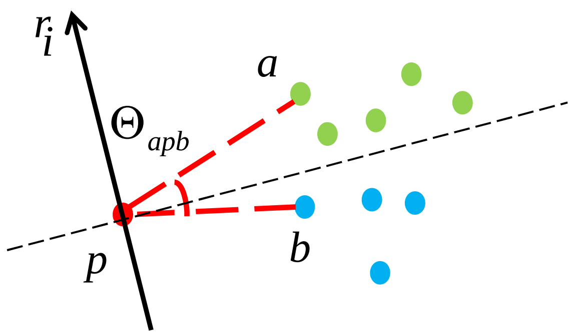


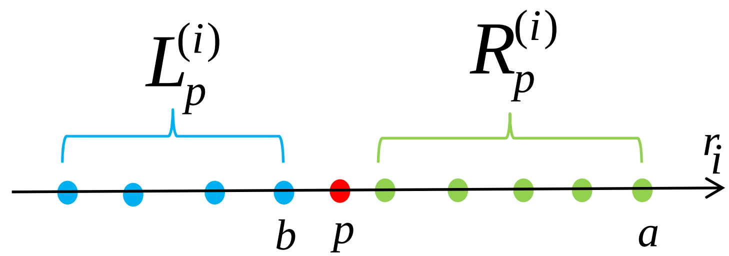


$$X_{apb}^{(i)} = \begin{cases} 1 & \text{if } a \cdot r_i < p \cdot r_i < b \cdot r_i \\ 0 & \text{otherwise} \end{cases} \Rightarrow \boxed{\mathbf{E}[X_{apb}^{(i)}] = \frac{\Theta_{apb}}{2\pi}}$$

$$MOA_1(p) = \frac{4\pi}{(n-1)(n-2)} \sum_{\substack{a,b \in S \setminus \{p\} \\ a \neq b}} \mathbf{E}[X_{apb}^{(i)}]$$

First moment estimation





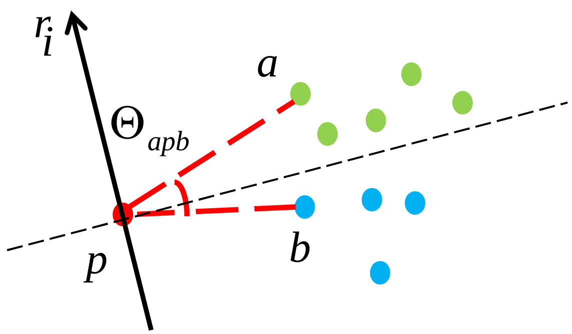
$$X_{apb}^{(i)} = \begin{cases} 1 & \text{if } a \cdot r_i < p \cdot r_i < b \cdot r_i \\ 0 & \text{otherwise} \end{cases} \Rightarrow \boxed{\mathbf{E}[X_{apb}^{(i)}] = \frac{\Theta_{apb}}{2\pi}}$$

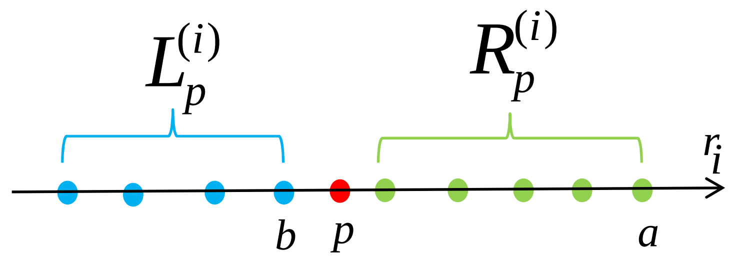
$$MOA_1(p) = \frac{4\pi}{(n-1)(n-2)} \sum_{\substack{a,b \in S \setminus \{p\} \\ a \neq b}} \mathbf{E}[X_{apb}^{(i)}]$$

Unbiased estimator

$$\boxed{F_1(p) = \frac{2}{(n-1)(n-2)} \left| L_p^{(i)} \right| \left| R_p^{(i)} \right|}$$

AMS Sketch





$$u_i = \langle 1,1,1,1,0,0,0,0,0 \rangle \qquad v_i = \langle 0,0,0,0,1,1,1,1,1 \rangle$$

$$P = \sum_{i=1}^{t} (u_i \otimes v_i)$$

$$AMS(L_p^{(i)}) = AMS(u_i) = s_1 \cdot u_i$$

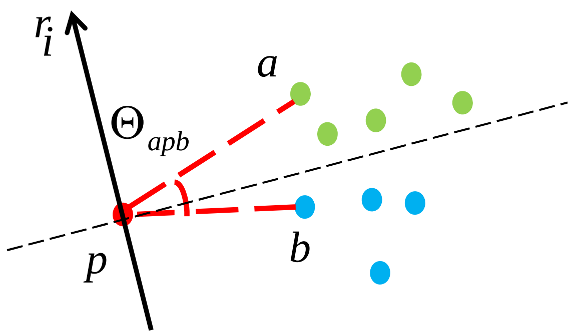
$$AMS(R_p^{(i)}) = AMS(v_i) = s_2 \cdot v_i$$

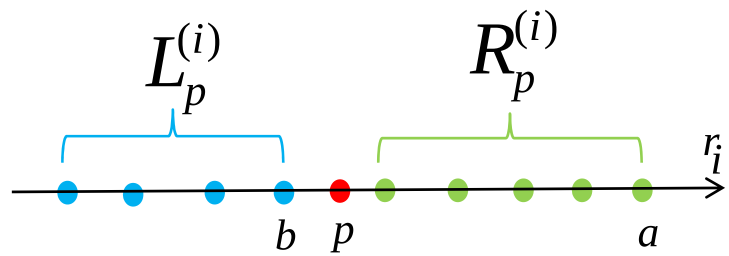
where s_1 and s_2 are two different 4-wise independent vectors in $\{\pm 1\}^{n-1}$

$$\|P\|_F^2 = \mathbf{E}\left[\left(\sum_{i=1}^{t} AMS(L_p^{(i)}) AMS(R_p^{(i)})\right)^2\right]$$

[Indyk & McGregor'08]

Second moment estimation



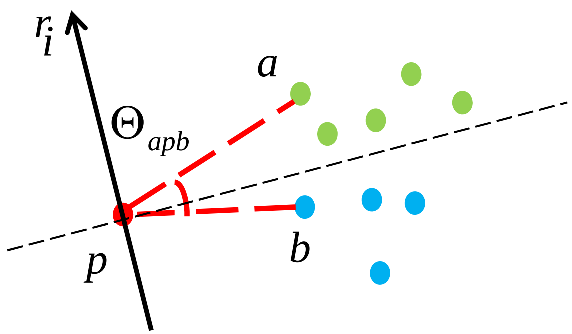


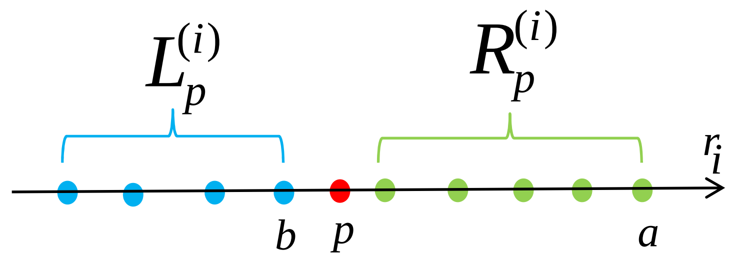
$$u_i = \langle 1,1,1,1,0,0,0,0,0 \rangle \qquad v_i = \langle 0,0,0,0,1,1,1,1,1 \rangle$$

$$AMS(L_p^{(i)}) = AMS(u_i) \qquad AMS(R_p^{(i)}) = AMS(v_i)$$

$$MOA_2(p) = \frac{4\pi^2}{t(t-1)(n-1)(n-2)} \mathbf{E}\left[\|P\|_F^2\right] - \frac{2\pi}{t-1} MOA_1(p)$$

Second moment estimation





$$u_i = \langle 1,1,1,1,0,0,0,0,0 \rangle \qquad v_i = \langle 0,0,0,0,1,1,1,1,1 \rangle$$

$$AMS(L_p^{(i)}) = AMS(u_i) \qquad AMS(R_p^{(i)}) = AMS(v_i)$$

$$MOA_2(p) = \frac{4\pi^2}{t(t-1)(n-1)(n-2)} \mathbf{E}\left[\|P\|_F^2\right] - \frac{2\pi}{t-1} MOA_1(p)$$

$$F_2'(p) = \frac{4\pi^2}{t(t-1)(n-1)(n-2)} \|P\|_F^2 - \frac{2\pi}{t-1} F_1(p)$$

Unbiased estimator

$$F_2(p) = \frac{4\pi^2}{t(t-1)(n-1)(n-2)} \left(\sum_{i=1}^{t} AMS(L_p^{(i)}) AMS(R_p^{(i)}) \right)^2 - \frac{2\pi}{t-1} F_1(p)$$

Algorithm Overview

Algorithm 1 FastVOA(S, t, s_1, s_2)

Ensure: Return the variance estimator for all points

1: $\mathcal{L} \leftarrow RandomProjection(S, t)$
2: $F1 \leftarrow FirstMomentEstimator(\mathcal{L}, t, n)$
3: **for** $i = 1 \rightarrow s_2$ **do**
4: $\quad \mathbf{Y}_i \leftarrow \sum_{j=1}^{s_1} (FrobeniusNorm(\mathcal{L}, t, n))^2 / s_1$
5: **end for**
6: $F2 \leftarrow median\{\mathbf{Y}_1, \cdots, \mathbf{Y}_{s_2}\}$
7: $\mathbf{Var} \leftarrow [0]^n$
8: **for** $j = 1 \rightarrow n$ **do**
9: $\quad F2[j] = \frac{4\pi^2}{t(t-1)(n-1)(n-2)} F2[j] - \frac{2\pi F1[j]}{t-1}$
10: $\quad \mathbf{Var}[j] = F2[j] - (F1[j])^2$
11: **end for**
12: **return** $\mathbf{Var}$

FastVOA runs in $O(tn(d+\log n+s_1 s_2))$ time.

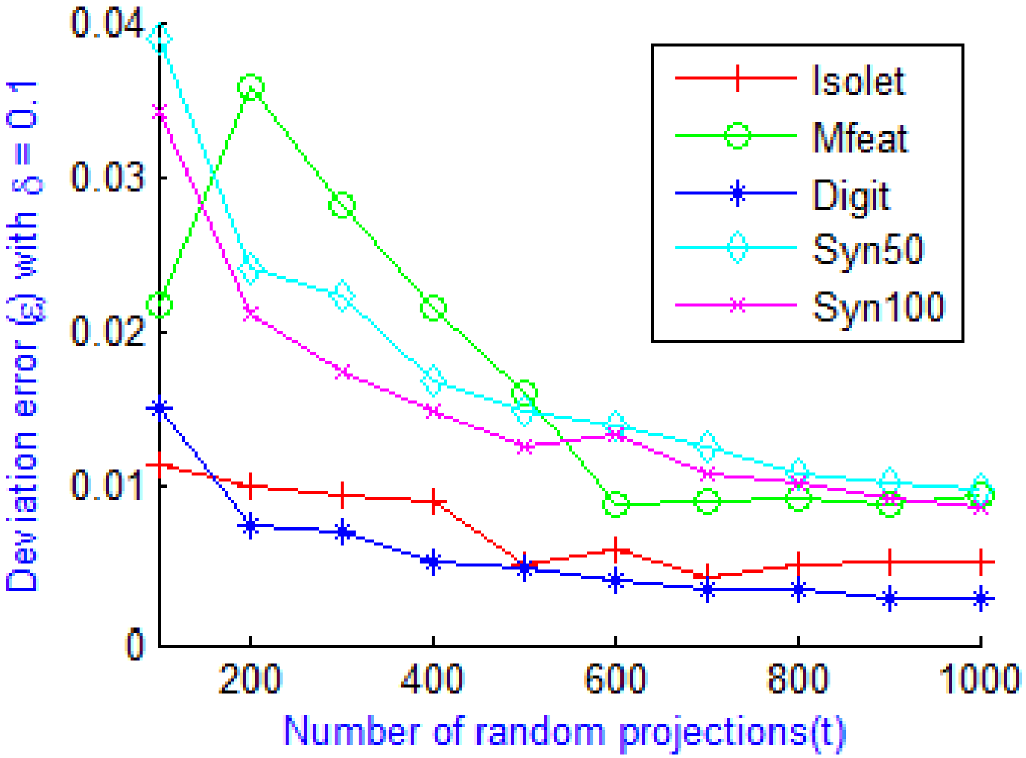
Error Analysis

- For every $\delta > 0$ and $\varepsilon > 0$, the number of random projections $t = O(\varepsilon^{-2} \log n)$, the AMS sketch size $s_1 = O(\varepsilon^{-2})$, and $s_2 = O(\log \delta^{-1})$, the probability that an unbiased estimator of VOA of a point deviates from its expectation by at most $O(\varepsilon)$ is $1 - O(n^{-2})$.

Experiments

- **Accuracy:**

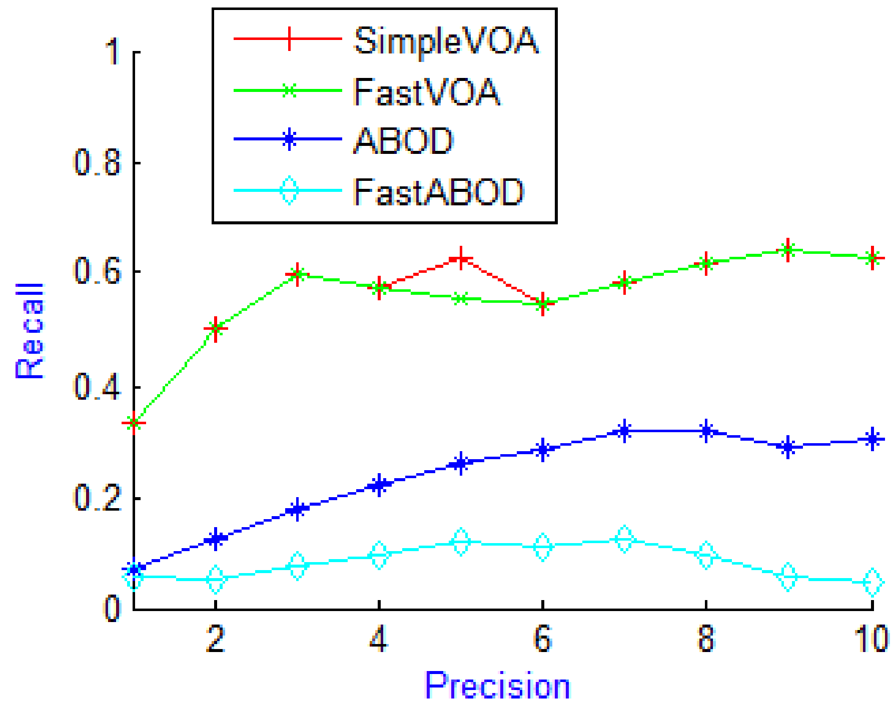
  The deviation error from its expectation of unbiased variance estimators on 5 data sets



Experiments

- **Effectiveness:**

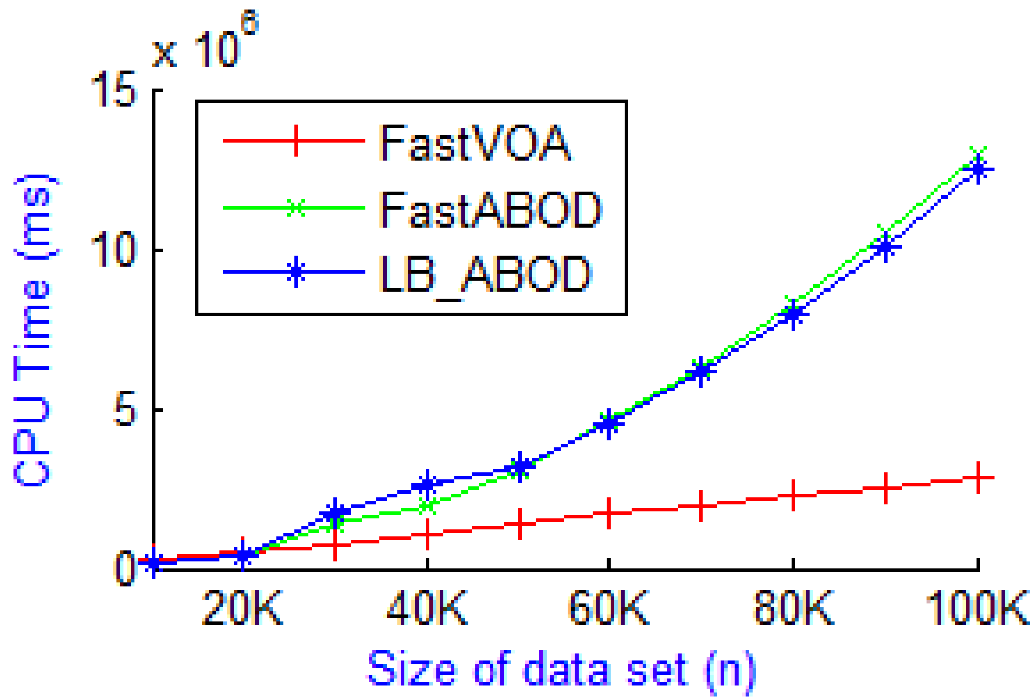
  The capability of algorithms {SimpleVOA, FastVOA} vs. {ABOD, FastABOD} [KSZ'08] to retrieve the most likely outliers on Multiple Features dataset



Experiments

- **Efficiency:**

  The CPU time of algorithms {FastVOA} vs. {FastABOD, LB_ABOD} [KSZ'08] on synthetic datasets of 100 dimensions



Conclusion

- A near-linear time algorithm to approximate variance of angle, a robust outlier factor
- A theoretical error analysis
- Experimental results on the accuracy, effectiveness and efficiency on synthetic and real world data sets