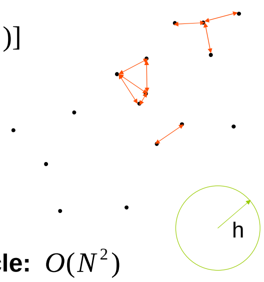
2-point correlation function

Definition:

$$dP = \lambda^2 dV_1 dV_2 [1 + \xi(h_{12})]$$

Must compute:

$$\boxed{\sum_{q}^{N} \sum_{r \neq q}^{N} I(\|x_q - x_r\| < h)}$$



➡ **Main obstacle:** $O(N^2)$

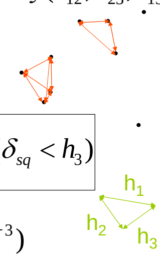
3-point correlation function

Def: $dP = \lambda^3 dV_1 dV_2 dV_3 \cdot$

$$[1 + \xi(h_{12}) + \xi(h_{23}) + \xi(h_{13}) + \zeta(h_{12}, h_{23}, h_{13})]$$

Must compute:

$$\boxed{\sum_{q}^{N} \sum_{r \neq q}^{N} \sum_{s \neq r \neq a}^{N} I(\delta_{qr} < h_1) I(\delta_{rs} < h_2) I(\delta_{sq} < h_3)}$$



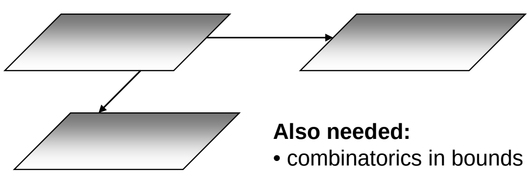
➡ **Main obstacle:** $O(N^3)$

Exclusion and inclusion,

on n>2 nodes.

Application of HODC principle

Do pairwise checking of constraints.



Also needed:

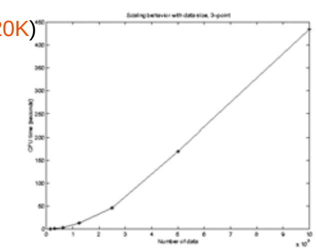
- combinatorics in bounds
- pairwise recursion
- recursion in leaves
- redundancy elimination

Speedup Results

Example:
(biggest previous: 20K)

VIRGO data
N = 75,000,000

naïve: 5×10^9
multi-tree:
small h: **55 sec.**
large h: 24 hrs



Analysis: $O(N^{\log n})$

c = p T

hard. known.

EASIER?

→ Monte Carlo?

$$\boxed{\hat{p} \pm z_{\varepsilon/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{S}}}$$

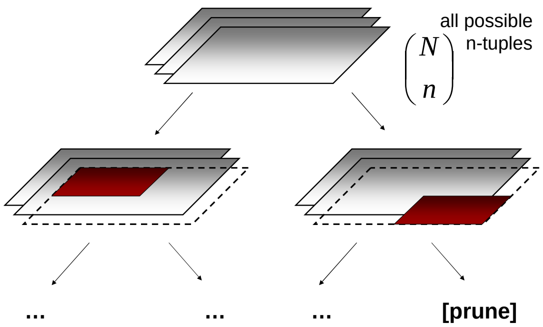
No dependence on N!
… but depends on p.

Basic idea:

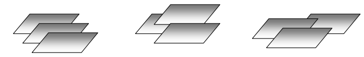
1. Remove some junk
(Run exact algorithm for a while)
→ **make p larger**

2. Sample from the rest

We get disjoint sets from the recursion tree



Now do stratified sampling



$$T_1 + T_2 + T_3 = T$$

$$\frac{T_1}{T}\hat{p}_1 + \frac{T_2}{T}\hat{p}_2 + \frac{T_3}{T}\hat{p}_3 = \hat{p}$$

$$\left(\frac{T_1}{T}\right)^2 \hat{\sigma}_1^{\,2} + \left(\frac{T_2}{T}\right)^2 \hat{\sigma}_2^{\,2} + \left(\frac{T_3}{T}\right)^2 \hat{\sigma}_3^{\,2} = \hat{\sigma}^2$$

Note 1: Wald interval not perfect

Poor coverage for small p

Agresti-Coull not usable here; use min. p

Note 2: Adaptive Neyman sampling

$$S_k^{\,opt} = \frac{\left(\frac{T_k}{T}\right)^2 \sigma_k^{\,2}}{\sigma}$$

→ Why not use $\hat{\sigma}_k^{\,2}$?

→ Update allocations periodically

Speedup Results

Example:
VIRGO data
N = 75,000,000

naïve: $5\text{x}10^9$
multi-tree:
large h: 24 hrs

multi-tree monte carlo:
99% confidence:
96 sec