Exploring the Hyperuniformity with Python

Diala Hawat

Guillaume Gautier, Rémi Bardenet and Raphaël Lachièze-Rey

Université de Lille, CNRS, Centrale Lille ; UMR 9189 – CRIStAL, F-59000 Lille, France.
Université de Paris, Map5, Paris, France.

Contents

structure-factor 1.0.1

```
pip install structure-factor
```


Let $\mathcal{X}$ be a point process of $\mathbb{R}^d$ of intensity ρ, $\mathcal{X}$ is hyperuniform iff[1]

- **Variance:**

$$\lim_{R\to\infty} \frac{\mathrm{Var}(\mathrm{Card}(\mathcal{X} \cap \mathbf{B}(0, R)))}{\mathcal{L}^d(\mathbf{B}(0, R))} = 0,$$

[1] **Tor:18**.

Hyperuniformity

Let $\mathcal{X}$ be a point process of $\mathbb{R}^d$ of intensity ρ, $\mathcal{X}$ is hyperuniform iff[1]

- **Variance:**

$$\lim_{R\to\infty} \frac{\mathrm{Var}(\mathrm{Card}(\mathcal{X}\cap \mathbf{B}(0,R)))}{\mathcal{L}^d(\mathbf{B}(0,R))} = 0,$$

- **Structure Factor:**

$$S(\mathbf{k}) = 1 + \rho\mathcal{F}(g-1)(\mathbf{k}) \xrightarrow{|\mathbf{k}|\to 0} 0$$

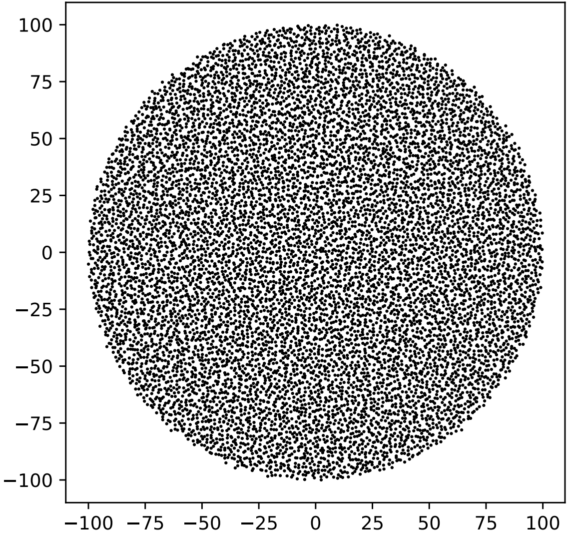
[1] **Tor:18**.

Ginibre Ensemble

structure-factor 1.0.1

`pip install structure-factor`

Point Pattern

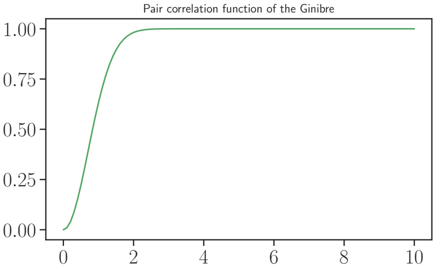


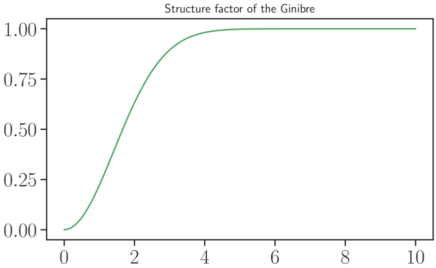
Ginibre Ensemble

structure-factor 1.0.1

`pip install structure-factor`

Point Pattern





Scattering intensity

Let $W = [-L/2, L/2]^d$ and $\mathcal{X} \cap W = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$,

- Scattering intensity[2]

$$\widehat{S}_{\mathrm{SI}}(\mathbf{k}) \triangleq \frac{1}{N} \left| \sum_{j=1}^{N} e^{-i\langle \mathbf{k}, \mathbf{x}_j \rangle} \right|^2, \ \mathbf{k} \in \mathbb{R}^d.$$

[2] **Kla+al:20**.

Scattering intensity

Let $W = [-L/2, L/2]^d$ and $\mathcal{X} \cap W = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$,

- Scattering intensity[2]

$$\widehat{S}_{\mathrm{SI}}(\mathbf{k}) \triangleq \frac{1}{N} \left| \sum_{j=1}^{N} e^{-i\langle \mathbf{k}, \mathbf{x}_j \rangle} \right|^2, \ \mathbf{k} \in \mathbb{R}^d.$$

- Allowed wavevectors

$$\mathbf{k} \in \{\frac{2\pi}{L}\mathbf{n},\ \mathbf{n} \in (\mathbb{Z}^d)^*\}.$$

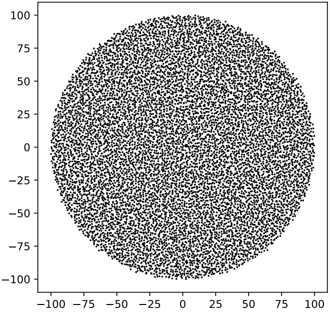
[2] **Kla+al:20**.

Test the scattering intenisty

structure-factor 1.0.1

`pip install structure-factor`

Estimators of the structure factor

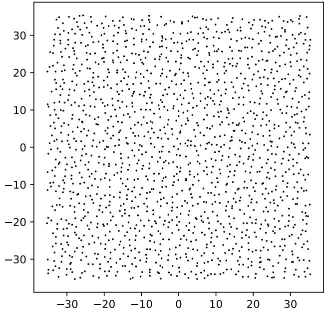


Test the scattering intenisty

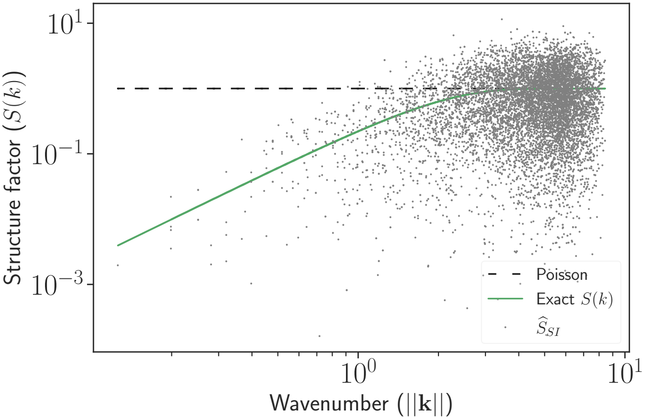
structure-factor 1.0.1

`pip install structure-factor`

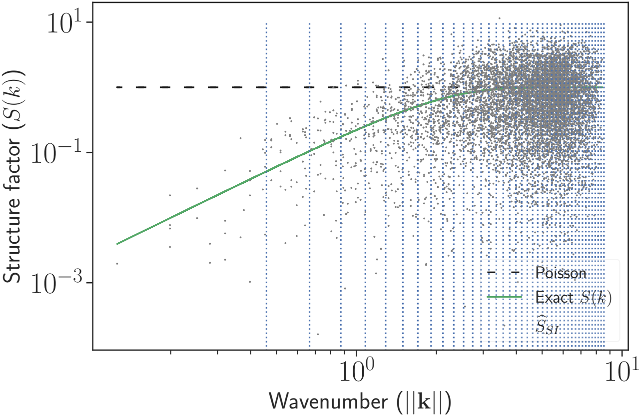
Estimators of the structure factor



Test the scattering intenisty



Test the scattering intenisty

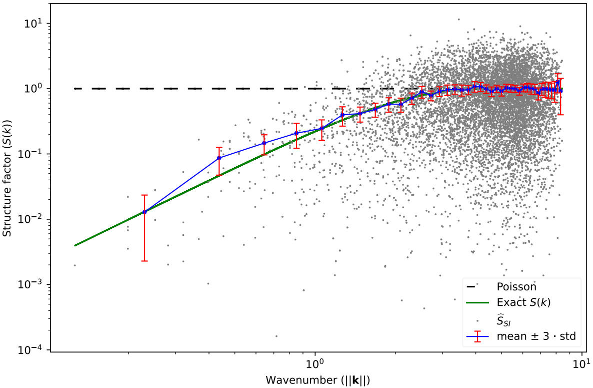


Test the scattering intenisty

structure-factor 1.0.1

`pip install structure-factor`

Estimators of the structure factor

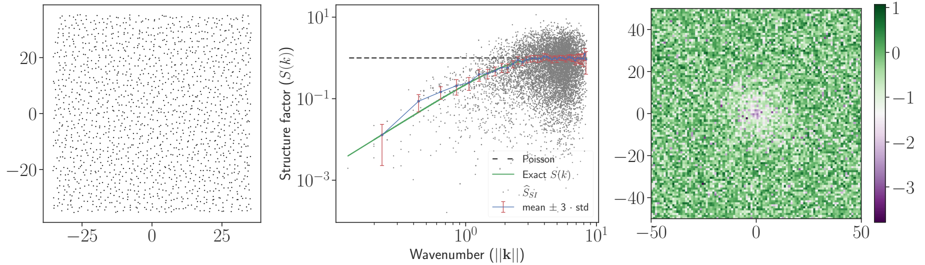


Test the scattering intenisty

structure-factor 1.0.1

`pip install structure-factor`

Estimators of the structure factor



Test the scattering intenisty

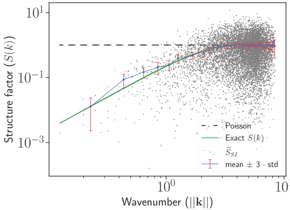


Figure: Box window and allowed wavevectors.

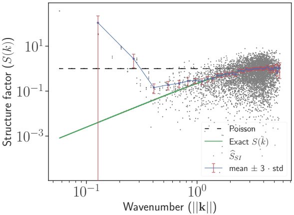


Figure: Box window and non allowed wavevectors.

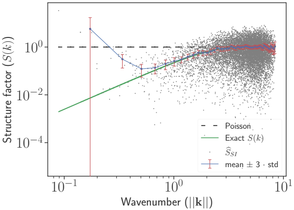


Figure: Ball window.

Estimators using Hankel transform

structure-factor 1.0.1
pip install structure-factor

For isotropic point processes:

- $S(\mathbf{k}) = 1 + \rho \underbrace{\mathcal{F}(\underbrace{g}_{g(\mathbf{r})=g(\|\mathbf{r}\|)} - 1)(\mathbf{k})}_{\text{Symmetric Fourier transform}}, \ \mathbf{k} \in \mathbb{R}^d.$
- $S(k) = 1 + \rho \mathcal{F}_s(g-1)(k), \ k \in \mathbb{R}.$
- $\mathcal{F}_s(f)(k) = \frac{(2\pi)^{d/2}}{k^{d/2-1}} \mathcal{H}_{d/2-1}(\tilde{f})(k), \quad \tilde{f} : x \mapsto f(x) x^{d/2-1}.$
- $S(k) = 1 + \rho \frac{(2\pi)^{d/2}}{k^{d/2-1}} \mathcal{H}_{d/2-1}(\tilde{g}-1)(k), \quad \tilde{g} : x \mapsto g(x) x^{d/2-1}.$

Method

Estimating the pcf $\rightarrow$ Interpolation $\rightarrow$ Estimating the Hankel transform

Method

Estimating the pcf $\rightarrow$ Interpolation $\rightarrow$ Estimating the Hankel transform

For isotropic point processes:

- R package `spatstat` .

A. Baddeley and E. Rubak and R. Turner, *Spatial Point Patterns Methodology and Applications with R.*

For isotropic point processes:

- R package `spatstat` .
- pcf.ppp: Direct kernel estimation.

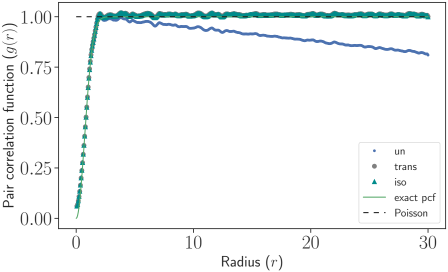
A. Baddeley and E. Rubak and R. Turner, *Spatial Point Patterns Methodology and Applications with R.*

Estimating the pair correlation function

structure-factor 1.0.1
`pip install structure-factor`

For isotropic point processes:

- R package `spatstat` .
- pcf.ppp: Direct kernel estimation.



pcf.ppp

A. Baddeley and E. Rubak and R. Turner, *Spatial Point Patterns Methodology and Applications with R.*

Estimating the pair correlation function

structure-factor 1.0.1

`pip install structure-factor`

For isotropic point processes:

- R package `spatstat` .
- pcf.ppp: Direct kernel estimation.
- pcf.fv:

A. Baddeley and E. Rubak and R. Turner, *Spatial Point Patterns Methodology and Applications with R.*

Estimating the pair correlation function

For isotropic point processes:

- R package `spatstat`.
- pcf.ppp: Direct kernel estimation.
- pcf.fv: $\widehat{g}(r) = \frac{K'(r)}{2\pi r}$.

A. Baddeley and E. Rubak and R. Turner, *Spatial Point Patterns Methodology and Applications with R.*

Estimating the pair correlation function

structure-factor 1.0.1

`pip install structure-factor`

For isotropic point processes:

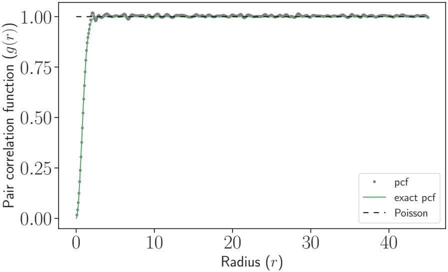
- R package `spatstat` .
- pcf.ppp: Direct kernel estimation.
- pcf.fv: Using the derivative of K .

A. Baddeley and E. Rubak and R. Turner, *Spatial Point Patterns Methodology and Applications with R.*

Estimating the pair correlation function

For isotropic point processes:

- R package `spatstat` .
- pcf.ppp: Direct kernel estimation.
- pcf.fv: Using the derivative of K .



pcf.fv

A. Baddeley and E. Rubak and R. Turner, *Spatial Point Patterns Methodology and Applications with R.*

Clean and Interpolate

Method

Estimate the pcf → Clean and Interpolate → Estimate the Hankel transform

	r	un	trans	iso
0	0.000000	inf	inf	inf
1	0.060120	0.060690	0.060690	0.060690
2	0.120240	0.068330	0.068330	0.068354
3	0.180361	0.086678	0.086678	0.086850
4	0.240481	0.108363	0.108363	0.108717
...	...	...	...	...
495	29.759519	0.816099	1.006761	1.004460
496	29.819639	0.815444	1.006340	1.004486
497	29.879760	0.815478	1.006760	1.005236
498	29.939880	0.815336	1.006956	1.005530
499	30.000000	0.809902	1.000599	0.999123

Figure: pcf.ppp

	r	pcf
0	0.000000	NaN
1	0.087891	0.016679
2	0.175781	0.041919
3	0.263672	0.077902
4	0.351562	0.124654
...	...	...
508	44.648438	1.005216
509	44.736328	1.003781
510	44.824219	1.001746
511	44.912109	0.999059
512	45.000000	0.995767

Figure: pcf.fv

Hankel Transform

Method

Estimate the pcf → Clean and Interpolate → Estimate the Hankel transform

Estimating the Hankel Transform

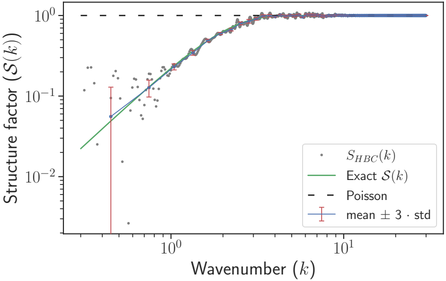
- Using the Discret Hankel Transform.

$$\mathcal{H}_\nu(f)(k_m) \approx \alpha \sum_{j=1}^{N-1} \frac{2}{\eta_{\nu N} J^2_{\nu+1}(\eta_{\nu j})} J_\nu\left(\frac{\eta_{\nu m}\eta_{\nu j}}{\eta_{\nu N}}\right) f(r_j).$$

N. Baddour and U. Chouinard, *Theory and operational rules for the discrete Hankel transform.*

Estimating the Hankel Transform

- Using the Discret Hankel Transform.



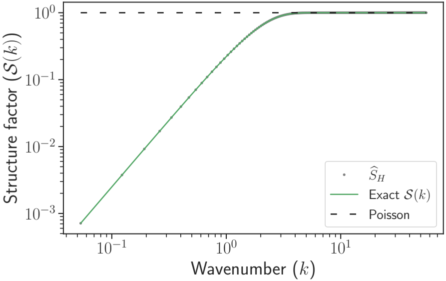
Approximation using the DHT.

Estimating the Hankel Transform

structure-factor 1.0.1

`pip install structure-factor`

- Using the Discret Hankel Transform.



Approximation using the DHT.

Estimating the Hankel Transform

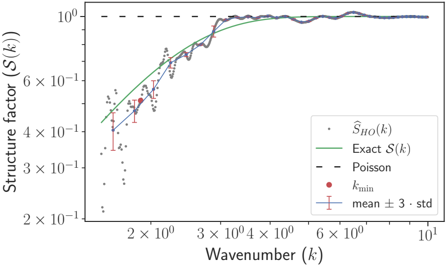
- Using the Discret Hankel Transform.
- Using Ogata quadrature.

$$\mathcal{H}_\nu(f)(k) \approx \pi \sum_{j=1}^{\infty} w_{\nu j} \frac{\pi}{k^2 h} \psi(h\xi_{\nu j}) f(\frac{\pi}{kh} \psi(h\xi_{\nu j})) J_\nu(\frac{\pi}{h} \psi(h\xi_{\nu j})) \psi'(h\xi_{\nu j}).$$

H. Ogata, *A Numerical Integration Formula Based on the Bessel Functions.*

Estimating the Hankel Transform

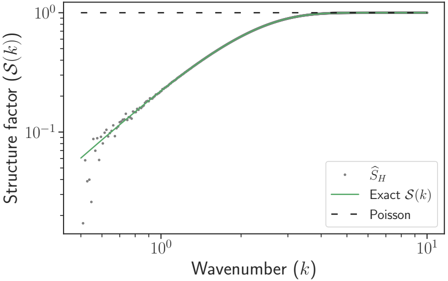
- Using the Discret Hankel Transform.
- Using Ogata quadrature.



Approximation using Ogata quadrature.

Estimating the Hankel Transform

- Using the Discret Hankel Transform.
- Using Ogata quadrature.



Approximation using Ogata quadrature.

H index

$$\mathcal{X} \text{ is effectively hyperuniform } \iff H = \frac{\widehat{S}(0)}{\widehat{S}(k_{peak})} \leq 10^{-3},$$

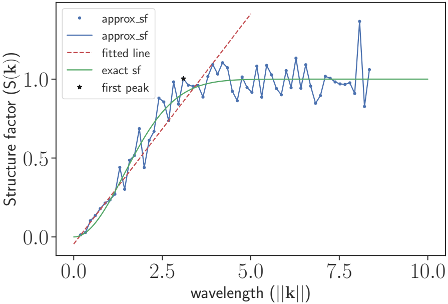
- $\widehat{S}(0)$ is a linear extrapolation of the estimated structure factor $\widehat{S}$ in $k = 0$.
- k_{peak} is the location of the first dominant peak value of $\widehat{S}$.

S. Torquato, *Hyperuniform States of Matter.*

H index

Effective Hyperuniformity

$$\mathcal{X} \text{ is effectively hyperuniform} \iff H = \frac{\widehat{S}(0)}{\widehat{S}(k_{peak})} \leq 10^{-3},$$



S. Torquato, *Hyperuniform States of Matter.*

Power decay of the structure factor

structure-factor 1.0.1

`pip install structure-factor`

$\mathcal{X}$ is hyperuniform with $|S(\mathbf{k})| \sim c\|\mathbf{k}\|^\alpha$ in the neighborhood of 0 then,

α	$\mathrm{Var}\left[\mathrm{Card}(\mathcal{X} \cap B(0, R))\right]$	class
> 1	$O(R^{d-1})$	I
1	$O(R^{d-1}\log(R))$	II
$]0, 1[$	$O(R^{d-\alpha})$	III

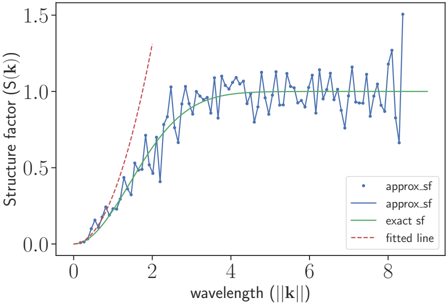
S. Cost, *Order, Fluctuations, Rigidities.*

Power decay of the structure factor

structure-factor 1.0.1

`pip install structure-factor`

α	$\mathrm{Var}\left[\mathrm{Card}(\mathcal{X} \cap B(0,R))\right]$	class
>1	$O(R^{d-1})$	I
1	$O(R^{d-1}\log(R))$	II
$]0,1[$	$O(R^{d-\alpha})$	III



S. Cost, *Order, Fluctuations, Rigidities.*

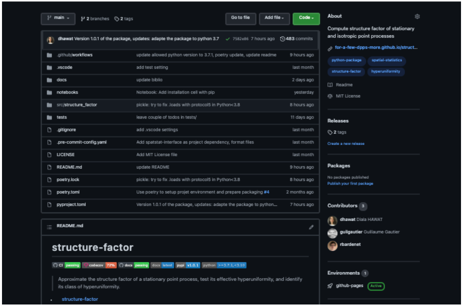
structure-factor

structure-factor 1.0.1

`pip install structure-factor`

Python Package

In [1]:
```
!pip install structure-factor
```



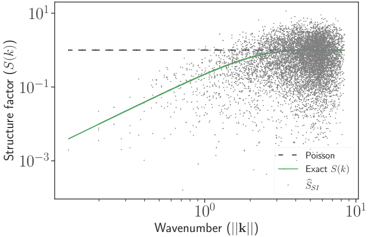
Perspective

Rajala2020SpectralEF.

THANK YOU.

structure-factor 1.0.1
pip install structure-factor

structure-factor 1.0.1
pip install structure-factor



Github

Documentation

Personal webpage