Statistical learning for spatio-temporal point processes

Jorge Mateu
University Jaume I, Castellón, Spain

Based on joint works with Abdollah Jalilian, Yao Xie and George Mohler

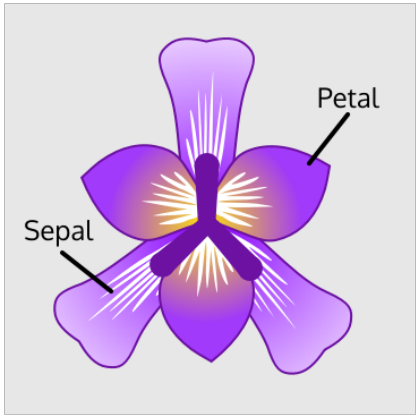
International Seminar Series, ITS, Departemen Statistika, Surabaya, Indonesia

May 23, 2023

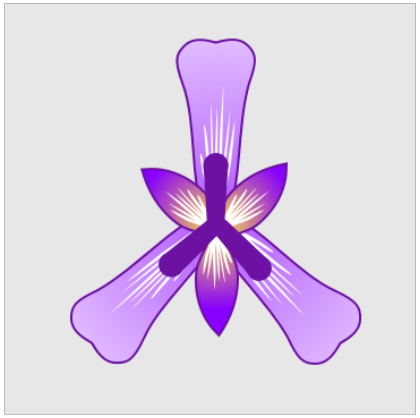
Presentation based on the following papers

- Jalilian, A. and Mateu, J. (2023). Assessing similarities between spatial point patterns with a Siamese Neural Network discriminant model. **Advances in Data Analysis and Classification**. doi: 10.1007/s11634-021-00485-0.
- Dong, Z., Zhu, S., Xie, Y., Mateu, J. and Rodriguez-Cortes, F. (2023). Non-stationary spatio-temporal point process modeling for high-resolution COVID-19 data. **Journal of the Royal Statistical Society C**. doi: 10.1093/jrsssc/qlad013.
- Mohler, G. and Mateu, J. (2023). Second order preserving point process permutations. **Stat**. doi: 10.1002/sta4.558.
- Mateu, J. and Jalilian, A. (2023). Spatial point processes and neural networks: a convenient couple. **Spatial Statistics**. doi: 10.1016/j.spasta.2022.100644.

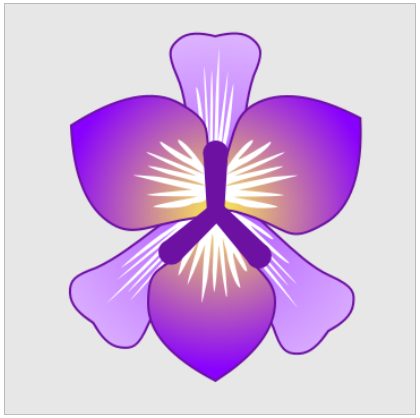
Fisher's or Anderson's iris dataset



Iris Versicolor



Iris Setosa



Iris Virginica

i	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
1	5.1	3.5	1.4	0.2	
⋮	⋮	⋮	⋮	⋮	setosa
50	5.0	3.3	1.4	0.2	
51	7.0	3.2	4.7	1.4	
⋮	⋮	⋮	⋮	⋮	versicolor
100	5.7	2.8	4.1	1.3	
101	6.3	3.3	6.0	2.5	
⋮	⋮	⋮	⋮	⋮	virginica
150	5.9	3.0	5.1	1.8	

Data for discriminant analysis

- observations

$$\mathbf{x}^{(s_i,t_i)} = (x_1^{(s_i,t_i)}, \ldots, x_4^{(s_i,t_i)}), \quad i = 1, \ldots, N = 150$$

- dataset

$$\mathcal{D} = \{\mathbf{x}^{(s_i,t_i)} : i = 1, \ldots, N\}$$

  - group membership is known

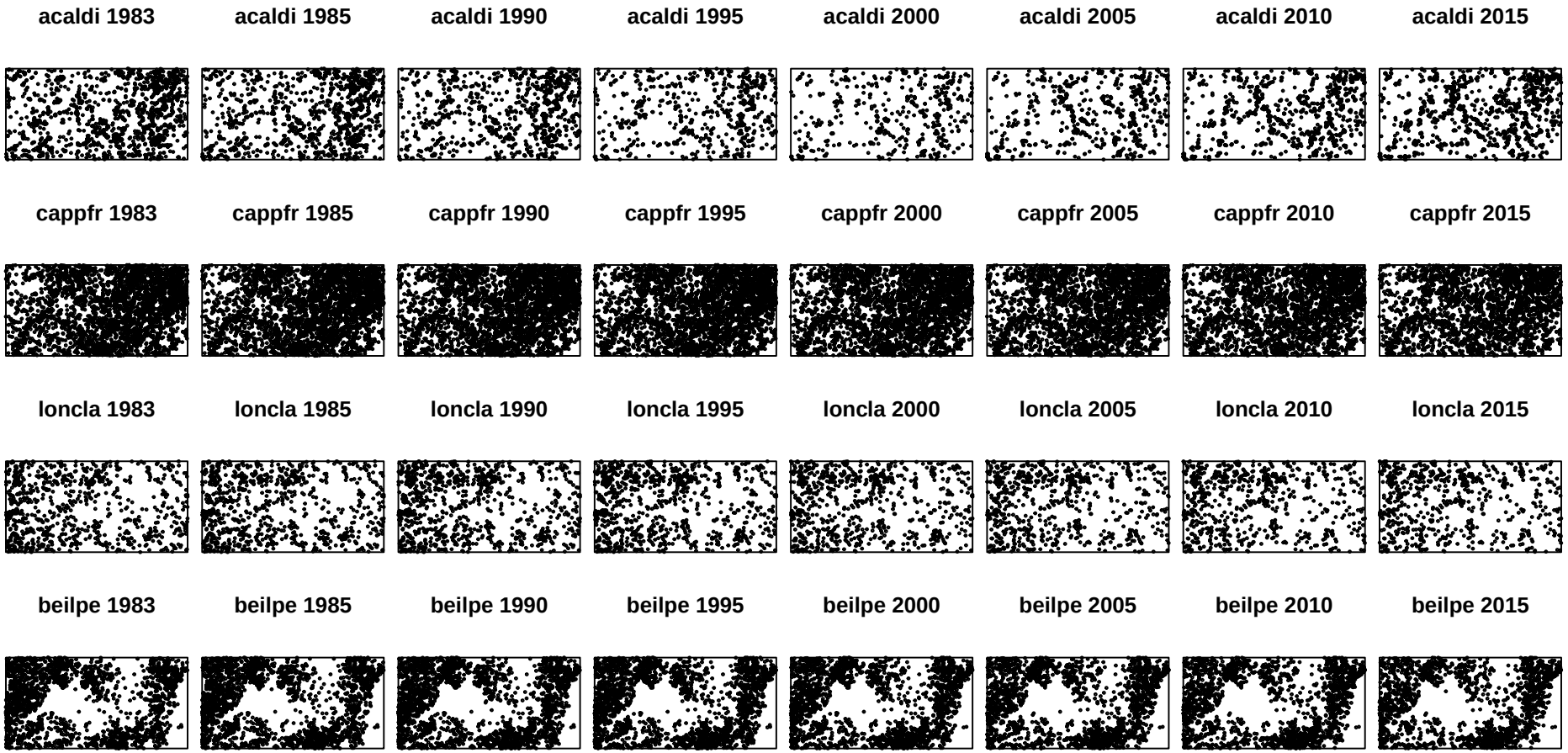
$$s_i \in \{1, \ldots, m\}, \quad m = 3$$

  - repetitions are available

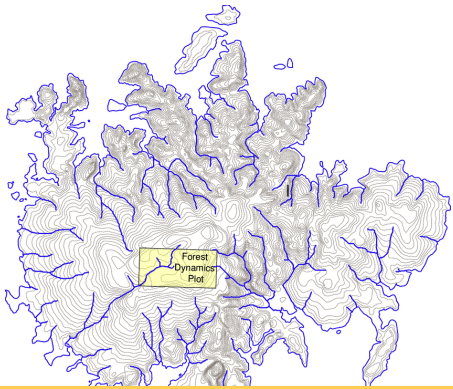
$$t_i \in \{1, \ldots, T\}, \quad T = 50$$

- new query data $\mathbf{x}^{(*)} = (x_1^{(*)}, \ldots, x_4^{(*)})$ belongs to which group?

BCI dataset



- $m = 130$ most abundant species
- $T = 8$ temporally dependent repititions (censuses)
- $N = 1040$ spatial point patterns (1808725 alive trees)
- a discriminant model for spatial point patterns?



Notations

W a specific bounded observation window $W \subset \mathbb{R}^d$

$\mathbf{x}$ a spatial points pattern observed on W

$\mathbf{u}, \mathbf{v}$ locations of events occurred inside W

$\mathcal{X}$ space of all spatial point patterns on W

X underlying spatial point process that generated $\mathbf{x}$

$$X : (\Omega, \mathcal{A}, \mathbb{P}) \to (\mathcal{X}, \mathcal{N}, \mathbb{P}_X)$$

$f_X(\mathbf{x})$ density of $\mathbb{P}_X$ w.r.t. standard Poisson process on W

$\rho(\mathbf{u})$ intensity function

$g(\mathbf{u}, \mathbf{v})$ pair correlation function

$K(r)$ K-function

$F(r)$ empty space distribution function

$G(r)$ nearest neighbor distance distribution function

$J(r)$ J-function

Dissimilarity function

- a dissimilarity (distance) function

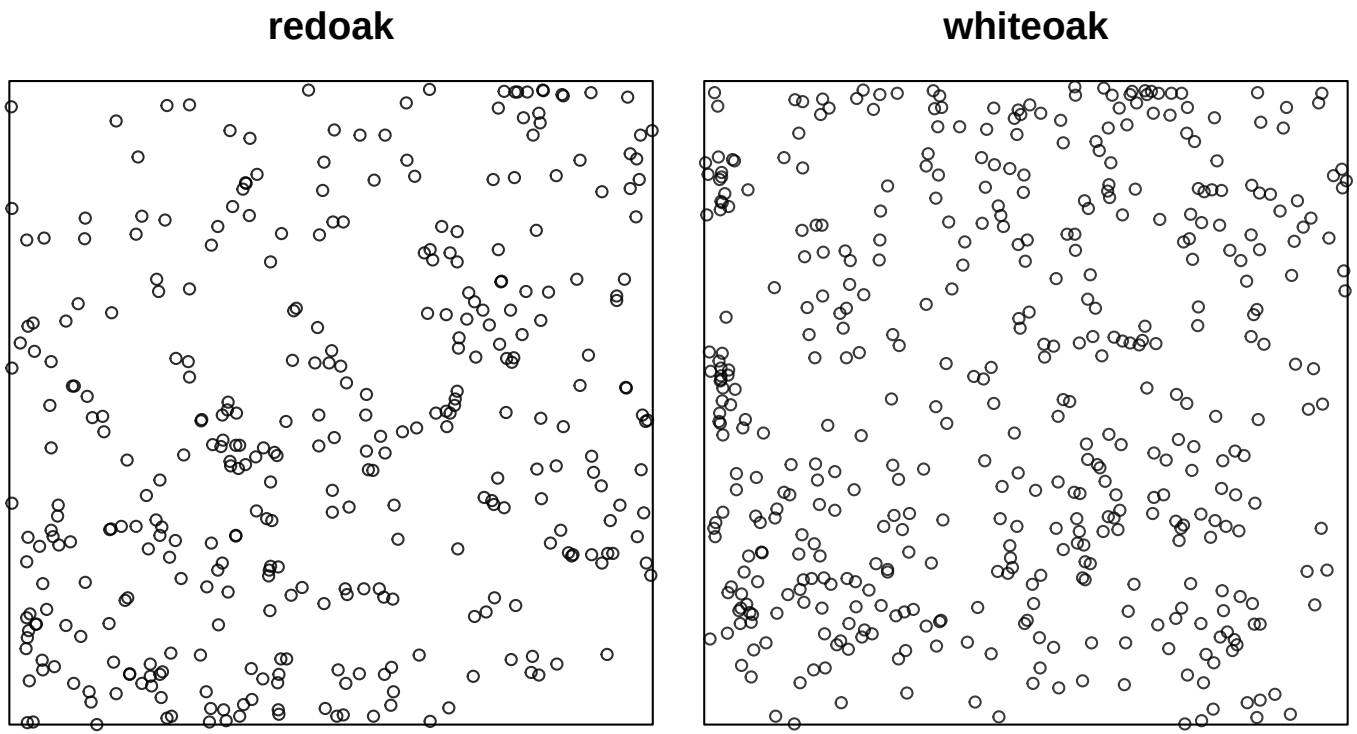
$$D : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$$

- good dissimilarity function

$$D(\mathbf{x}^{(s_i,t_i)}, \mathbf{x}^{(s_{i'},t_{i'})}) = \begin{cases} \text{as small as possible} & s_i = s_{i'} \\ \text{as large as possible} & s_i \neq s_{i'} \end{cases}$$

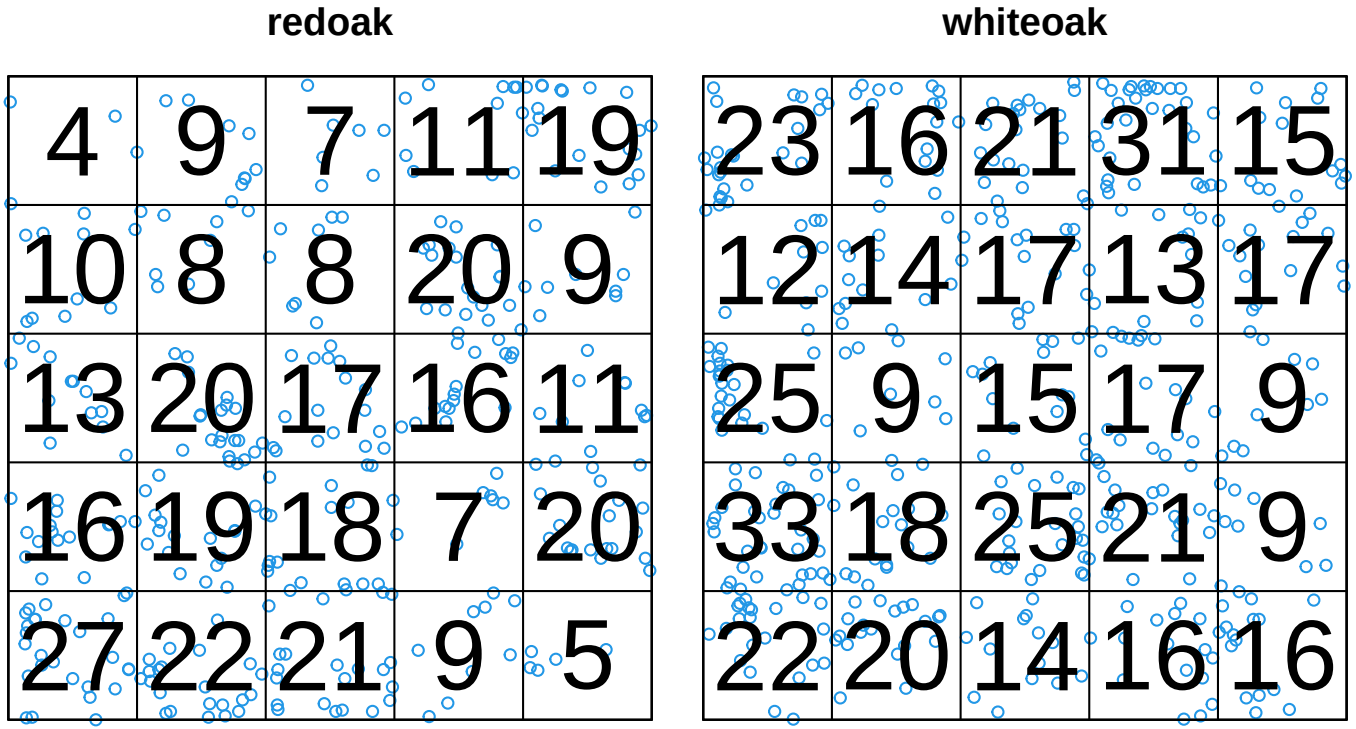
  - compares distinctive features between groups
  - allows for variations within groups
- comparing patterns versus comparing processes
  - pattern matching: similarities of observed point patterns
  - process matching: similarities of the generative point processes

Dissimilarity based on pattern matching



observed point patterns

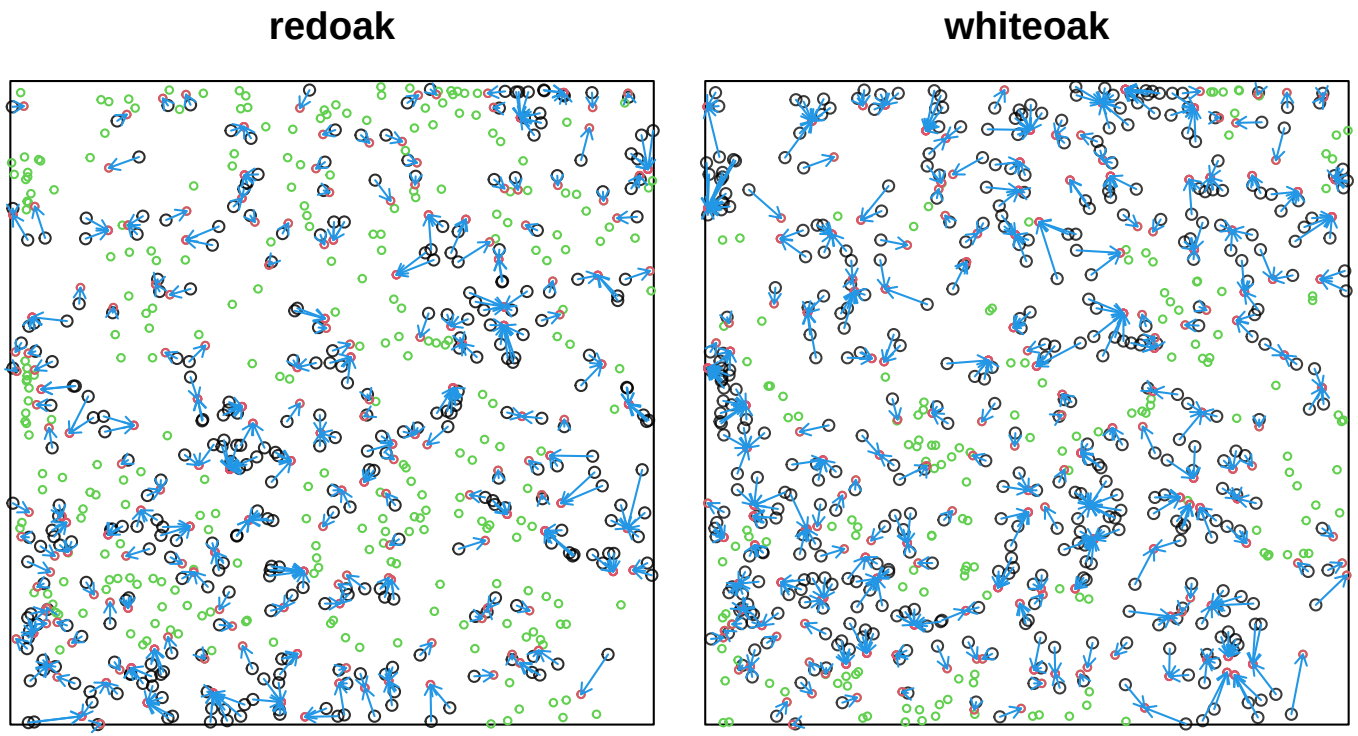
Dissimilarity based on pattern matching



Alba-Fernández et al. (2016): dissimilarity of quadrat counts

$$D(\mathbf{x},\mathbf{x}') = \sum_{j=1}^{m} \left\{ \left(1 - \frac{n(\mathbf{x}\cap B_j)[n(\mathbf{x})+n(\mathbf{x}')]}{n(\mathbf{x})[n(\mathbf{x}\cap B_j)+n(\mathbf{x}'\cap B_j)]}\right)^2 + \left(1 - \frac{n(\mathbf{x}'\cap B_j)[n(\mathbf{x})+n(\mathbf{x}')]}{n(\mathbf{x}')[n(\mathbf{x}\cap B_j)+n(\mathbf{x}'\cap B_j)]}\right)^2 \right\}$$

Dissimilarity based on pattern matching



Cholaquidis, et al. (2017): dissimilarity of nearest neighbour distances

$$D(\mathbf{x}, \mathbf{x}') = \max \left\{ \sup_{\mathbf{u} \in \mathbf{x}} \inf_{\mathbf{v} \in \mathbf{x}'} \frac{\|\mathbf{u} - \mathbf{v}\|}{\operatorname{diam}(W)}, \sup_{\mathbf{u} \in \mathbf{x}'} \inf_{\mathbf{v} \in \mathbf{x}} \frac{\|\mathbf{u} - \mathbf{v}\|}{\operatorname{diam}(W)} \right\} + |n(\mathbf{x}) - n(\mathbf{x}')|$$

Dissimilarity based on process matching

- $\mathbf{x}$ and $\mathbf{x}'$ are realizations of point processes X and X'
  - $\mathbb{P}_X = \mathbb{P}_{X'}$: $\mathbf{x}$ and $\mathbf{x}'$ are statistically indistinguishable; their differences are due to pure chance
  - $\mathbb{P}_X \neq \mathbb{P}_{X'}$: $\mathbf{x}$ and $\mathbf{x}'$ are expected to reflect some structural differences; their differences are results of some distinctive features in $\mathbb{P}_X$ and $\mathbb{P}_{X'}$
- a suitable summary statistic to quantify/reflect some relevant features of the underlying point processes

$$G : \mathcal{X} \to \mathcal{F}$$

- dissimilarity function based on distance of patterns in the feature space $\mathcal{F}$

$$D(\mathbf{x}, \mathbf{x}') = D_{\mathcal{F}}(G(\mathbf{x}), G(\mathbf{x}')) = \left\| G(\mathbf{x}) - G(\mathbf{x}') \right\|_{\mathcal{F}}$$

Examples of summary statistics

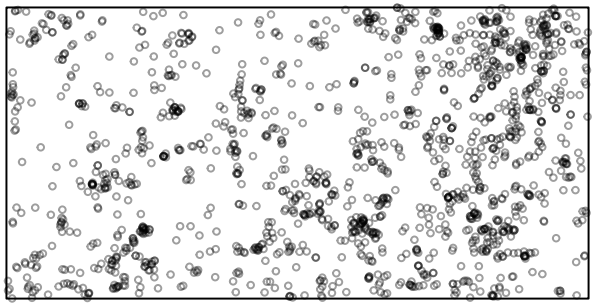
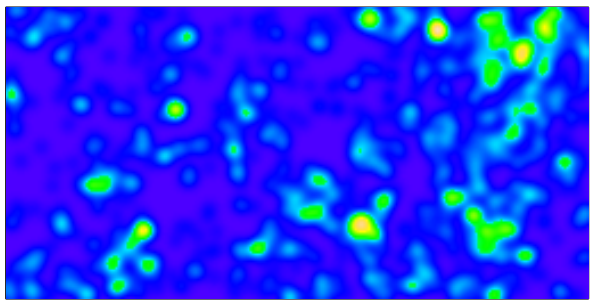
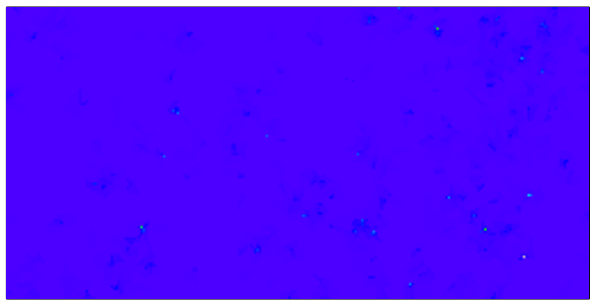
- general form of common summary statistics

$$G(\mathbf{x}) = \sum_{\mathbf{u}_1,\ldots,\mathbf{u}_\ell \in \mathbf{x}}^{\neq} h(\mathbf{u}_1,\ldots,\mathbf{u}_\ell; \mathbf{x} \backslash \{\mathbf{u}_1,\ldots,\mathbf{u}_\ell\})$$

- first order summary statistics

$$G(\mathbf{x}) = \widehat{\rho}(\mathbf{u};\mathbf{x}) = \sum_{\mathbf{v}\in\mathbf{x}} \frac{k(\mathbf{u}-\mathbf{v})}{\int_W k(\mathbf{u}-\mathbf{v}')\mathrm{d}\mathbf{v}'}$$

feature space $\mathcal{F}$ consists of non-negative pixel images on W

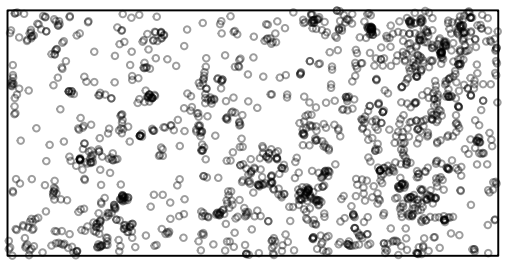
Examples of summary statistics

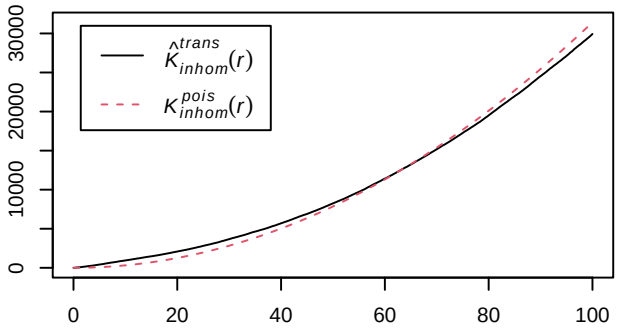
- second order summary statistics

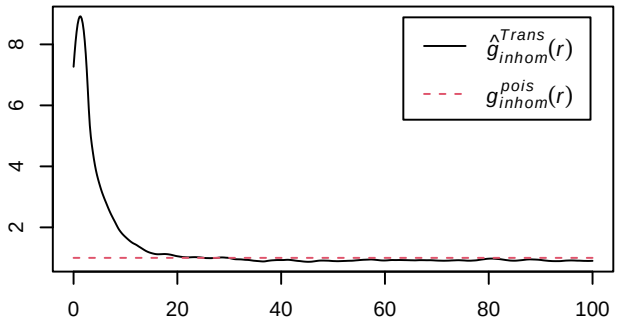
$$G(\mathbf{x}) = \widehat{K}(r;\mathbf{x}) = \sum_{\mathbf{u},\mathbf{v}\in\mathbf{x}}^{\neq} \frac{\mathbb{1}[\|\mathbf{u}-\mathbf{v}\| \leq r]}{\widehat{\rho}(\mathbf{u};\mathbf{x})\widehat{\rho}(\mathbf{v};\mathbf{x})|W \cap W_{\mathbf{u}-\mathbf{v}}|}$$

$$G(\mathbf{x}) = \widehat{g}(r;\mathbf{x}) = \frac{1}{2\pi} \sum_{\mathbf{u},\mathbf{v}\in\mathbf{x}}^{\neq} \frac{k(r-\|\mathbf{u}-\mathbf{v}\|)}{\|\mathbf{u}-\mathbf{v}\|\widehat{\rho}(\mathbf{u};\mathbf{x})\widehat{\rho}(\mathbf{v};\mathbf{x})|W \cap W_{\mathbf{u}-\mathbf{v}}|}$$

feature space $\mathcal{F}$ consists of matrices with at least two columns







Examples of summary statistics

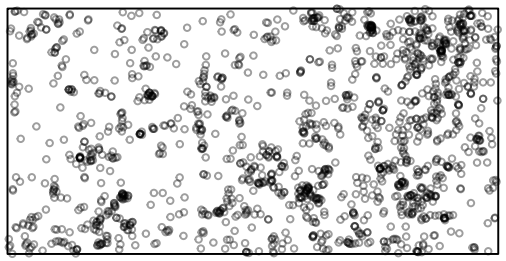
- inter-point distance summary statistics

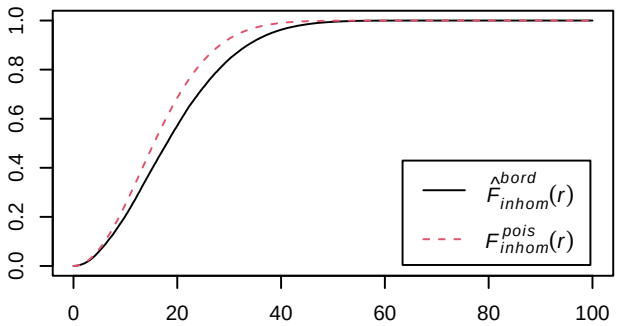
$$G(\mathbf{x}) = \widehat{F}(r;\mathbf{x}) = 1 - \frac{1}{\#(L \cap W_{\ominus r})} \sum_{\mathbf{v} \in L \cap W_{\ominus r}} \prod_{\substack{\mathbf{u} \in \mathbf{x} \\ \|\mathbf{u}-\mathbf{v}\| \leq r}} \left[1 - \frac{\inf_{\mathbf{u}'} \widehat{\rho}(\mathbf{u}';\mathbf{x})}{\widehat{\rho}(\mathbf{u};\mathbf{x})} \right]$$

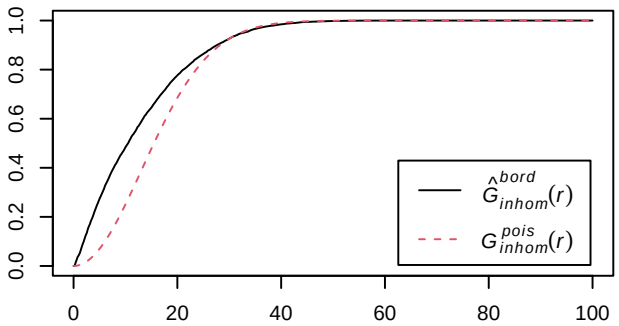
$$G(\mathbf{x}) = \widehat{G}(r;\mathbf{x}) = 1 - \frac{1}{\#(\mathbf{x} \cap W_{\ominus r})} \sum_{\mathbf{v} \in \mathbf{x} \cap W_{\ominus r}} \prod_{\substack{\mathbf{u} \in \mathbf{x} \setminus \{\mathbf{v}\} \\ \|\mathbf{u}-\mathbf{v}\| \leq r}} \left[1 - \frac{\inf_{\mathbf{u}'} \widehat{\rho}(\mathbf{u}';\mathbf{x})}{\widehat{\rho}(\mathbf{u};\mathbf{x})} \right]$$

$$G(\mathbf{x}) = \widehat{J}(r;\mathbf{x}) = \frac{1 - \widehat{G}(r;\mathbf{x})}{1 - \widehat{F}(r;\mathbf{x})}$$

feature space $\mathcal{F}$ consists of matrices with at least two columns







Which summary statistic?

- which summary statistic should be used?
  - visual inspection
  - prior information

$$\widehat{\rho}(\cdot;\mathbf{x}) \quad \widehat{K}(\cdot;\mathbf{x}) \quad \widehat{g}(\cdot;\mathbf{x}) \quad \widehat{F}(\cdot;\mathbf{x}) \quad \widehat{G}(\cdot;\mathbf{x}) \quad \widehat{J}(\cdot;\mathbf{x})$$

- which norm/distance in the feature space?

$$D(\mathbf{x},\mathbf{x}') = \left\|\widehat{\rho}(\cdot;\mathbf{x}) - \widehat{\rho}(\cdot;\mathbf{x}')\right\|_2 = \int_W \left[\widehat{\rho}(\mathbf{u};\mathbf{x}) - \widehat{\rho}(\mathbf{u};\mathbf{x}')\right]^2 \mathrm{d}\mathbf{u}$$

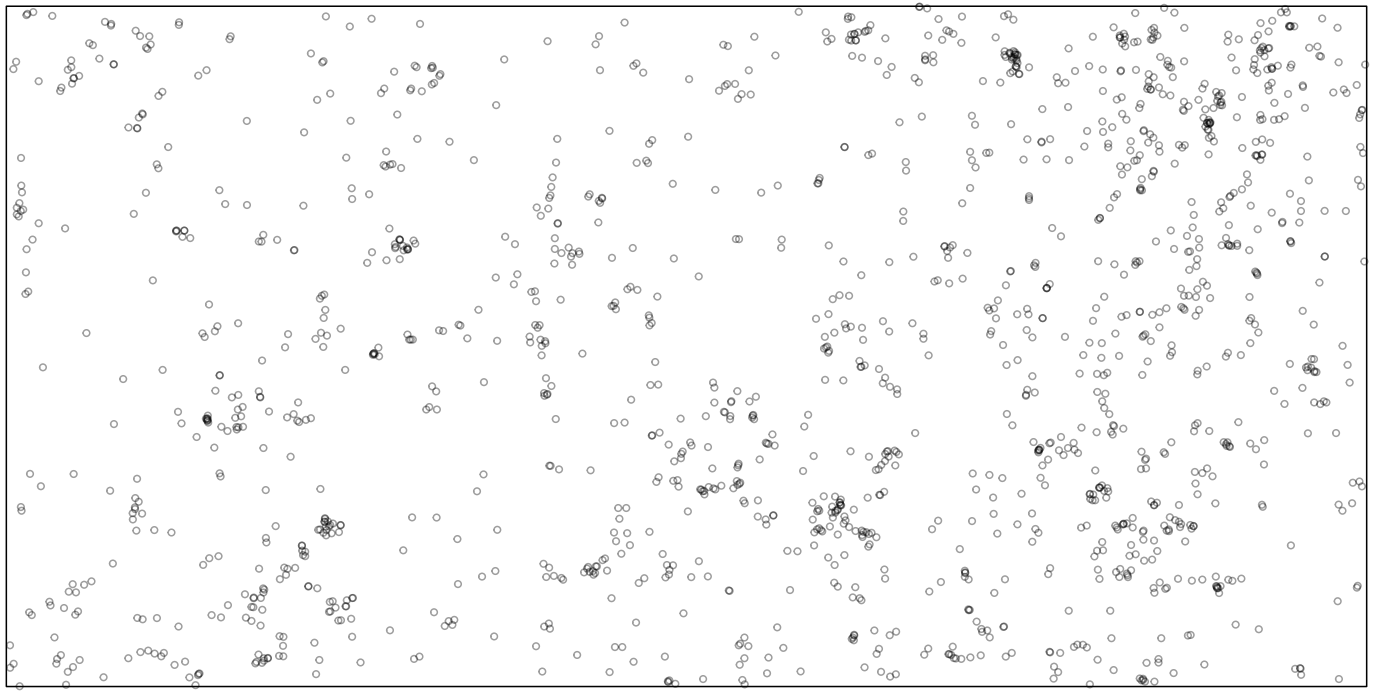
$$D(\mathbf{x},\mathbf{x}') = \left\|\widehat{K}(\cdot;\mathbf{x}) - \widehat{K}(\cdot;\mathbf{x}')\right\|_\infty = \sup_{0\le r\le r_{\max}} \left|\widehat{K}(r;\mathbf{x}) - \widehat{K}(r;\mathbf{x}')\right|$$

Mateu *et al.* (2015). On measures of dissimilarity between point patterns: classification based on prototypes and multidimensional scaling. **Biometrical Journal**.

Feature extraction by CNN

- partition $W \subset \mathbb{R}^2$ into a $d_1 \times d_2$ regular grid of cells

$$W = \bigcup_{i=1}^{d_1} \bigcup_{j=1}^{d_2} B_{ij}$$

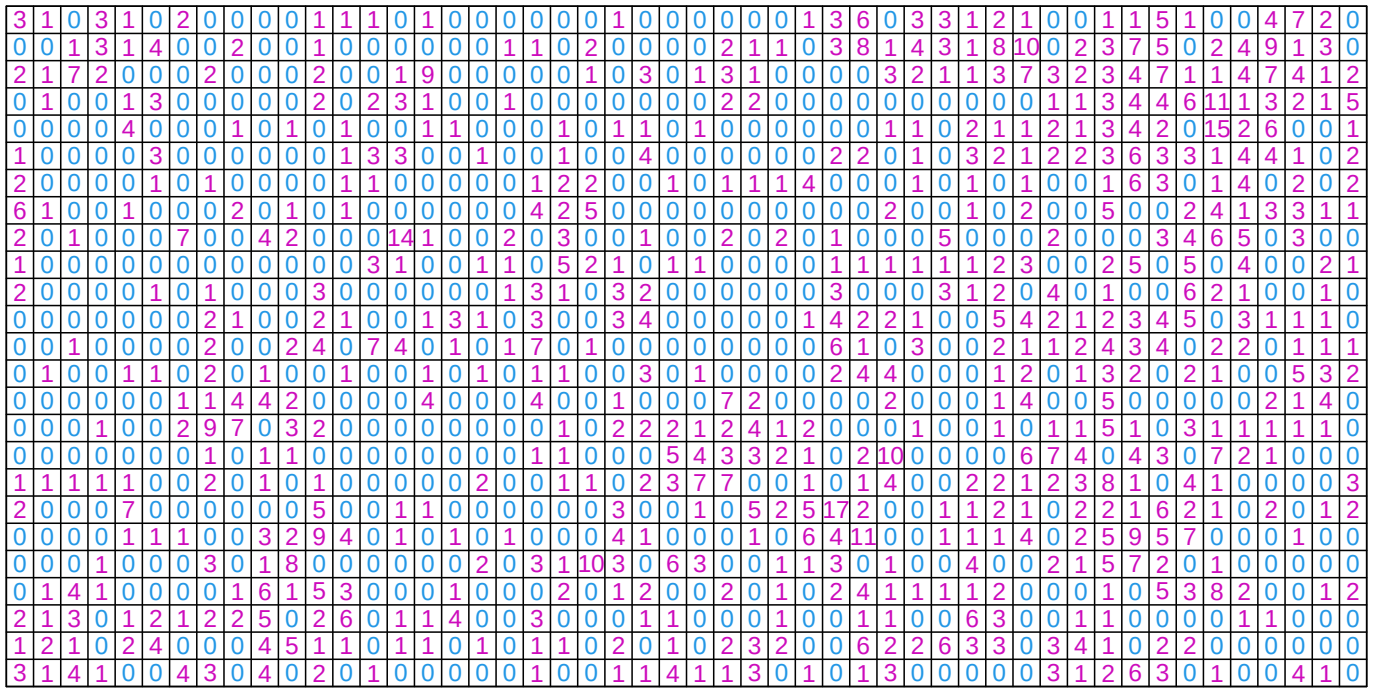


observed point pattern: $\mathbf{x}_W = \{\mathbf{u}_1, \ldots, \mathbf{u}_{n(\mathbf{x})}\}$
observation window: $W = [0, 1000] \times [0, 500]$

Feature extraction by CNN

- partition $W \subset \mathbb{R}^2$ into a $d_1 \times d_2$ regular grid of cells

$$W = \bigcup_{i=1}^{d_1} \bigcup_{j=1}^{d_2} B_{ij}$$



discretized point pattern: $\tilde{\mathbf{x}} = [x_{ij}]_{d_1 \times d_2}$

cell counts: $x_{ij} = n(\mathbf{x} \cap B_{ij}), i = 1, \ldots, d_1, j = 1, \ldots, d_2$

Convolution with kernels

- input layer of the network: discretized point pattern $\tilde{\mathbf{x}} = [x_{ij}]$
- given a set of $d_1^{(1)} \times d_2^{(1)}$ kernels/filters

$$\mathbf{F}^{(1,k)} = \left[F_{ij}^{(1,k)}\right]_{d_1^{(1)} \times d_2^{(1)}}, \quad k = 1, \ldots, \ell_1$$

- $\tilde{\mathbf{x}} = [x_{ij}]$ is convolved with each $\mathbf{F}^{(1,k)}$

$$(\tilde{\mathbf{x}} * \mathbf{F}^{(1,k)})_{ij} = \sum_{i'} \sum_{j'} x_{i+i',j+j'} F_{i'j'}^{(1,k)}$$

valid padding $i = 1, \ldots, d_1 - d_1^{(1)} + 1$ and $j = 1, \ldots, d_2 - d_2^{(1)} + 1$

zero padding $x_{i+i',j+j'}$ is set to be zero for $i + i' > d_1$ and $j + j' > d_2$

discretized pattern

17	13	6	4	10	7	4	4	9	16	22	37	10	40	33	21
1	11	1	4	11	3	2	6	4	2	6	9	14	35	47	12
12	4	9	6	21	2	22	6	4	8	3	9	10	18	24	16
4	1	6	10	5	9	16	12	1	11	11	14	19	28	15	5
1	3	28	12	9	4	9	7	17	9	11	6	15	10	10	18
5	8	4	4	0	2	4	15	29	26	18	7	38	16	17	8
4	6	8	39	5	6	15	19	8	25	18	11	13	47	12	4
19	11	18	22	6	6	6	11	11	8	11	21	15	14	3	5

$*$

kernel

−1	−1	−1
−1	8	−1
−1	−1	−1

convolution

25	−49	−36	31	−55	−38	−7	−23	−56	−58	−39	−56	84	177
−13	29	−19	118	−73	120	−19	−18	22	−46	−14	−66	−48	16
−59	−25	−16	−33	−16	57	14	−66	24	17	28	42	103	−9
−33	176	30	26	−22	3	−47	26	−52	−14	−85	−18	−78	−37
5	−76	−73	−81	−36	−45	12	106	73	31	−74	179	−34	11
−29	−48	245	−45	4	51	53	−80	71	17	−53	−65	248	−18

input: $(d_1, d_2) = (8, 16)$

kernel: $(d_1^{(1)}, d_2^{(1)}) = (3, 3)$

convolution: $(d_1 - d_1^{(1)} + 1, d_2 - d_2^{(1)} + 1) = (6, 14)$

discretized pattern

17	13	6	4	10	7	4	4	9	16	22	37	10	40	33	21
1	11	1	4	11	3	2	6	4	2	6	9	14	35	47	12
12	4	9	6	21	2	22	6	4	8	3	9	10	18	24	16
4	1	6	10	5	9	16	12	1	11	11	14	19	28	15	5
1	3	28	12	9	4	9	7	17	9	11	6	15	10	10	18
5	8	4	4	0	2	4	15	29	26	18	7	38	16	17	8
4	6	8	39	5	6	15	19	8	25	18	11	13	47	12	4
19	11	18	22	6	6	6	11	11	8	11	21	15	14	3	5

$*$

kernel

0	−1	0
−1	5	−1
0	−1	0

convolution

36	−25	−2	17	−7	−25	14	−1	−24	−6	−21	6	56	131
−13	28	−14	81	−45	84	−14	1	20	−19	9	−10	−7	24
−12	−18	21	−24	18	28	30	−39	26	16	25	28	78	8
−23	115	9	24	−9	14	−18	39	−20	11	−17	2	−19	−10
22	−28	−35	−20	−4	−21	16	79	49	28	−38	139	−32	39
−1	−27	156	−26	2	40	46	−44	65	25	−4	−46	180	−11

input: $(d_1, d_2) = (8, 16)$

kernel: $(d_1^{(1)}, d_2^{(1)}) = (3, 3)$

convolution: $(d_1 - d_1^{(1)} + 1, d_2 - d_2^{(1)} + 1) = (6, 14)$

discretized pattern

17	13	6	4	10	7	4	4	9	16	22	37	10	40	33	21
1	11	1	4	11	3	2	6	4	2	6	9	14	35	47	12
12	4	9	6	21	2	22	6	4	8	3	9	10	18	24	16
4	1	6	10	5	9	16	12	1	11	11	14	19	28	15	5
1	3	28	12	9	4	9	7	17	9	11	6	15	10	10	18
5	8	4	4	0	2	4	15	29	26	18	7	38	16	17	8
4	6	8	39	5	6	15	19	8	25	18	11	13	47	12	4
19	11	18	22	6	6	6	11	11	8	11	21	15	14	3	5

$*$

kernel

−2	−1	0
−1	1	1
0	1	2

convolution

−14	−17	46	17	13	21	10	3	−16	−20	−35	2	74	−33
1	14	32	21	19	58	−8	13	30	33	47	62	27	−70
34	50	15	−12	−2	16	−12	19	36	18	43	40	16	−22
37	41	−25	−20	−5	12	44	75	52	7	57	42	−11	−20
24	52	−19	−18	20	53	50	67	33	10	36	127	26	−1
39	83	58	−18	32	48	22	−18	−19	−13	14	40	−26	−67

input: $(d_1, d_2) = (8, 16)$

kernel: $(d_1^{(1)}, d_2^{(1)}) = (3, 3)$

convolution: $(d_1 - d_1^{(1)} + 1, d_2 - d_2^{(1)} + 1) = (6, 14)$

discretized pattern

17	13	6	4	10	7	4	4	9	16	22	37	10	40	33	21
1	11	1	4	11	3	2	6	4	2	6	9	14	35	47	12
12	4	9	6	21	2	22	6	4	8	3	9	10	18	24	16
4	1	6	10	5	9	16	12	1	11	11	14	19	28	15	5
1	3	28	12	9	4	9	7	17	9	11	6	15	10	10	18
5	8	4	4	0	2	4	15	29	26	18	7	38	16	17	8
4	6	8	39	5	6	15	19	8	25	18	11	13	47	12	4
19	11	18	22	6	6	6	11	11	8	11	21	15	14	3	5

kernel

$$*$$

0	1	0	0	−1	0
2	0	0	0	0	−2
0	1	0	0	−1	0

convolution

−18	24	−26	21	14	−4	−29	−49	−58	−123	−23
16	−41	2	2	33	3	27	−26	−48	−78	−65
−33	1	−25	35	−27	10	3	−8	−62	−22	−7
−2	−13	36	−32	−19	−31	4	−7	−6	−17	−25
1	34	5	−62	−67	−42	−9	−38	32	−10	15
9	−4	−6	42	−72	−48	5	24	−97	25	36

input: $(d_1, d_2) = (8, 16)$

kernel: $(d_1^{(1)}, d_2^{(1)}) = (3, 6)$

convolution: $(d_1 - d_1^{(1)} + 1, d_2 - d_2^{(1)} + 1) = (6, 11)$

discretized pattern

17	13	6	4	10	7	4	4	9	16	22	37	10	40	33	21
1	11	1	4	11	3	2	6	4	2	6	9	14	35	47	12
12	4	9	6	21	2	22	6	4	8	3	9	10	18	24	16
4	1	6	10	5	9	16	12	1	11	11	14	19	28	15	5
1	3	28	12	9	4	9	7	17	9	11	6	15	10	10	18
5	8	4	4	0	2	4	15	29	26	18	7	38	16	17	8
4	6	8	39	5	6	15	19	8	25	18	11	13	47	12	4
19	11	18	22	6	6	6	11	11	8	11	21	15	14	3	5

kernel

$*$

1 / 56	2 / 56	4 / 56	4 / 56	2 / 56	1 / 56
2 / 56	4 / 56	8 / 56	8 / 56	4 / 56	2 / 56
1 / 56	2 / 56	4 / 56	4 / 56	2 / 56	1 / 56

convolution

6.607	7.446	7.268	6.589	6.446	6.071	7.286	8.929	12.589	17.089	22
7.232	8.929	9.411	9.411	9.161	7.018	6.732	7.125	9.5	13.554	17.536
8.768	9.464	9.714	10.054	10.232	9	8.929	9.5	11.196	13.696	15.75
9.321	8.339	7.625	8.179	10	12.071	13.054	13.321	13.357	14.589	15.607
9.268	8.304	7.036	8.321	11.536	15.554	17.518	17.714	16.714	17.679	18.732
12.75	11.911	9.536	10	11.929	14.518	16.214	16.982	17.25	18.25	19.393

input: $(d_1, d_2) = (8, 16)$

kernel: $(d_1^{(1)}, d_2^{(1)}) = (3, 6)$

convolution: $(d_1 - d_1^{(1)} + 1, d_2 - d_2^{(1)} + 1) = (6, 11)$

First layer of feature maps

- convolution image: a first-order summary statistic

$$(\tilde{\mathbf{x}} * \mathbf{F}^{(1,k)})_{ij} = \sum_{\mathbf{u} \in \mathbf{x}} A_{ij}^{(k)}(\mathbf{u})$$

$$A_{ij}^{(k)}(\mathbf{u}) = \sum_{i'} \sum_{j'} \mathbb{1}[\mathbf{u} \in B_{i+i',j+j'}] F_{i'j'}^{(1,k)}$$

- output: feature map

$$H_{ij}^{(1,k)} = f_1\left(b_0^{(1,k)} + (\tilde{\mathbf{x}} * \mathbf{F}^{(1,k)})_{ij}\right) = f_1\left(b_0^{(1,k)} + \sum_{\mathbf{u} \in \mathbf{x}} A_{ij}^{(k)}(\mathbf{u})\right)$$

  - intercept (bias) term $b_0^{(1,k)} \in \mathbb{R}$
  - activation function $f_1 : \mathbb{R} \to \mathbb{R}$
  - allows for non-linearity

- first layer of feature maps

$$\mathbf{H}^{(1,k)} = \left[H_{ij}^{(1,k)}\right]_{(d_1 - d_1^{(1)}+1) \times (d_2 - d_2^{(1)}+1)}, \quad k = 1, \ldots, \ell_1$$

discretized pattern

17	13	6	4	10	7	4	4	9	16	22	37	10	40	33	21
1	11	1	4	11	3	2	6	4	2	6	9	14	35	47	12
12	4	9	6	21	2	22	6	4	8	3	9	10	18	24	16
4	1	6	10	5	9	16	12	1	11	11	14	19	28	15	5
1	3	28	12	9	4	9	7	17	9	11	6	15	10	10	18
5	8	4	4	0	2	4	15	29	26	18	7	38	16	17	8
4	6	8	39	5	6	15	19	8	25	18	11	13	47	12	4
19	11	18	22	6	6	6	11	11	8	11	21	15	14	3	5

$*$

kernel

−1	−1	−1
−1	8	−1
−1	−1	−1

$=$

convolution

25	−49	−36	31	−55	−38	−7	−23	−56	−58	−39	−56	84	177
−13	29	−19	118	−73	120	−19	−18	22	−46	−14	−66	−48	16
−59	−25	−16	−33	−16	57	14	−66	24	17	28	42	103	−9
−33	176	30	26	−22	3	−47	26	−52	−14	−85	−18	−78	−37
5	−76	−73	−81	−36	−45	12	106	73	31	−74	179	−34	11
−29	−48	245	−45	4	51	53	−80	71	17	−53	−65	248	−18

convolution

25	−49	−36	31	−55	−38	−7	−23	−56	−58	−39	−56	84	177
−13	29	−19	118	−73	120	−19	−18	22	−46	−14	−66	−48	16
−59	−25	−16	−33	−16	57	14	−66	24	17	28	42	103	−9
−33	176	30	26	−22	3	−47	26	−52	−14	−85	−18	−78	−37
5	−76	−73	−81	−36	−45	12	106	73	31	−74	179	−34	11
−29	−48	245	−45	4	51	53	−80	71	17	−53	−65	248	−18

$+$

5

bias

$=$

convolution + bias

30	−44	−31	36	−50	−33	−2	−18	−51	−53	−34	−51	89	182
−8	34	−14	123	−68	125	−14	−13	27	−41	−9	−61	−43	21
−54	−20	−11	−28	−11	62	19	−61	29	22	33	47	108	−4
−28	181	35	31	−17	8	−42	31	−47	−9	−80	−13	−73	−32
10	−71	−68	−76	−31	−40	17	111	78	36	−69	184	−29	16
−24	−43	250	−40	9	56	58	−75	76	22	−48	−60	253	−13

convolution + bias

30	−44	−31	36	−50	−33	−2	−18	−51	−53	−34	−51	89	182
−8	34	−14	123	−68	125	−14	−13	27	−41	−9	−61	−43	21
−54	−20	−11	−28	−11	62	19	−61	29	22	33	47	108	−4
−28	181	35	31	−17	8	−42	31	−47	−9	−80	−13	−73	−32
10	−71	−68	−76	−31	−40	17	111	78	36	−69	184	−29	16
−24	−43	250	−40	9	56	58	−75	76	22	−48	−60	253	−13

$\diamond$

activation

Relu

$=$

feature map

30	0	0	36	0	0	0	0	0	0	0	0	89	182
0	34	0	123	0	125	0	0	27	0	0	0	0	21
0	0	0	0	0	62	19	0	29	22	33	47	108	0
0	181	35	31	0	8	0	31	0	0	0	0	0	0
10	0	0	0	0	0	17	111	78	36	0	184	0	16
0	0	250	0	9	56	58	0	76	22	0	0	253	0

First layer of feature maps

- each element of matrices $\mathbf{H}^{(1,1)}, \ldots, \mathbf{H}^{(1,\ell_1)}$ contains some information about f_X

$$\mathbb{P}\left\{H_{ij}^{(1,k)} \leq z\right\} = \mathbb{P}\left\{f_1\left(b_0^{(1,k)} + \sum_{\mathbf{u}\in X} A_{ij}^{(k)}(\mathbf{u})\right) \leq z\right\}$$

$$= \sum_{n=0}^{\infty} \frac{\exp(-|W|)}{n!} \int_W \cdots \int_W$$

$$\mathbb{1}\left[f_1\left(b_0^{(1,k)} + \sum_{l=1}^{n} A_{ij}^{(k)}(\mathbf{u}_l)\right) \leq z\right]$$

$$f_X(\{\mathbf{u}_1, \ldots, \mathbf{u}_n\})\mathrm{d}\mathbf{u}_1 \cdots \mathrm{d}\mathbf{u}_n$$

- transforming point patterns to feature maps

$$\mathbf{x} \xrightarrow{\text{discretization}} \tilde{\mathbf{x}} \xrightarrow{\text{feature maps}} (\mathbf{H}^{(1,1)}, \ldots, \mathbf{H}^{(1,\ell_1)})$$

Pooling

- partition $\mathbf{H}^{(1,k)}$ into $p_1^{(1)} \times p_2^{(1)}$ block submatrices
- summarize entries of each submatrix into a single value by a pooling function (mean, sum, maximum)

$$\mathbf{H}^{(1,k)} \xrightarrow{\text{pooling}} \tilde{\mathbf{H}}^{(1,k)}$$

$$\tilde{\mathbf{H}}^{(1,k)} = \left[\tilde{H}_{ij}^{(1,k)}\right]_{\tilde{d}_1 \times \tilde{d}_2}$$

$$\tilde{d}_i^{(1)} = \lfloor (d_i - d_i^{(1)} + 1)/p_i^{(1)} \rfloor, \quad i = 1, 2$$

  - $\tilde{\mathbf{H}}^{(1,k)}$ more robust to small variations in $\tilde{\mathbf{x}}$ than $\mathbf{H}^{(1,k)}$
  - reduces the effect of discretization of $\mathbf{x}$

$$\mathbf{x} \xrightarrow{\text{discretization}} \tilde{\mathbf{x}} \xrightarrow{\text{feature maps}} (\mathbf{H}^{(1,1)}, \ldots, \mathbf{H}^{(1,\ell_1)}) \xrightarrow{\text{pooling}} (\tilde{\mathbf{H}}^{(1,1)}, \ldots, \tilde{\mathbf{H}}^{(1,\ell_1)})$$

discretized pattern

17	13	6	4	10	7	4	4	9	16	22	37	10	40	33	21
1	11	1	4	11	3	2	6	4	2	6	9	14	35	47	12
12	4	9	6	21	2	22	6	4	8	3	9	10	18	24	16
4	1	6	10	5	9	16	12	1	11	11	14	19	28	15	5
1	3	28	12	9	4	9	7	17	9	11	6	15	10	10	18
5	8	4	4	0	2	4	15	29	26	18	7	38	16	17	8
4	6	8	39	5	6	15	19	8	25	18	11	13	47	12	4
19	11	18	22	6	6	6	11	11	8	11	21	15	14	3	5

$*$

kernel

1	0	1
0	1	0
1	0	1

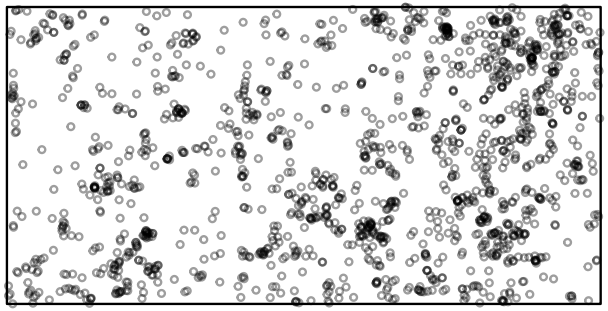
convolution

55	28	50	30	60	21	45	38	40	76	54	118	112	142
16	35	29	47	36	52	29	35	30	39	59	96	113	104
51	31	77	29	70	35	64	31	46	43	53	62	87	77
22	51	27	34	29	47	57	81	68	69	92	80	99	67
49	64	54	61	40	40	64	89	80	69	64	112	66	96
52	53	67	39	22	49	69	68	94	80	93	71	120	55

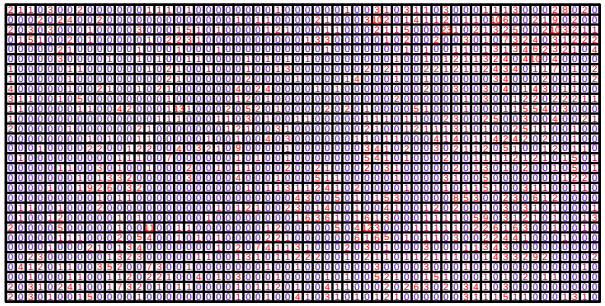
max pooling

55	50	60	45	76	118	142
51	77	70	81	69	92	99
64	67	49	89	94	112	120

observed point pattern



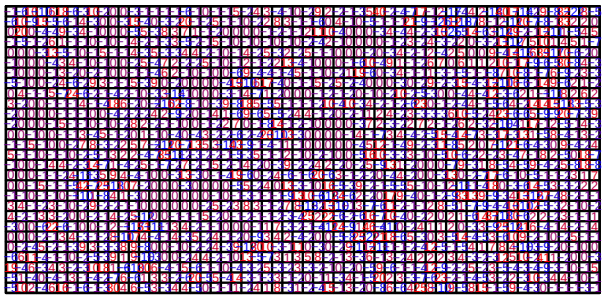
discretized pattern



kernel

0	−1	0
−1	5	−1
0	−1	0

convolution



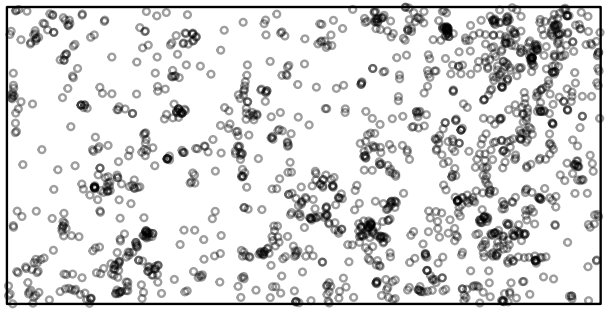
feature map

15	20	23	9	15	5	20	5	25	5	10	10	9	13	5	14	9	10	45	26	22	117	9	13	45	25	14	38	13
25	5	14	9	5	5	10	13	12	6	10	5	5	5	7	16	5	5	9	9	9	15	19	8	34	12	17	10	10
5	5	18	9	10	5	10	12	9	10	5	9	10	18	10	5	5	15	14	9	11	12	11	14	19	22	44	35	9
5	5	9	5	14	5	10	14	7	5	5	24	22	5	10	10	10	24	5	9	14	4	20	17	21	4	19	7	8
9	5	10	29	4	23	5	8	67	5	14	23	10	10	5	15	15	9	5	15	28	5	9	17	9	19	20	18	11
5	5	10	4	5	5	13	9	5	5	9	14	10	19	9	5	5	12	9	8	12	11	15	9	28	22	7	25	14
5	10	5	12	13	7	12	5	25	18	8	48	25	18	5	5	5	10	17	14	3	20	19	12	26	18	13	14	9
10	5	9	5	19	8	10	40	12	4	10	9	10	14	5	9	5	21	18	8	5	15	14	13	9	14	13	10	23
5	10	4	47	30	15	12	5	18	5	5	24	9	18	4	25	8	14	5	10	5	18	5	23	5	19	8	8	12
9	5	8	5	14	9	5	5	5	5	10	13	8	12	18	18	29	8	22	14	7	13	38	14	15	18	32	13	5
9	8	27	5	5	9	56	5	9	10	5	5	6	12	30	27	29	51	15	5	9	7	6	42	30	26	7	9	9
5	10	8	14	8	13	22	5	9	10	9	14	35	9	16	5	14	30	26	10	17	8	19	12	15	14	5	10	5
24	16	9	5	23	14	35	11	10	9	5	18	10	8	10	13	8	11	14	9	17	8	10	8	17	15	46	5	10
15	7	21	8	4	35	11	8	9	25	10	19	9	13	6	20	9	5	25	9	30	28	13	20	14	15	9	9	5

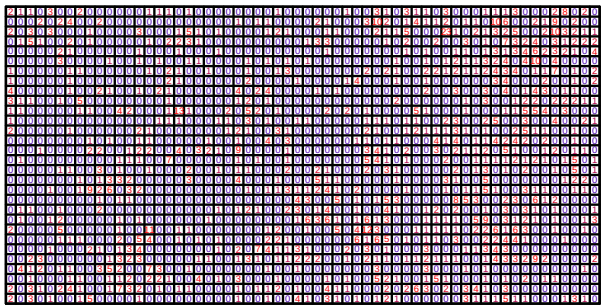
pooled feature map

25	23	15	20	25	10	13	16	10	45	117	19	45	38	0
5	18	14	14	10	24	22	10	24	14	14	20	22	44	0
9	29	23	13	67	23	19	15	15	15	28	17	28	25	0
10	12	19	40	25	48	25	9	21	18	20	19	26	14	0
10	47	30	12	18	24	18	25	29	22	18	38	19	32	0
10	27	13	56	10	14	35	30	51	26	17	42	30	10	0
24	21	35	35	25	19	13	20	11	25	30	20	17	46	0

observed point pattern



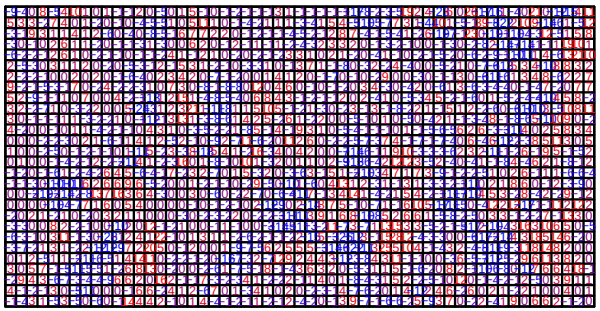
discretized pattern



kernel

0	1	0	0	−1	0
2	0	0	0	0	−2
0	1	0	0	−1	0

convolution



feature map

10	13	12	15	6	5	7	6	10	16	5	4	8	6	6	10	0	3	24	9	15	31	44	7	26	5	9	5
5	14	11	9	7	5	6	5	12	12	7	6	6	3	7	13	12	4	6	11	4	28	7	4	4	17	24	5
7	5	11	7	5	5	4	9	10	7	5	5	6	8	12	7	5	7	9	5	3	6	3	10	9	13	26	5
14	4	5	12	9	4	6	12	9	5	4	17	9	11	5	5	8	14	5	7	6	8	2	6	9	13	12	5
10	4	6	12	7	10	4	7	26	6	6	11	15	8	6	7	8	8	5	9	10	7	6	5	1	9	15	5
9	5	6	6	3	6	9	8	18	3	6	10	10	14	8	5	4	6	6	5	7	11	13	4	10	15	14	5
6	5	5	7	6	6	10	8	13	10	9	21	9	16	11	5	0	12	15	4	2	11	11	4	13	16	8	5
6	5	6	4	11	10	9	12	21	7	6	15	7	6	8	10	3	9	17	12	7	5	6	9	11	7	4	5
5	5	3	11	18	21	11	5	8	5	5	14	5	6	12	18	19	6	10	9	4	4	10	13	13	7	14	5
5	7	6	12	11	10	9	5	5	5	7	7	3	7	14	12	13	10	11	15	0	5	17	17	4	16	18	5
2	5	16	4	5	7	17	27	6	8	6	5	4	7	10	16	8	20	33	8	5	5	7	9	23	19	11	5
6	7	8	7	0	12	25	10	7	5	6	7	17	14	10	8	2	30	16	9	8	11	0	3	16	19	13	5
14	12	4	2	6	11	18	21	5	12	6	9	13	8	11	7	2	6	10	3	13	17	4	5	12	11	23	5
9	8	8	5	5	19	11	9	7	12	6	9	6	7	5	9	14	5	9	10	12	11	7	14	11	11	9	5

pooled feature map

14	15	7	7	16	7	8	13	12	24	31	44	26	24
14	12	9	12	10	17	11	12	14	9	8	10	13	26
10	12	10	9	26	11	15	8	8	9	11	13	15	15
6	7	11	12	21	21	16	11	12	17	11	11	16	8
7	12	21	11	8	14	7	18	19	15	5	17	16	18
7	16	12	27	8	7	17	16	30	33	11	9	23	13
14	8	19	21	12	9	13	11	14	10	17	14	12	23

More layers

- second layer of feature maps $\mathbf{H}^{(2,k')} = \left[H_{ij}^{(2,k')}\right]$

$$H_{ij}^{(2,k')} = f_2\left(b_0^{(2,k')} + \sum_k (\tilde{\mathbf{H}}^{(1,k)} * \mathbf{F}^{(2,k,k')})_{ij}\right), \quad k' = 1, \ldots, \ell_2$$

  - a second layer set of $d_1^{(2)} \times d_2^{(2)}$ kernels $\mathbf{F}^{(2,k,k')}$
- $\mathbf{H}^{(2,1)}, \ldots, \mathbf{H}^{(2,\ell_2)}$ contain some more detailed information about f_X

$$\mathbb{P}\left\{H_{ij}^{(2,k')} \leq z\right\} = \sum_{n=0}^{\infty} \frac{\exp(-|W|)}{n!} \int_W \cdots \int_W \mathbb{1}\left[f_2\left(b_0^{(2,k')} + \sum_k \sum_{i'} \sum_{j'} f_1\left(b_0^{(1,k)} + \sum_{l=1}^{n} A^{(k)}_{i+i'-1,j+j'-1}(\mathbf{u}_l)\right) F_{i'j'}^{(2,k,k')}\right) \leq z\right] f_X(\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}) \mathrm{d}\mathbf{u}_1 \cdots \mathrm{d}\mathbf{u}_n$$

- $p_1^{(2)} \times p_2^{(2)}$ pooling

$$(\tilde{\mathbf{H}}^{(1,1)}, \ldots, \tilde{\mathbf{H}}^{(1,\ell_1)}) \xrightarrow{\text{feature maps}} (\mathbf{H}^{(2,1)}, \ldots, \mathbf{H}^{(2,\ell_2)}) \xrightarrow{\text{pooling}} (\tilde{\mathbf{H}}^{(2,1)}, \ldots, \tilde{\mathbf{H}}^{(2,\ell_2)})$$

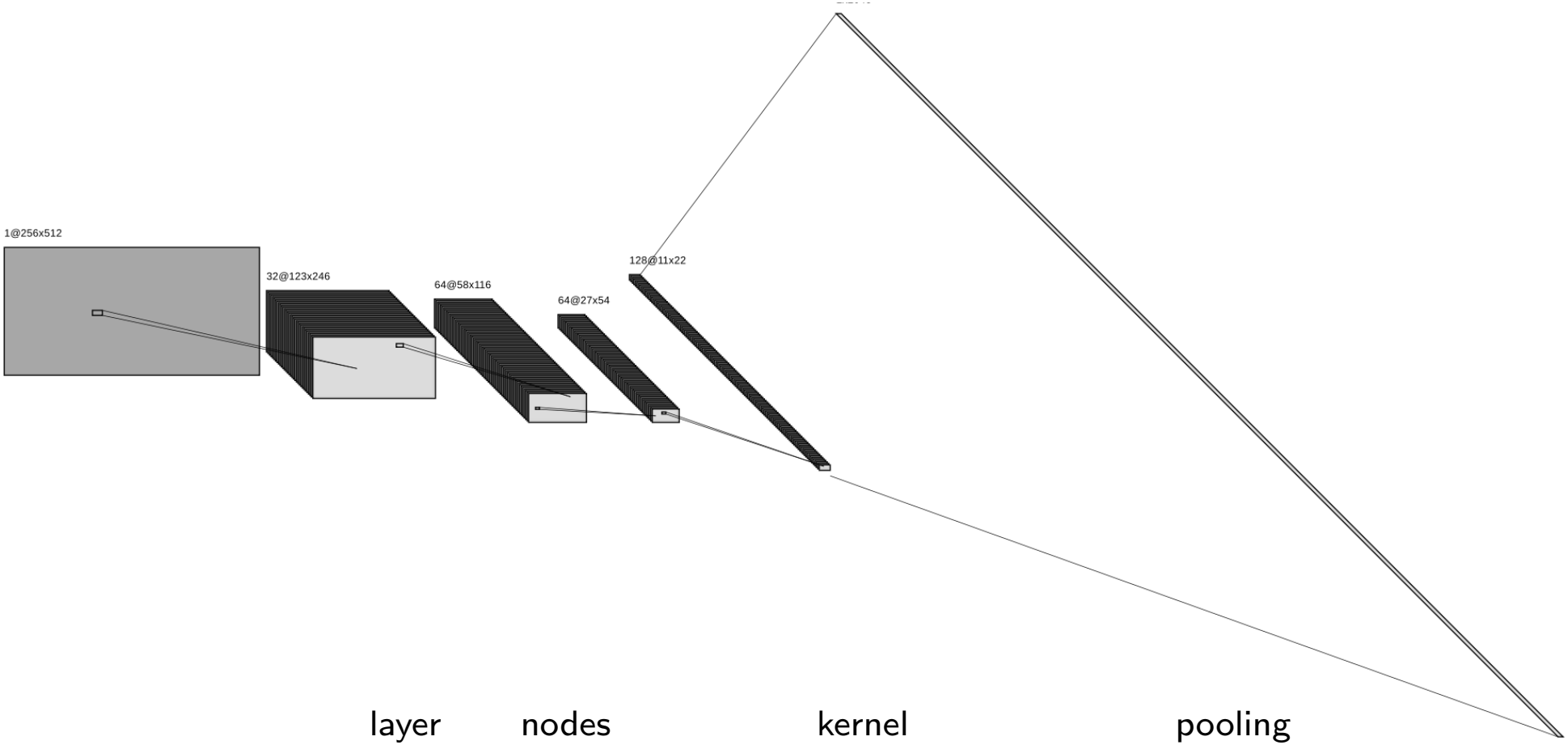
More layers

- deeper feature maps: more layers
  - convolution with kernels
  - adding biases and applying activation functions
  - performing pooling
- number of layers: $L \geq 2$
- given $(\mathbf{H}^{(L-1,1)}, \ldots, \mathbf{H}^{(L-1,\ell_{L-1})})$
- final feature vector $\mathbf{G} = [g_{k'}]$: a perceptron layer instead of pooling

$$g_{k'} = f_L \left(b_0^{(L,k')} + \sum_{i,j,k} H_{ij}^{(L-1,k)} w_{ij}^{(k,k')} \right), \quad k' = 1, \ldots, \ell_L$$

  - $b_0^{(L,k')}$ and $\mathbf{w}^{(k,k')} = \left[w_{ij}^{(k,k')} \right]$ are biases and weights

$$\mathbf{x} \xrightarrow{\text{network}} \mathbf{G} = (g_1, \ldots, g_{\ell_L})$$



layer	nodes	kernel	pooling
input:	1		
1st convolution:	$\ell_1 = 32$	$(d_1^{(1)}, d_2^{(1)}) = (10, 20)$	$(p_1^{(1)}, p_2^{(1)}) = (2, 2)$
2nd convolution:	$\ell_2 = 64$	$(d_1^{(2)}, d_2^{(2)}) = (7, 14)$	$(p_1^{(2)}, p_2^{(2)}) = (2, 2)$
3rd convolution:	$\ell_3 = 64$	$(d_1^{(3)}, d_2^{(3)}) = (4, 8)$	$(p_1^{(3)}, p_2^{(3)}) = (2, 2)$
4th convolution:	$\ell_4 = 128$	$(d_1^{(4)}, d_2^{(4)}) = (4, 8)$	
perceptron:	$\ell_5 = 2048$		

Single layer perceptron

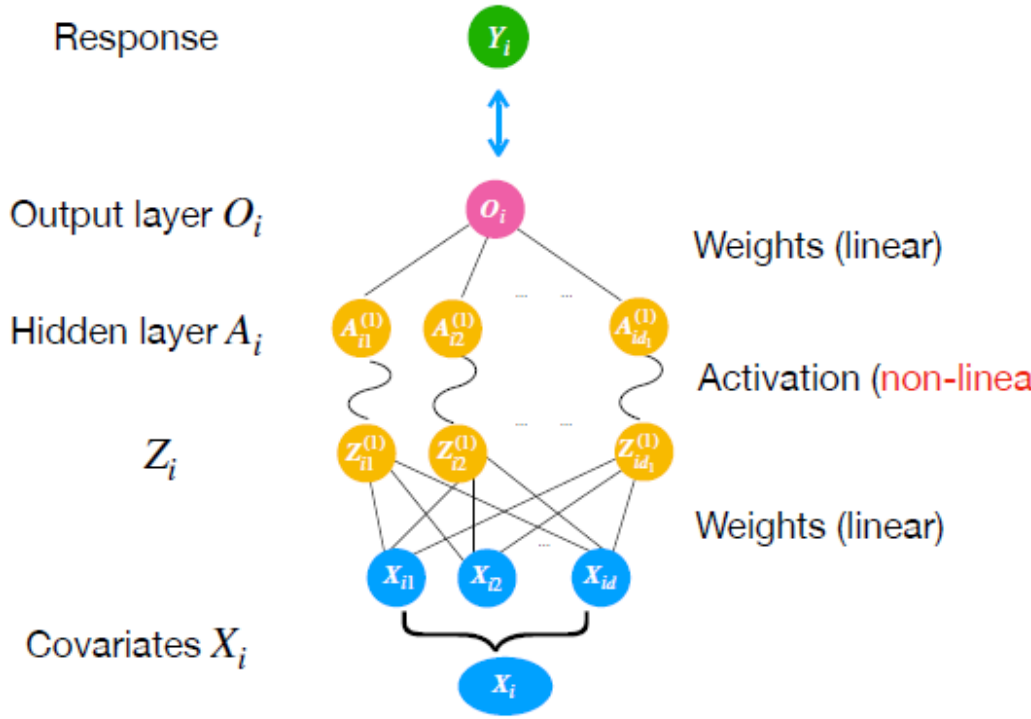
Neural networks

Neural network model: $E(Y_i) = m(X_i)$

Single layer perceptron:

$$m(X_i) = \beta^\top g_1(W_1 * X_i)$$

- Output layer $O_i = m(X_i)$ is fitted to the response Y_i to estimate the weights W_1 and β



Multi layer perceptron

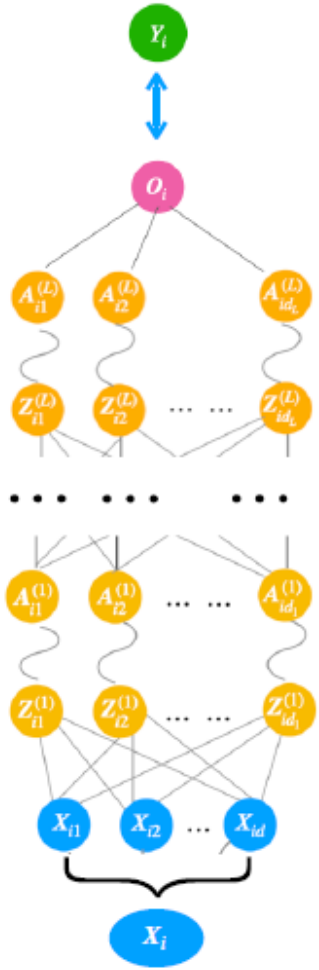
Neural networks

Neural network model: $E(Y_i) = m(X_i)$

Multi-layer perceptron (MLP):

$$m(X_i) = \beta^\top g_L(W_L * g_{L-1}(W_{L-1} * \dots g_1(W_1 * X_i)\dots)$$

- L hidden layers
- Output layer $O_i = m(X_i)$ is fitted to the response Y_i to estimate the wieghts $W_i, i = 1, \dots, L$ and β



Convolutional neural network

- vector of all network parameters

$$\boldsymbol{\vartheta} = \left(\{b_0^{l,k}\}_{l,k}, \{F_{ij}^{l,k,k'}\}_{l,k,k',i,j}, \{w_{ij}^{k,k'}\}_{k,k',i,j} \right)$$

- for every $x \in \mathbb{R}$, $0 \leq f_L(x) \leq 1$
- a very flexible and versatile summary statistic

$$G_{\boldsymbol{\vartheta}} : \mathcal{X} \to \mathcal{F} = [0,1]^{\ell_L}$$

  - transforms any observed point pattern $\mathbf{x}$ to its corresponding final feature vector $\mathbf{G}$

$$\mathbf{x} \to \tilde{\mathbf{x}} \to \underset{G_{\boldsymbol{\vartheta}}}{\left(\begin{array}{c} (\mathbf{H}^{(1,1)}, \ldots, \mathbf{H}^{(1,\ell_1)}) \xrightarrow{\text{pooling}} (\tilde{\mathbf{H}}^{(1,1)}, \ldots, \tilde{\mathbf{H}}^{(1,\ell_1)}) \\ (\mathbf{H}^{(2,1)}, \ldots, \mathbf{H}^{(2,\ell_2)}) \xrightarrow{\text{pooling}} (\tilde{\mathbf{H}}^{(2,1)}, \ldots, \tilde{\mathbf{H}}^{(2,\ell_2)}) \\ \vdots \\ \ldots \to (\mathbf{H}^{(L-1,1)}, \ldots, \mathbf{H}^{(L-1,\ell_{L-1})}) \end{array} \right)} \to \mathbf{G}$$

Kolmogorov representation theorem

- Suppose that $f : [0,1]^\ell \to \mathbb{R}$, $\ell \geq 2$, is a continuous function. Then there exists continuous functions $\phi^{(k)} : \mathbb{R} \to \mathbb{R}$ and $\varphi^{(k,k')} : \mathbb{R} \to \mathbb{R}$ such that

$$f(x_1, \ldots, x_\ell) = \sum_{k=0}^{2\ell} \phi^{(k)} \left(\sum_{k'=1}^{\ell} \varphi^{(k,k')}(x_{k'}) \right)$$

- Braun (2009): There exists a continuous function $\phi : \mathbb{R} \to \mathbb{R}$, a continuous and monotone function $\varphi : \mathbb{R} \to \mathbb{R}$ and constants $a, b_1, \ldots, b_\ell, c_0, \ldots, c_{2\ell} \in \mathbb{R}$ such that

$$f(x_1, \ldots, x_\ell) = \sum_{k=0}^{2\ell} \phi \left(\sum_{k'=1}^{\ell} b_{k'} \varphi(x_{k'} + ak) + c_k \right)$$

- searching for suitable summary statistic $G : \mathcal{X} \to \mathcal{F}$

$$G \approx G_{\boldsymbol{\vartheta}}$$

Discriminant model

- response

$$y(\mathbf{x}^{(s_1,t_1)}, \mathbf{x}^{(s_2,t_2)}) = \begin{cases} 1 & s_1 = s_2 \\ 0 & s_1 \neq s_2 \end{cases}$$

- $y(\mathbf{x}^{(s_1,t_1)}, \mathbf{x}^{(s_2,t_2)})$: an observation of a Bernoulli random variable $Y(\mathbf{x}^{(s_1,t_1)}, \mathbf{x}^{(s_2,t_2)})$ with

$$p(\mathbf{x}^{(s_1,t_1)}, \mathbf{x}^{(s_2,t_2)}) = \mathbb{P}\{Y(\mathbf{x}^{(s_1,t_1)}, \mathbf{x}^{(s_2,t_2)}) = 1\}$$

- a general (nonlinear) discriminant model

$$p : \mathcal{X} \times \mathcal{X} \to [0, 1]$$

- convert to a dissimilarity function

$$D(\mathbf{x}, \mathbf{x}') = 1 - p(\mathbf{x}, \mathbf{x}')$$

Discriminant analysis

- given $\mathcal{D} = \{\mathbf{x}^{(s_i,t_i)} : i = 1, \ldots, N\}$
- good discriminant model

$$p(\mathbf{x}^{(s_1,t_1)}, \mathbf{x}^{(s_2,t_2)}) = \begin{cases} \approx 1 & s_1 = s_2 \\ \\ \approx 0 & s_1 \neq s_2 \end{cases}$$

- new observation $\mathbf{x}^{(*)} \notin \mathcal{D}$
  - for each $i = 1, \ldots, N$, $y(\mathbf{x}^{(*)}, \mathbf{x}^{(s_i,t_i)})$ is unknown
  - there exits an unknown $s^* \in \{1, \ldots, m\}$ such that

$$y(\mathbf{x}^{(*)}, \mathbf{x}^{(s_i,t_i)}) = \begin{cases} 1 & s_i = s^* \\ 0 & s_i \neq s^* \end{cases}$$

- detect the group membership of the query data $\mathbf{x}^{(*)}$ based on the support set $\mathcal{D}$

$$s^* = ?$$

Discriminant analysis

- one approach: 1-NN , first nearest neighbor classification

$$\widehat{s^*} = s_{\widehat{i}}, \quad \widehat{i} = \arg \max_{1 \leq i \leq N} p(\mathbf{x}^{(*)}, \mathbf{x}^{(s_i,t_i)})$$

- corresponding classifier

$$\widehat{y}(\mathbf{x}^{(*)}, \mathbf{x}^{(s_i,t_i)}) = \begin{cases} 1 & s_i = \widehat{s^*} \\ 0 & s_i \neq \widehat{s^*} \end{cases}$$

- misclassification error

$$\max_{\mathbf{x} \in \mathcal{D}} \left| y(\mathbf{x}^{(*)}, \mathbf{x}) - \widehat{y}(\mathbf{x}^{(*)}, \mathbf{x}) \right|$$

one-shot learning task (a measure the the accuracy of our classifier)

- randomly select a query point pattern

$$\mathbf{x}^{(*)} \in \mathcal{D}$$

- randomly select a support set

$$\mathcal{S} \subset \mathcal{D} \backslash \{\mathbf{x}^{(*)}\}$$

  - consisting of $2 \leq M \leq m$ point patterns from different groups
  - exactly one of them is from the same group as $\mathbf{x}^{(*)}$
- use the discriminant model, perform the classification

$$\widehat{y}(\mathbf{x}^{(*)}, \mathbf{x}), \mathbf{x} \in \mathcal{S}$$

- repeat the one-shot learning task n_{try} times

$$\text{ACC} = \frac{1}{n_{\text{try}}} \sum_{l=1}^{n_{\text{try}}} \left(1 - \max_{\mathbf{x} \in \mathcal{S}_l} \left| y(\mathbf{x}_l^{(*)}, \mathbf{x}) - \widehat{y}(\mathbf{x}_l^{(*)}, \mathbf{x}) \right| \right).$$

Siamese neural network (to obtain the discriminant model p)

- for any $\mathbf{x}, \mathbf{x}' \in \mathcal{D}$, compare their extracted features

$$\left.\begin{array}{rcl} \mathbf{x} & \rightarrow & G_{\boldsymbol{\vartheta}}(\mathbf{x}) \\ \mathbf{x}' & \rightarrow & G_{\boldsymbol{\vartheta}}(\mathbf{x}') \end{array}\right\} \rightarrow G_{\boldsymbol{\vartheta}}(\mathbf{x}) - G_{\boldsymbol{\vartheta}}(\mathbf{x}')$$

- discriminant model

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = f_{L+1}\left(\beta_0 + \sum_{k=1}^{\ell_L} \beta_k \left|\left[G_{\boldsymbol{\vartheta}}(\mathbf{x}) - G_{\boldsymbol{\vartheta}}(\mathbf{x}')\right]_k\right|\right),$$

- activation function

$$f_{L+1} : \mathbb{R} \rightarrow [0, 1]$$

- extended parameter vector

$$\boldsymbol{\theta} = (\boldsymbol{\vartheta}, \beta_0, \beta_1, \ldots, \beta_{\ell_L})$$

Hyper parameters of the network

- dimensions of the discretization grid (d_1, d_2)

$$\mathbf{x} \quad \rightarrow \quad \tilde{\mathbf{x}}$$

- most efficient activation functions: difficult and problem specific
  - $f_1, \ldots, f_{L-1}$: ReLU activation function $f(x) = \max(0, x)$
  - f_L and f_{L+1}: logistic activation function $f(x) = \exp(x)/(1 + \exp(x))$
- efficient architecture of the network: heuristic and problem specific
  - number of layers L
  - number of nodes at each layer $\ell_1, \ldots, \ell_L$
  - pooling dimension and function for each convolutional layer $(p_1^{(1)}, p_2^{(1)}), \ldots, (p_1^{(L-1)}, p_2^{(L-1)})$

Parameter estimation

- split the observed point patterns $\mathcal{D}$ into a training and a validation set

$$\mathcal{D}_{\text{train}} = \{\mathbf{x}^{(s,t)} : s = 1, \ldots, m, t \in \{1, \ldots, T\} \backslash \mathcal{J}^{(s)}\}$$
$$\mathcal{D}_{\text{valid}} = \{\mathbf{x}^{(s,t)} : s = 1, \ldots, m, t \in \mathcal{J}^{(s)}\}$$

  - $\mathcal{J}^{(s)} = \{J_1^{(s)}, \ldots, J_{T_{\text{valid}}}^{(s)}\}$: a random sample from $\{1, \ldots, T\}$ without replacement
  - $\mathcal{D}_{\text{valid}}$ has T_{valid} replicates for each group
  - $\mathcal{D}_{\text{train}}$ has $T - T_{\text{valid}}$ replicates for each group
- for example

$$\frac{T_{\text{valid}}}{T} \approx 0.3$$

- estimate $\boldsymbol{\theta}$ by maximizing the Bernoulli composite log-likelihood

$$l(\boldsymbol{\theta}; \mathcal{D}_{\text{train}}) = \sum_{\{\mathbf{x}, \mathbf{x}'\} \subset \mathcal{D}_{\text{train}}} \left\{ y(\mathbf{x}, \mathbf{x}') \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')}{1 - p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')} + \log \left[1 - p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')\right] \right\}$$

Over-fitting

- parameter vector $\boldsymbol{\theta}$ has a large number of elements
- over-fitting is a serious problem in such networks
  - the discriminant model works perfect for the observed point patterns
  - the discriminant model performs poorly for new replicates of the groups

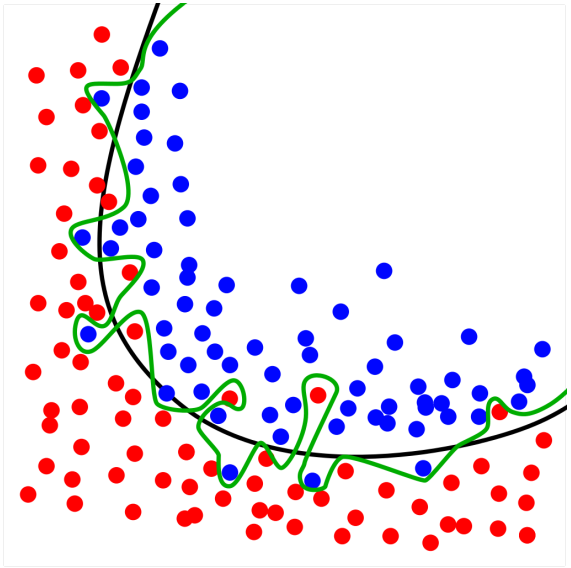


image source: Wikipedia

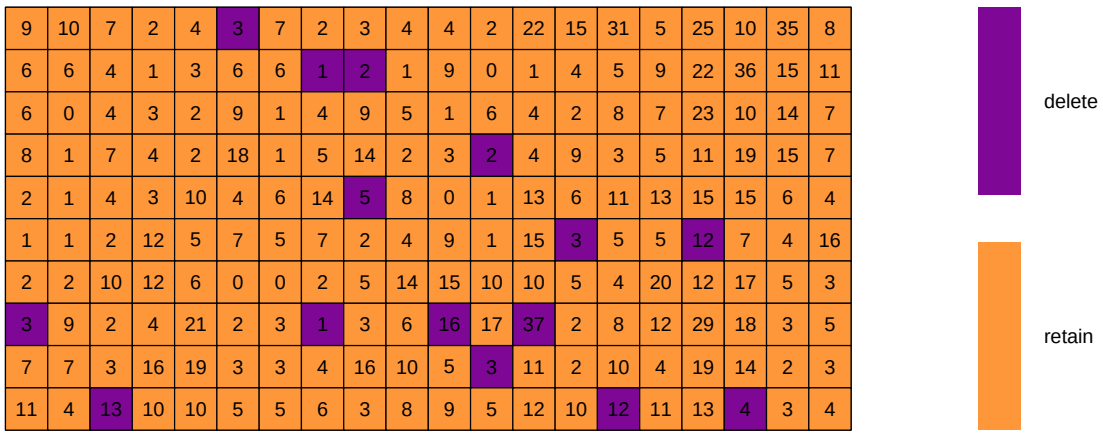
- strategies to prevent over-fitting
  - parameter regularization
  - dropout

Over-fitting

- parameter regularization

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\theta} \{l(\boldsymbol{\theta}, \mathcal{D}_{\text{train}}) - q\Omega(\boldsymbol{\theta})\}$$

  - penalty term $\Omega(\boldsymbol{\theta})$
    - ★ L^2-norm of filters and weights in all layers
  - penalty strength $q \geq 0$
    - ★ equivalent to $\|\boldsymbol{\theta}\|^2 \leq \delta$ for some $\delta > 0$
- dropout
  - each unit in a given layer of network is retained with a fixed probability p_{retian} independent of other units
  - similar to random independent thinning



Evaluation of the model

- substitute $\boldsymbol{\theta}$ with the final $\widehat{\boldsymbol{\theta}}$
- classify any given query data $\mathbf{x}^{(*)}$ using $p_{\widehat{\boldsymbol{\theta}}}$

$$\widehat{y}(\mathbf{x}^{(*)}, \mathbf{x}) = \widehat{y}_{\widehat{\boldsymbol{\theta}}}(\mathbf{x}^{(*)}, \mathbf{x})$$

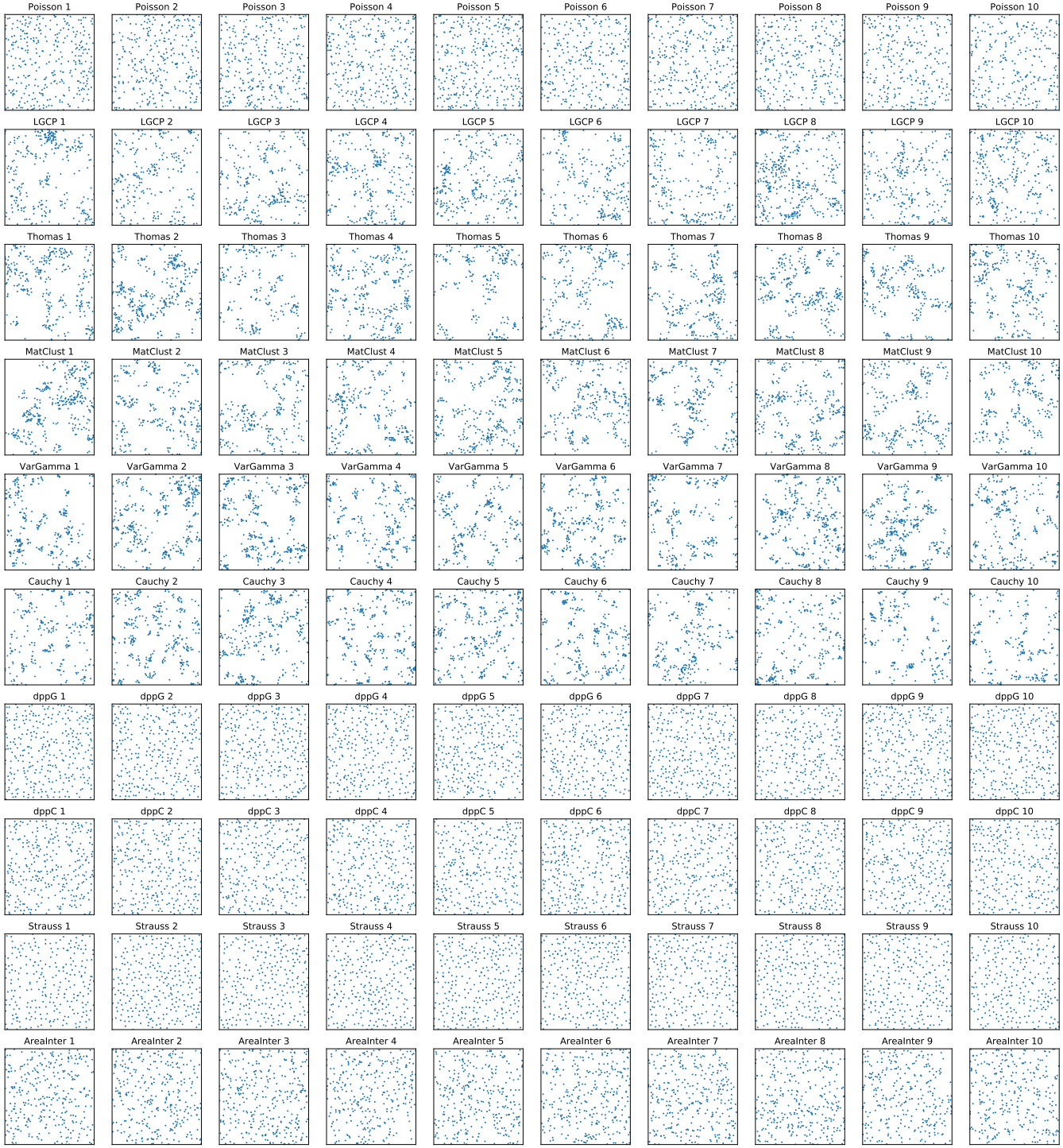
- perform one-shot learning task with support sets $\mathcal{S}$ of different size M
- estimate the accuracy of $\widehat{y}(\mathbf{x}^{(*)}, \mathbf{x})$ for both $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{valid}}$
- generalizability of the discriminant model: accuracy of the classifier on point patterns in $\mathcal{D}_{\text{valid}}$

Simulated point patterns

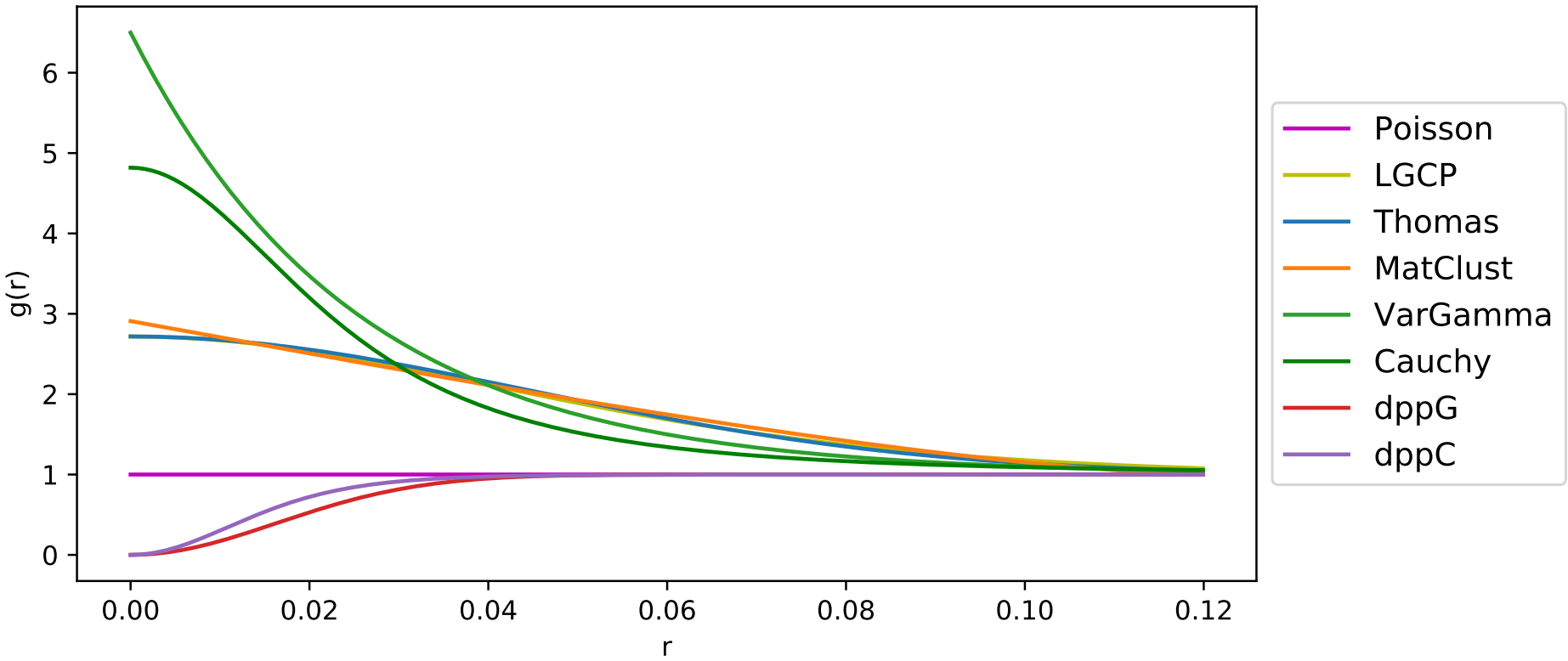
- $T = 10$ realizations from $m = 10$ stationary planar point processes
- constant intensity function $\rho(\mathbf{u}) = 300$ on $W = [0, 1]^2$
- processes
  - Poisson process
  - log-Gaussian Cox process (LGCP)
    - ★ Gaussian random field with covariance function $\exp(-(13.45r)^2)$
  - Thomas process
    - ★ parent intensity $\kappa = \frac{1}{0.004\pi(e-1)}$ and kernel variance $\omega^2 = 0.001$
  - Matérn cluster (MatClust) process
    - ★ parent intensity $\kappa = \frac{1}{0.004\pi(e-1)}$ and cluster radius $R = 0.06$
  - Variance Gamma (VarGamma) process
    - ★ parent intensity $\kappa = \frac{1}{0.004\pi(e-1)}$, shape $\nu = -1/4$ and scale $\omega = 0.025$
  - Cauchy process
    - ★ parent intensity $\kappa = \frac{1}{0.004\pi(e-1)}$ and scale $\omega = 0.015$

Simulated point patterns

- $T = 10$ realizations from $m = 10$ stationary planar point processes
- constant intensity function $\rho(\mathbf{u}) = 300$ on $W = [0, 1]^2$
- processes
  - Gaussian determinantal point process (dppG)
    - ★ scale parameter $\alpha = 0.03257$
  - Cauchy determinantal point process (dppC)
    - ★ shape parameter $\nu = 1$ and scale parameter $\alpha = 0.03257$
  - Strauss process
    - ★ interaction parameter $\gamma = 0.1$ and interaction radius $R = 0.035$
  - area-interaction (AreaInter) process
    - ★ canonical interaction parameter $\eta = 2$ and disc radius $r = 0.07$ $R = 0.035$
    - ★ $\eta = 2 > 1$ implies weak aggregating interactions
    - ★ its realizations are similar to realizations of a Poisson process ($\eta = 1$)


Poisson 1
Poisson 2
Poisson 3
Poisson 4
Poisson 5
Poisson 6
Poisson 7
Poisson 8
Poisson 9
Poisson 10
LGCP 1
LGCP 2
LGCP 3
LGCP 4
LGCP 5
LGCP 6
LGCP 7
LGCP 8
LGCP 9
LGCP 10
Thomas 1
Thomas 2
Thomas 3
Thomas 4
Thomas 5
Thomas 6
Thomas 7
Thomas 8
Thomas 9
Thomas 10
MatClust 1
MatClust 2
MatClust 3
MatClust 4
MatClust 5
MatClust 6
MatClust 7
MatClust 8
MatClust 9
MatClust 10
VarGamma 1
VarGamma 2
VarGamma 3
VarGamma 4
VarGamma 5
VarGamma 6
VarGamma 7
VarGamma 8
VarGamma 9
VarGamma 10
Cauchy 1
Cauchy 2
Cauchy 3
Cauchy 4
Cauchy 5
Cauchy 6
Cauchy 7
Cauchy 8
Cauchy 9
Cauchy 10
dppG 1
dppG 2
dppG 3
dppG 4
dppG 5
dppG 6
dppG 7
dppG 8
dppG 9
dppG 10
dppC 1
dppC 2
dppC 3
dppC 4
dppC 5
dppC 6
dppC 7
dppC 8
dppC 9
dppC 10
Strauss 1
Strauss 2
Strauss 3
Strauss 4
Strauss 5
Strauss 6
Strauss 7
Strauss 8
Strauss 9
Strauss 10
AreaInter 1
AreaInter 2
AreaInter 3
AreaInter 4
AreaInter 5
AreaInter 6
AreaInter 7
AreaInter 8
AreaInter 9
AreaInter 10

Simulated point patterns



- $g(r)$ of the LGCP and Thomas processes are nearly identical
- parameter settings of dppG and dppC result in very similar kernels, their corresponding processes produce similar point patterns

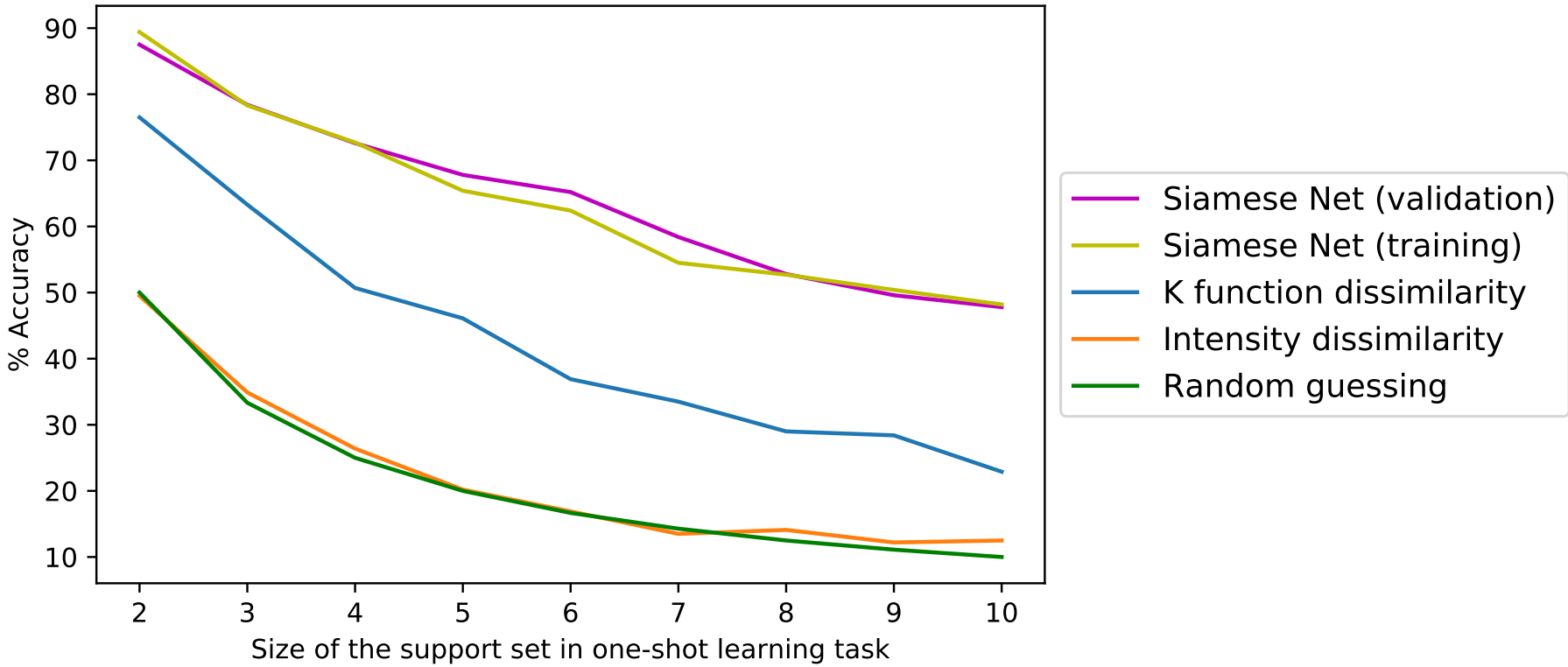
Discriminant model for the simulated point patterns

- set aside $T_{\text{valid}} = 3$ replicates for validation
- discretization $(d_1, d_2) = (128, 128)$
- network design

layer	nodes	dimensions	regularization	dropout
input:	1	$(d_1, d_2) = (128, 128)$	$\delta = 0.0025$	$p_{\text{retain}} = 0.9$
1st:	$\ell_1 = 8$	$(d_1^{(1)}, d_2^{(1)}) = (9, 9), (p_1^{(1)}, p_2^{(1)}) = (3, 3)$	$\delta = 0.0025$	$p_{\text{retain}} = 0.9$
2nd :	$\ell_2 = 16$	$(d_1^{(2)}, d_2^{(2)}) = (5, 5), (p_1^{(2)}, p_2^{(2)}) = (3, 3)$	$\delta = 0.0025$	$p_{\text{retain}} = 1$
3rd :	$\ell_3 = 32$	$(d_1^{(3)}, d_2^{(3)}) = (3, 3), (p_1^{(3)}, p_2^{(3)}) = (2, 2)$	$\delta = 0.0025$	$p_{\text{retain}} = 1$
final:	$\ell_4 = 256$		$\delta = 0.001$	$p_{\text{retain}} = 1$

- network parameter $\boldsymbol{\theta}$, with 213825 components
- estimated accuracy for $M = 2, \ldots, 10$ and $n_{\text{try}} = 1000$

Discriminant model for the simulated point pattern



- intensity function dissimilarity: as good as random guessing
- K function dissimilarity: $\geq 30\%$ more accurate than random guessing
- Siamese network: $\geq 10\%$ more accurate than K function dissimilarity
  - effective in distinguishing structural differences among the processes
  - generalizable to new unobserved replicates from these processes

BCI data set

- mean number of trees per census
  - between 100 and 700 for 66 species
  - between 700 and 1500 for 32 species
  - between 1500 and 3000 for 18 species
  - between 3000 to 7000 for 8 species
  - more than 7000 for 6 species
- dissimilarity in spatial inhomogeneities of species is more definite than dissimilarity in tree to tree interactions
- ecological hypotheses: large scale variations (inhomogeneity) in diverse rainforests are more prominent in the spatial forest structure than small scale variations (tree to tree interactions)

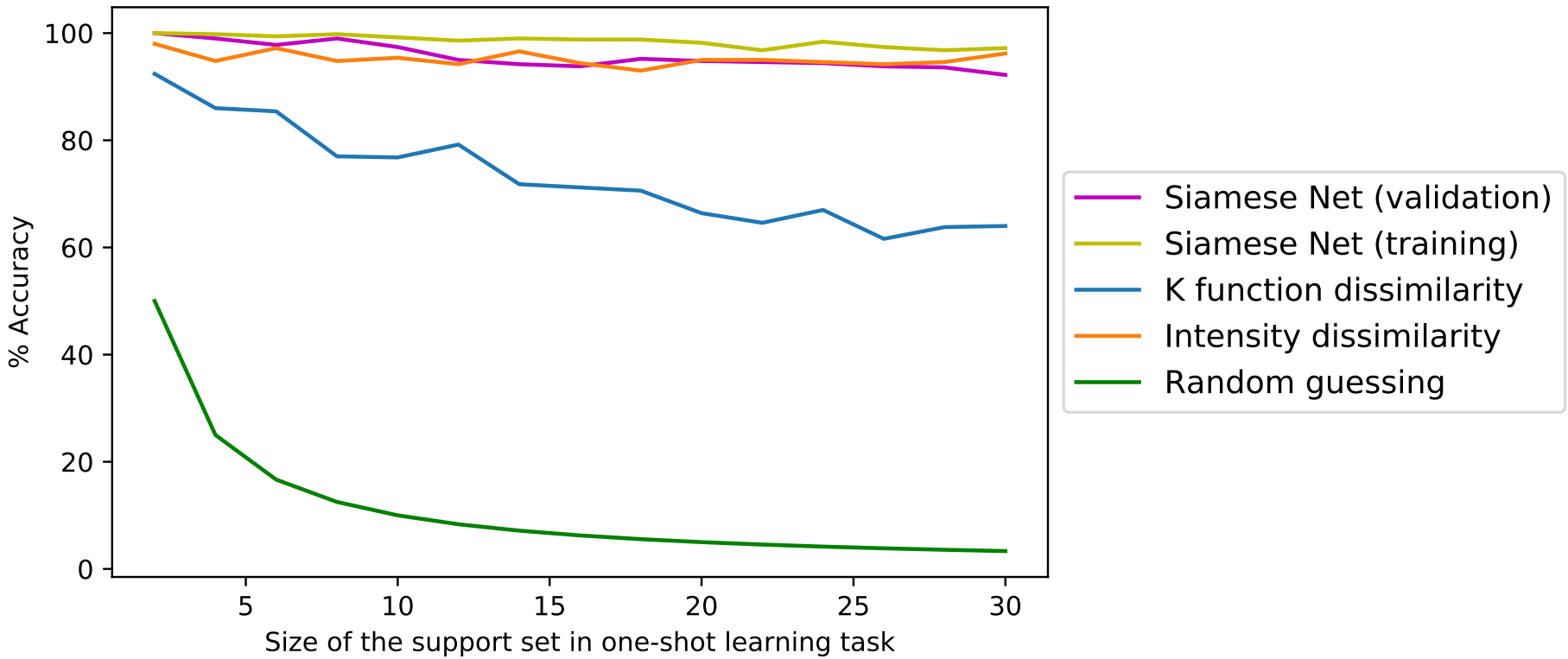
BCI data set

- $T_{\text{valid}} = 2$
- discretization $(d_1, d_2) = (256, 512)$
- network design

layer	nodes	dimensions	regularization	dropout
input:	1	$(d_1, d_2) = (256, 512)$	$\delta = 10^{-4}$	$p_{\text{retain}} = 0.9$
1st:	$\ell_1 = 64$	$(d_1^{(1)}, d_2^{(1)}) = (20, 10), (p_1^{(1)}, p_2^{(1)}) = (2, 2)$	$\delta = 10^{-4}$	$p_{\text{retain}} = 0.9$
2nd :	$\ell_2 = 128$	$(d_1^{(2)}, d_2^{(2)}) = (7, 14), (p_1^{(2)}, p_2^{(2)}) = (2, 2)$	$\delta = 10^{-4}$	$p_{\text{retain}} = 1$
3rd :	$\ell_3 = 128$	$(d_1^{(3)}, d_2^{(3)}) = (4, 8), (p_1^{(3)}, p_2^{(3)}) = (2, 2)$	$\delta = 10^{-4}$	$p_{\text{retain}} = 1$
4th :	$\ell_4 = 256$	$(d_1^{(4)}, d_2^{(4)}) = (4, 8), (p_1^{(4)}, p_2^{(4)}) = (2, 2)$	$\delta = 10^{-4}$	$p_{\text{retain}} = 1$
output:	$\ell_5 = 2048$		$\delta = 0.001$	$p_{\text{retain}} = 1$

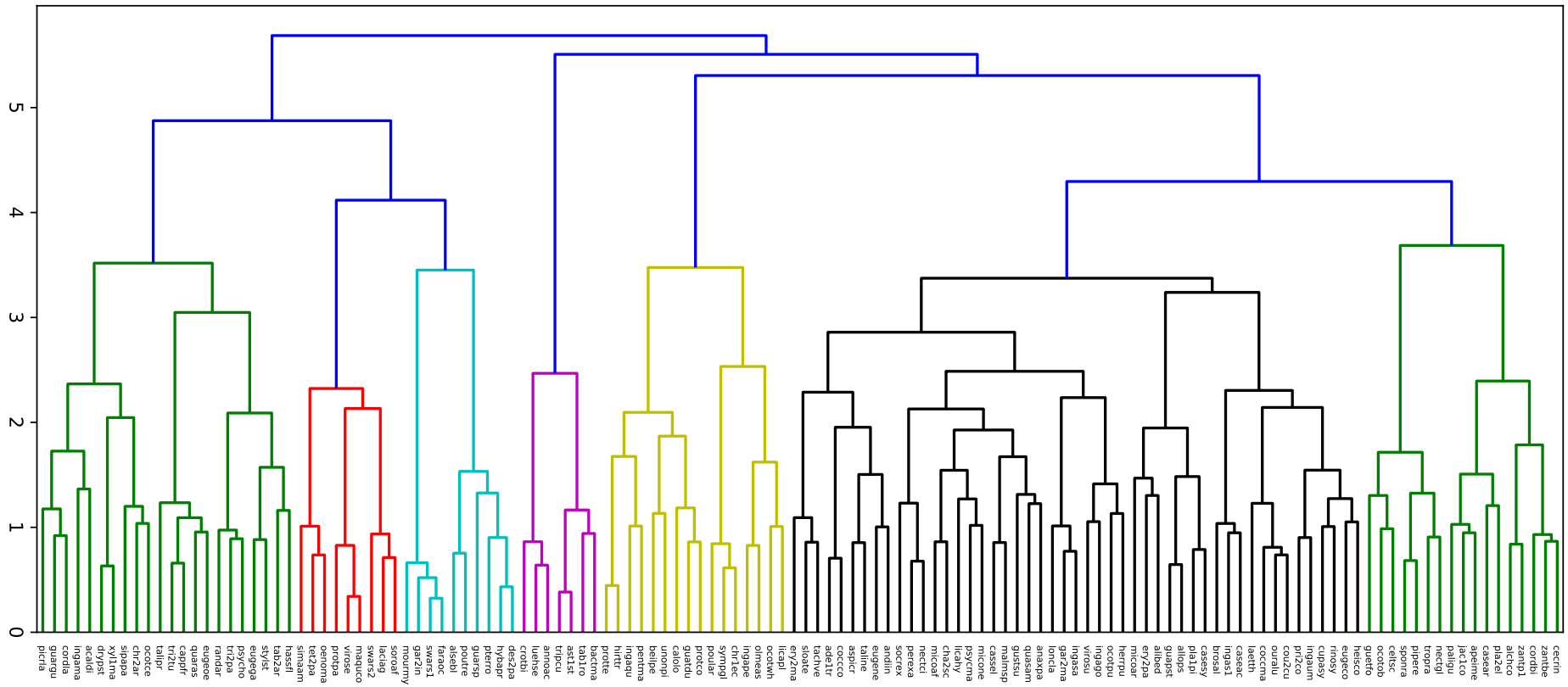
- network parameter $\boldsymbol{\theta}$, 72956449 components
- estimated accuracy for $M = 2, 4, \ldots, 30$ and $n_{\text{try}} = 1000$

BCI data set



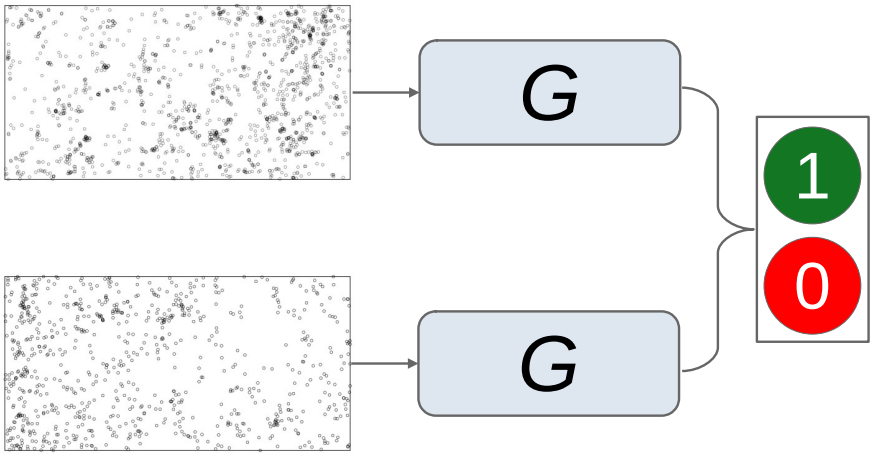
- intensity function dissimilarity: as good as Siamese network classifier
- K function dissimilarity: inferior
- better performance by tweaking the network architecture and tuning parameters

BCI data set



- the model distinguishes seven distinct clusters of species

Conclusions

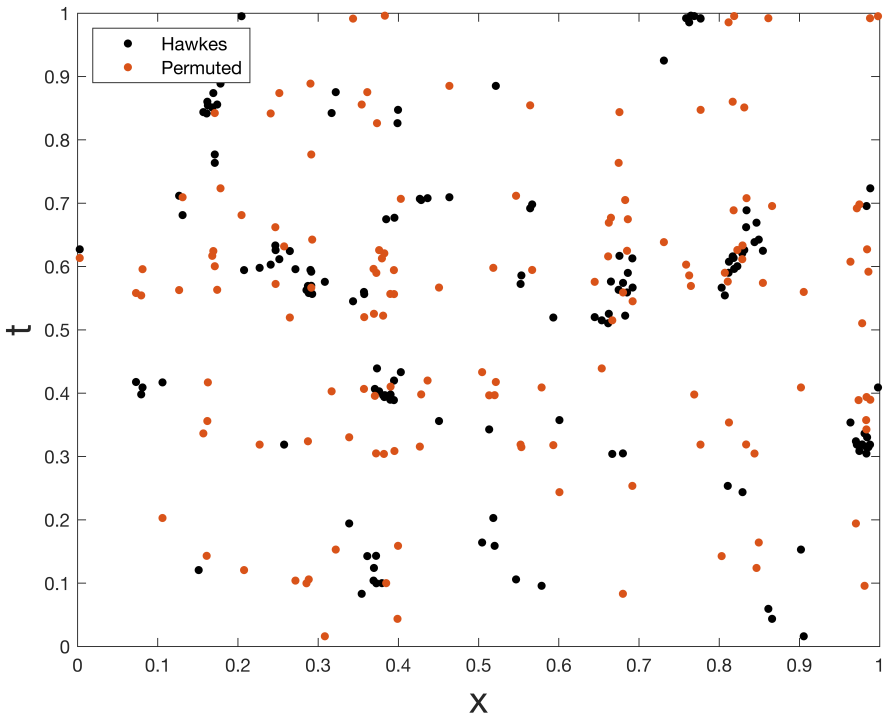


- which summary statistic?
  - interpretation and explanation
    - ★ usual point process summary statistics
  - prediction and discrimination
    - ★ neural networks
- computations
  - Siamese network: `Python` library `Keras` and `TensorFlow`
  - intensity and K functions: `R` package `spatstat`
  - code is available on: `https://github.com/jalilian/SiameseNetPPP`

Permutations for point processes

- Problem of testing for interactions between two spatial-temporal point processes
- In a two sample randomization test, the event times of a point process are permuted, while the spatial coordinates of the process are fixed
- This "permuted point process" provides a counterfactual that can be compared to the second point process for which an interaction is being assessed
- However, the permuted point process may not have the same statistics as the original data

Permutations for point processes



Simulated Hawkes process in black (first spatial coordinate vs. time) along with the same Hawkes process with times randomly permuted in red. Whereas the Hawkes process exhibits significant space-time clustering, the permuted process only exhibits spatial clustering.

Permutations for point processes

- Goal: generate permutations of a point process that preserve second order statistics of the process
- Let $\mathbf{z}_i = (\mathbf{x}_i, t_i)_{i=1}^N$ denote the spatial coordinates and event times of a point process and $\widetilde{t}_i$ be a random permutation of the event times.
- Second order statistics

$$K(r) = \frac{1}{N^2} \sum_{i,j} 1\{\|\mathbf{z}_i - \mathbf{z}_j\| < r\} \tag{1}$$

  and the L function, given by $L(r) = [K(r)]^{1/2}$.
- We generate permutations such that the L function, $L(r)$, of the data better matches the L function, $\widetilde{L}(r)$, of the permuted point process
- Goal: is to generate multiple permuted point processes such that the distribution of the L functions closely matches the distribution of the L function of the data.

Permutations for point processes

- The algorithm proceeds in two stages.
- First stage, M independent random permutations $\tilde{\mathbf{z}}^k = (\mathbf{x}_i, \tilde{t}_i^k)$, $k = 1, ..., M$ of the data are generated and the L function $L^k(r)$ is computed for each permutation. Next, the mean over the M L functions is calculated,

$$\mu(r) = \frac{1}{M} \sum_{k=1}^{M} L^k(r), \tag{2}$$

along with the error,

$$\epsilon^k(r) = L^k(r) - \mu(r). \tag{3}$$

Permutations for point processes

- Second stage, we generate M second order preserving (SOP) permutations. For each k, we initialize the permutation with the random permutation $\tilde{\mathbf{z}}^k$ from stage one.

- We then iteratively swap two random times to generate a proposal permutation $\tilde{\mathbf{q}}^k$. The L function, $L^{prop}(r)$, of this proposal permutation is then computed, along with the proposal error,

$$error_{prop} = \Big(\int |L^{prop}(r) - L^{data}(r) - \epsilon^k(r)|^2 dr \Big)^{1/2}. \qquad (4)$$

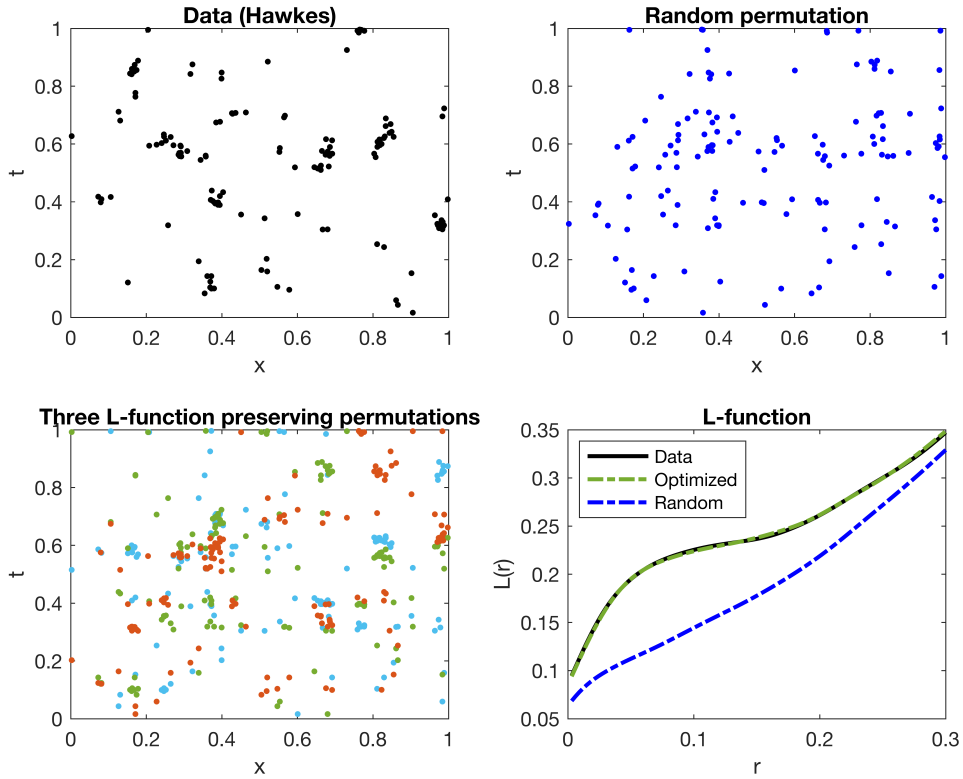
Permutations for point processes

- Generate data from a self-exciting Hawkes point process with intensity,

$$\lambda(x,t) = \mu + \sum_{t>t_i} \theta f(t-t_i;\omega) g(x-x_i;\sigma). \tag{5}$$

- We simulate the process on the unit cube in space-time with background rate $\mu = 40$, reproduction number $\theta = 0.75$, exponential kernel f in time with parameter $\omega = 100$ (mean 0.01) and Gaussian kernel g in space (2d) with standard deviation $\sigma = 0.01$.

Permutations for point processes



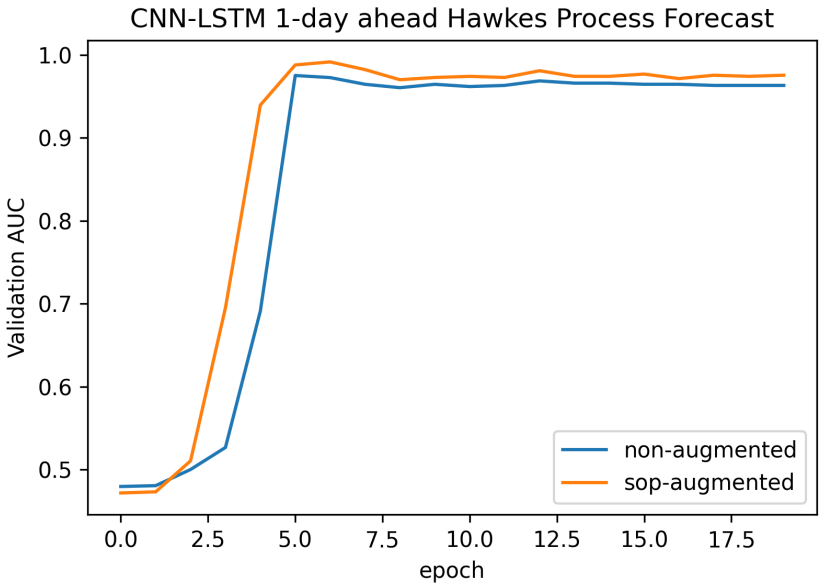
Top Left: Realization of the Hawkes process (first spatial coordinate vs. time). See text for the parameters used. Top right: example random permutation. Lower left: Three example SOP permutations of the Hawkes process. Lower Right: L function of data vs. L functions of SOP and random permutations.

Data augmentation for CNN-LSTM 1-day ahead point process forecast

- A number of neural network based models for space-time point processes have been introduced recently
- These models can improve accuracy of forecasts over simpler parametric models, but typically require larger datasets due to the increased number of parameters.
- We show how SOP permutations can be used to improve model performance on held out data through augmentation of training data.
- We generate data from a space-time Hawkes process defined on $[0,1] \times [0,1] \times [0,730]$. Next we discretize space into 25×25 grid cells and time into 1 "day" intervals from 0 to 730. We then use a sliding window and create $14 \times 25 \times 25$ features consisting of a binary indicator in each space-time cell for whether at least one event occurred $(y = 1)$ or no events occurred $(y = 0)$.

Data augmentation for CNN-LSTM 1-day ahead point process forecast

- Use this feature as input to a CNN-LSTM to predict whether an event will occur or not in each grid cell in the following day.
- First we train the model with no data augmentation.
- We then use SOP permutations to increase the data set size by a factor of 10.



AUC on held-out data versus training epochs for a CNN-LSTM forecast using sop-augmented training and non-augmented training.

Some further food for thought

- Point process data are becoming ubiquitous in modern applications
- A type of deep model for point process data is based on representing the influence kernel (rather than the intensity function) by neural networks.
- We can then develop a deep non-stationary influence kernel that can model non-stationary spatio-temporal point processes.
- The main idea is to approximate the influence kernel with a novel and general low-rank decomposition, enabling efficient representation through deep neural networks and computational efficiency and better performance.

Some further food for thought

$$\lambda(t, s \mid \mathcal{H}_t) = \lim_{\Delta t \downarrow 0, \Delta s \downarrow 0} \frac{\mathbb{E}\left[\mathbb{N}([t, t+\Delta t] \times B(s, \Delta s)) \mid \mathcal{H}_t\right]}{|B(s, \Delta s)| \Delta t},$$

$$\lambda(t, s) = \mu + \sum_{(t', s') \in \mathcal{H}_t} k(t', t, s', s),$$

$$k(t', t - t', s', s - s') = \sum_{r=1}^{R} \sum_{l=1}^{L} \alpha_{lr} \psi_l(t') \varphi_l(t - t') u_r(s') v_r(s - s').$$

Here $\{\psi_l, \varphi_l : [0, T] \to \mathbb{R}, l = 1, \ldots, L\}$ are two sets of temporal basis functions that characterize the temporal influence of event at t' and the decaying effect brought by elapsed time $t - t'$. Similarly, spatial basis functions $\{u_r, v_r : \mathcal{S} \to \mathbb{R}, r = 1, \ldots, R\}$ capture the spatial influence of event at s' and the decayed influence after spreading over the displacement of $s - s'$. The corresponding weight α_{lr} at different spatio-temporal ranks combines each set of basis functions into a weighted summation, leading to the final expression of influence kernel k.

Some further food for thought

- Use a fully-connected neural network to represent each basis function. The history or displacement is taken as the input and fed through multiple hidden layers equipped with Softplus non-linear activation function.
- For an influence kernel with temporal rank L and spatial rank R, we need 2(L + R) independent neural networks for modeling.
- Benefits:
- (i) The kernel parameterization with displacement significantly reduces the rank needed when representing complicated kernels
- (ii) The non-stationarity of original influence of historical events over spatio-temporal space can be conveniently captured by inhomogeneous ψ_l and u_r, making the model applicable in modeling general STPPs.
- (iii) The propagating patterns of influence are characterized by ϕ_l and v_r which go beyond simple parametric forms

References

- Alba-Fernández, M. V., Ariza-López, F. J., Jiménez-Gamero, M. D., & Rodríguez-Avi, J. (2016). On the similarity analysis of spatial patterns. *Spatial Statistics*, 18, 352-362.
- Cholaquidis, A., Forzani, L., Llop, P., & Moreno, L. (2017). On the classification problem for Poisson point processes. *Journal of Multivariate Analysis*, 153, 1-15.
- Cronie, O., Moradi, M., & Biscio, C. A. (2021). Statistical learning and cross-validation for point processes. arXiv preprint arXiv:2103.01356.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- Chicco, D. (2021). Siamese neural networks: An overview. In *Artificial Neural Networks*, 73-94.
- Koch, G., Zemel, R. & Salakhutdinov, R. (2015). Siamese neural networks for one- shot image recognition. In: *ICML deep learning workshop*, Lille, vol 2

Thank you for your attention!