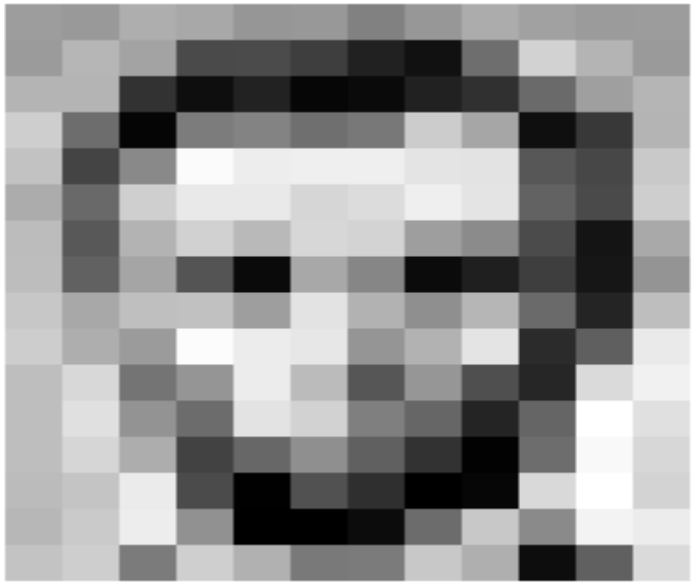
Deep Learning for Remote Sensing

AI60002

28th Feb 2022

Deep Learning for Images

- Provided: images (2D or 3D matrices)
- X(i,j,k): a pixel on the i-th row, j-th column and k-th channel
- Information: neighboring pixels likely to have similar values (spatial autocorrelation)
- Further: neighboring pixels likely to belong to same object



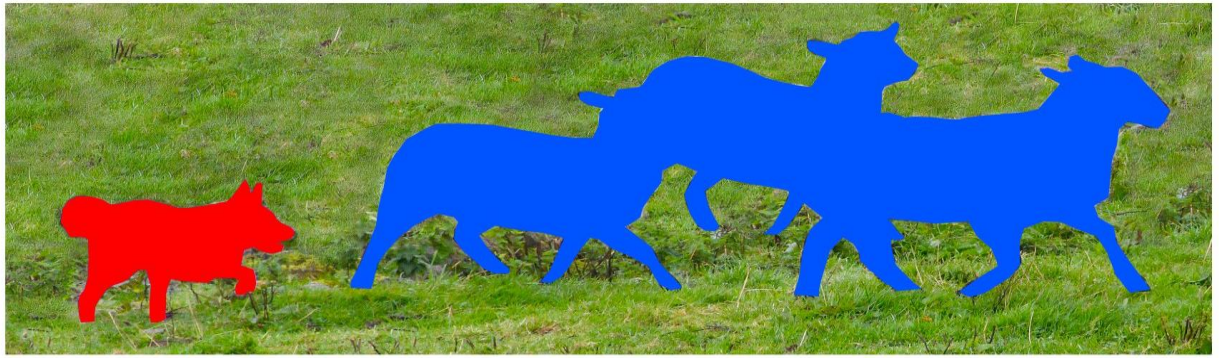
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

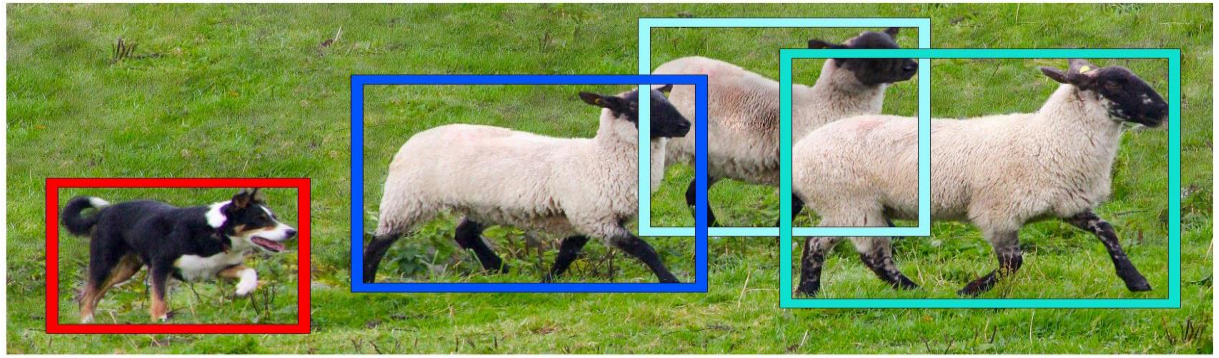
Deep Learning for Images

- Object Recognition
- Object Detection (for a particular object/set of objects)
- Image segmentation

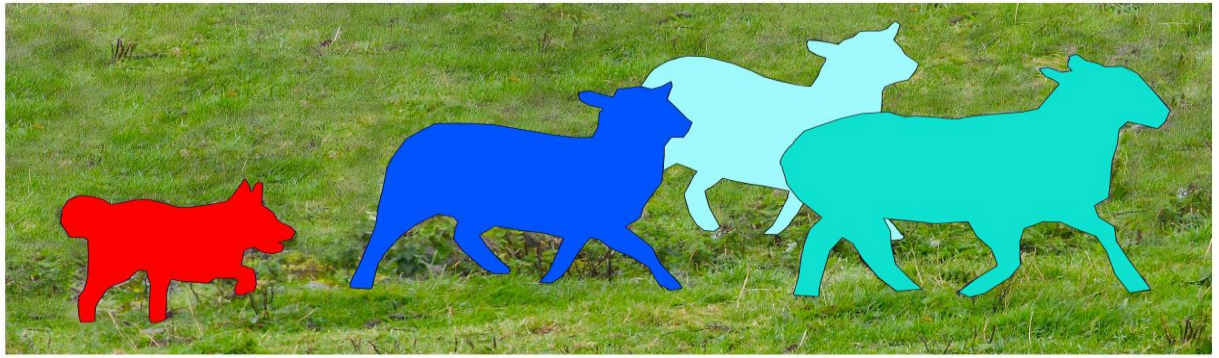
Image Recognition



Semantic Segmentation



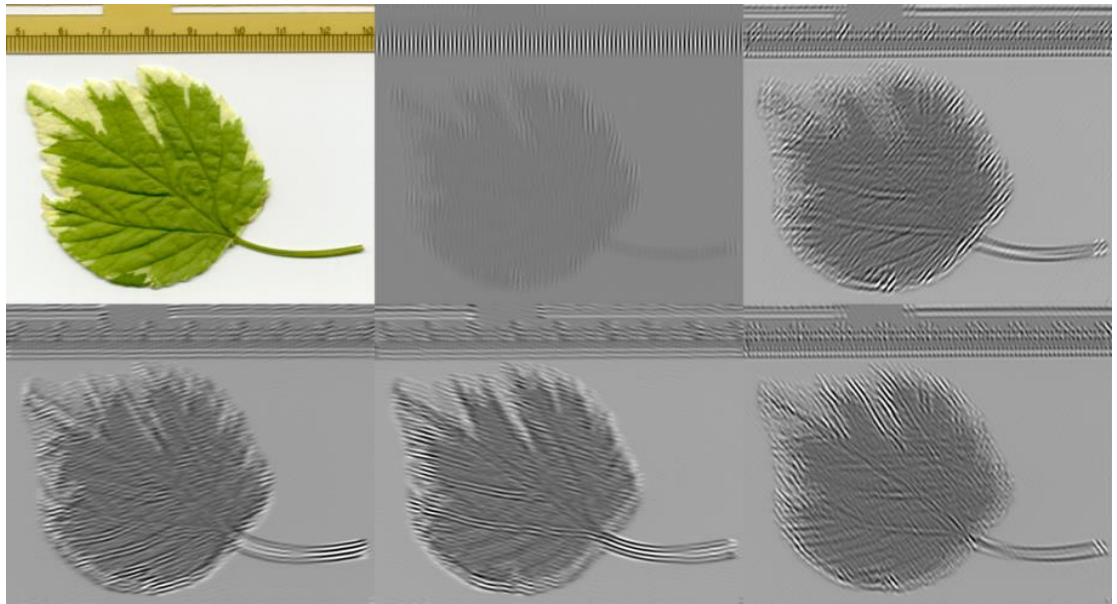
Object Detection

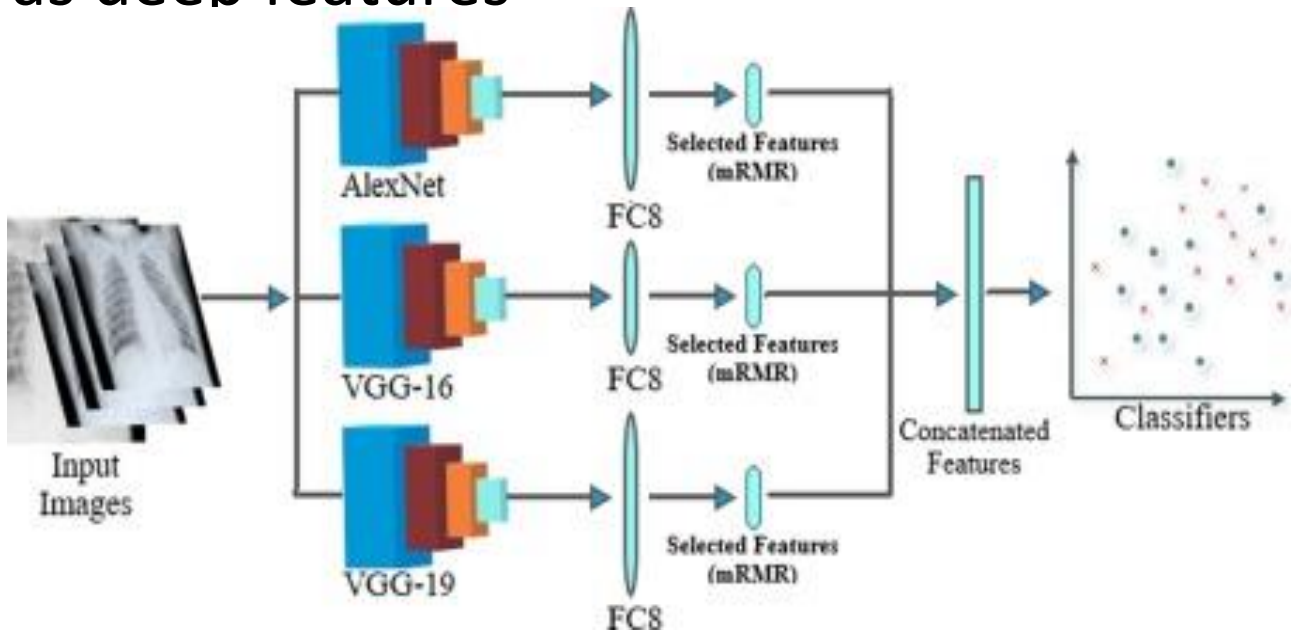


Instance Segmentation

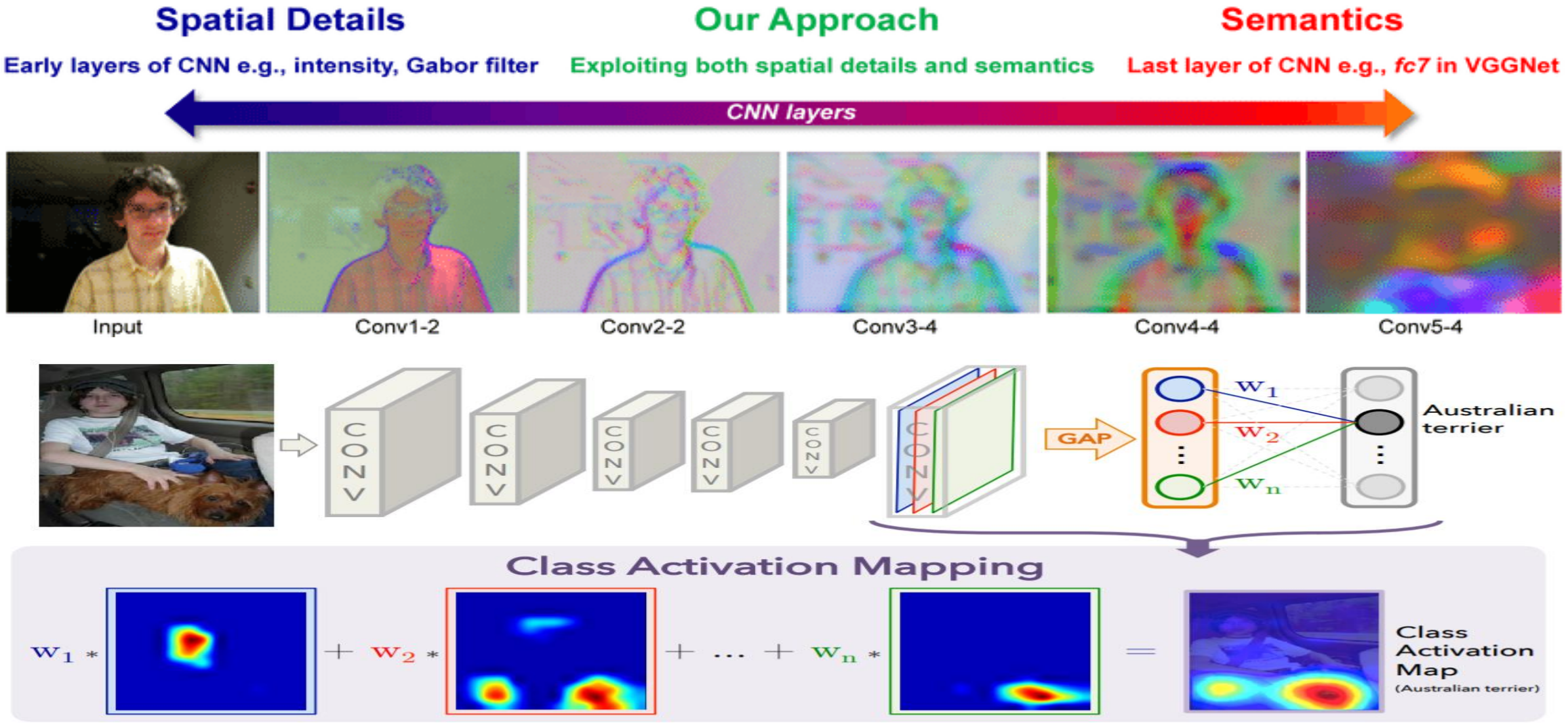
Deep Learning for Images

- Image representation – each image must be represented, as a whole and in parts
- Representation may be using raw pixel values, filter outputs
- Deep features: an image/sub-image provided as input to a neural network
- Each layer of the neural network creates a new representation of the image
- These representations can be used as deep features





Deep Learning for Images



Max-Pooling and Convolution Operations

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

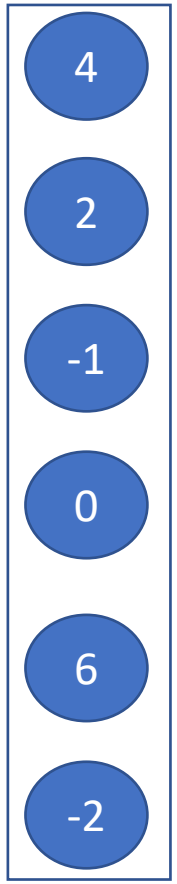
Block size:2
Stride: 2

6	8
3	4

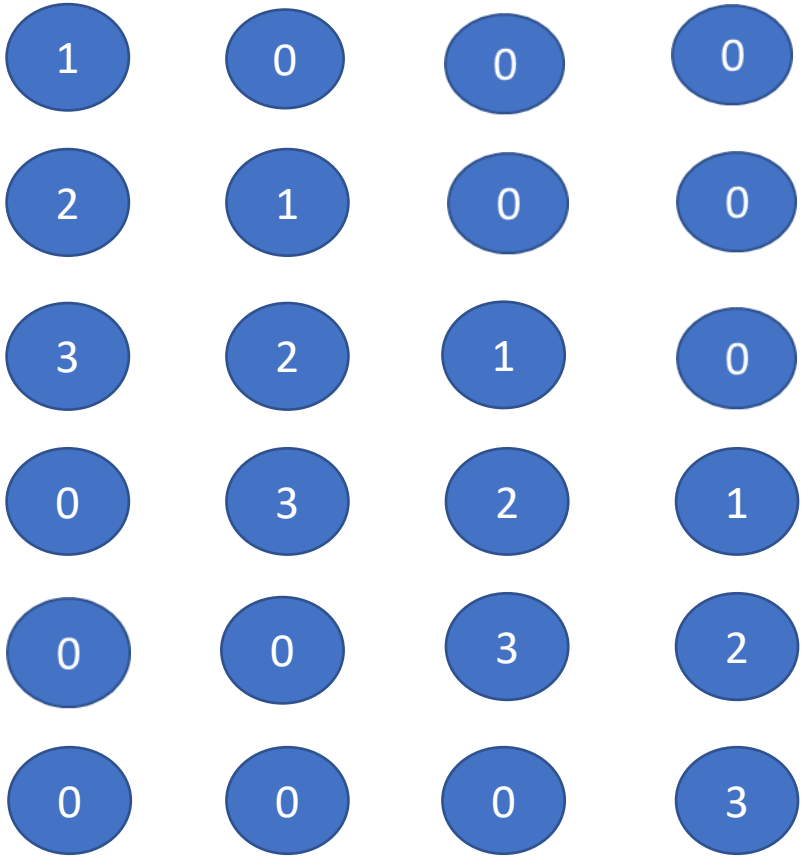
1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

Block size:3
Stride: 1

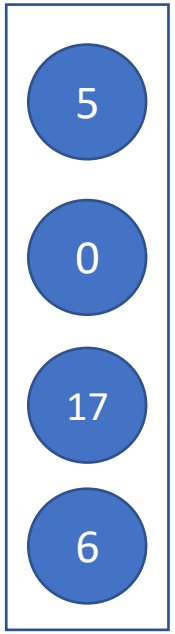
7	8
7	8



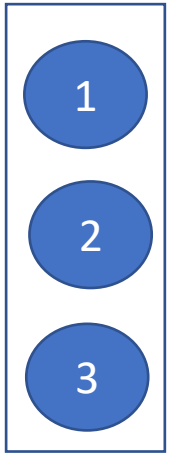
INPUT



WEIGHT MATRIX



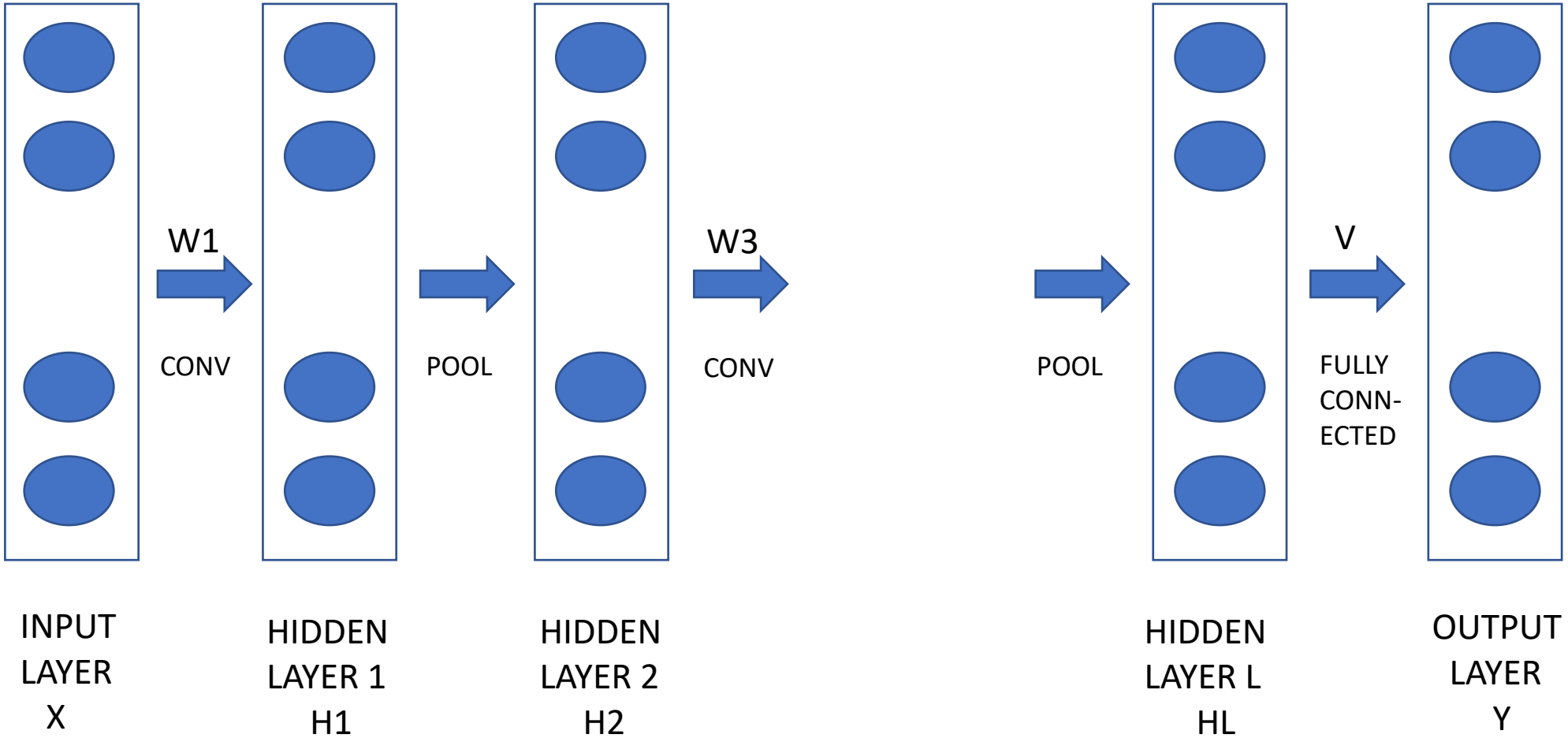
OUTPUT



REPEATING STRUCTURE

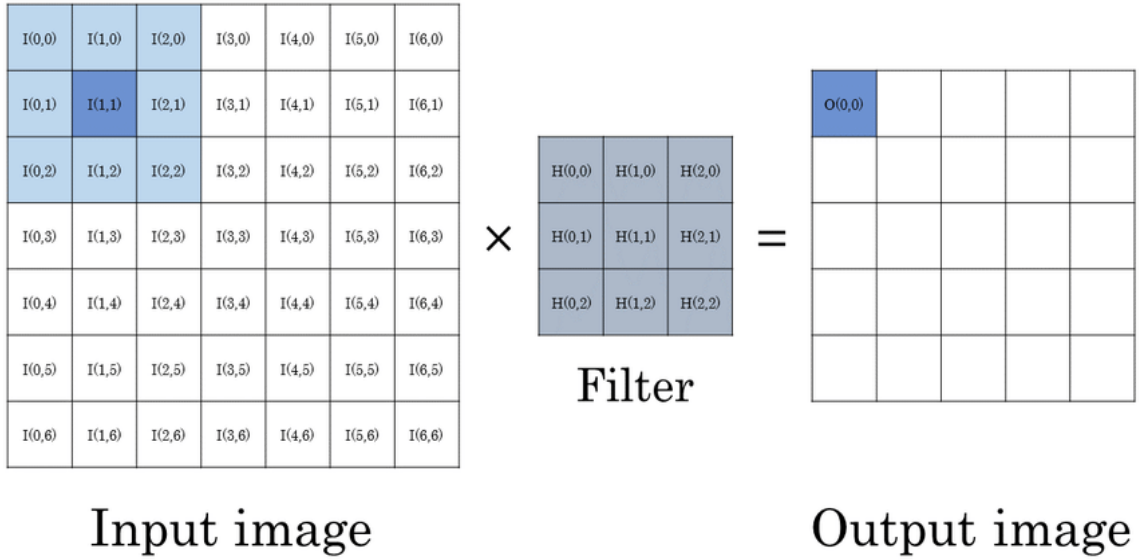
Convolutional Neural Network

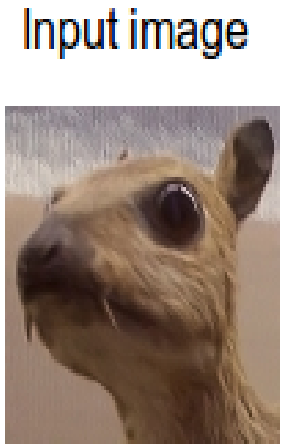
- A convolutional neural network has many “convolution layers”
- Usually, each convolutional layer is followed by a pooling layer



Deep Learning for Images

- Convolution – a standard operation on images
- Useful to identify local structures/orientations in images
- One pass of a convolution filter over the whole image provides the location of a certain kind of orientations



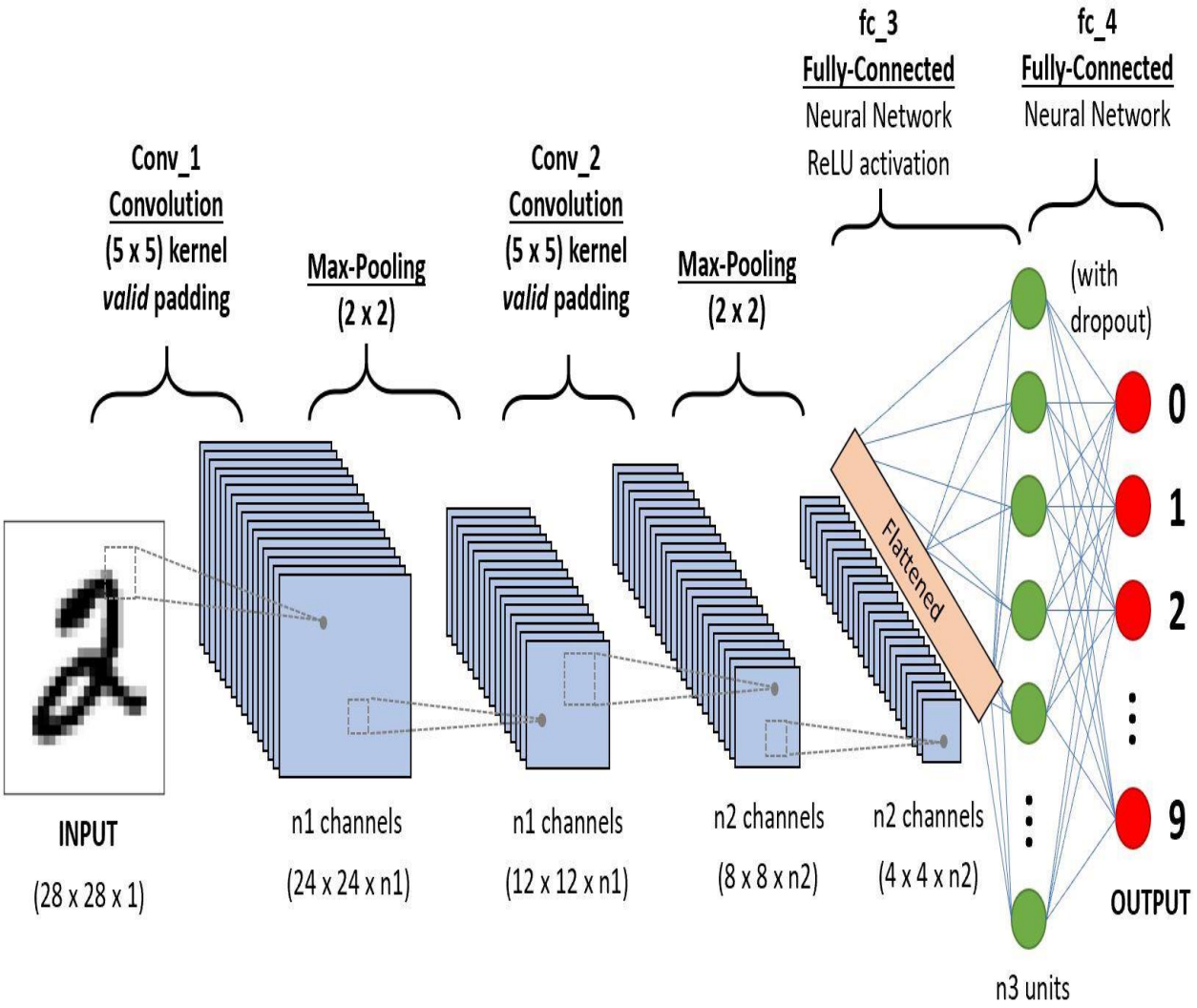


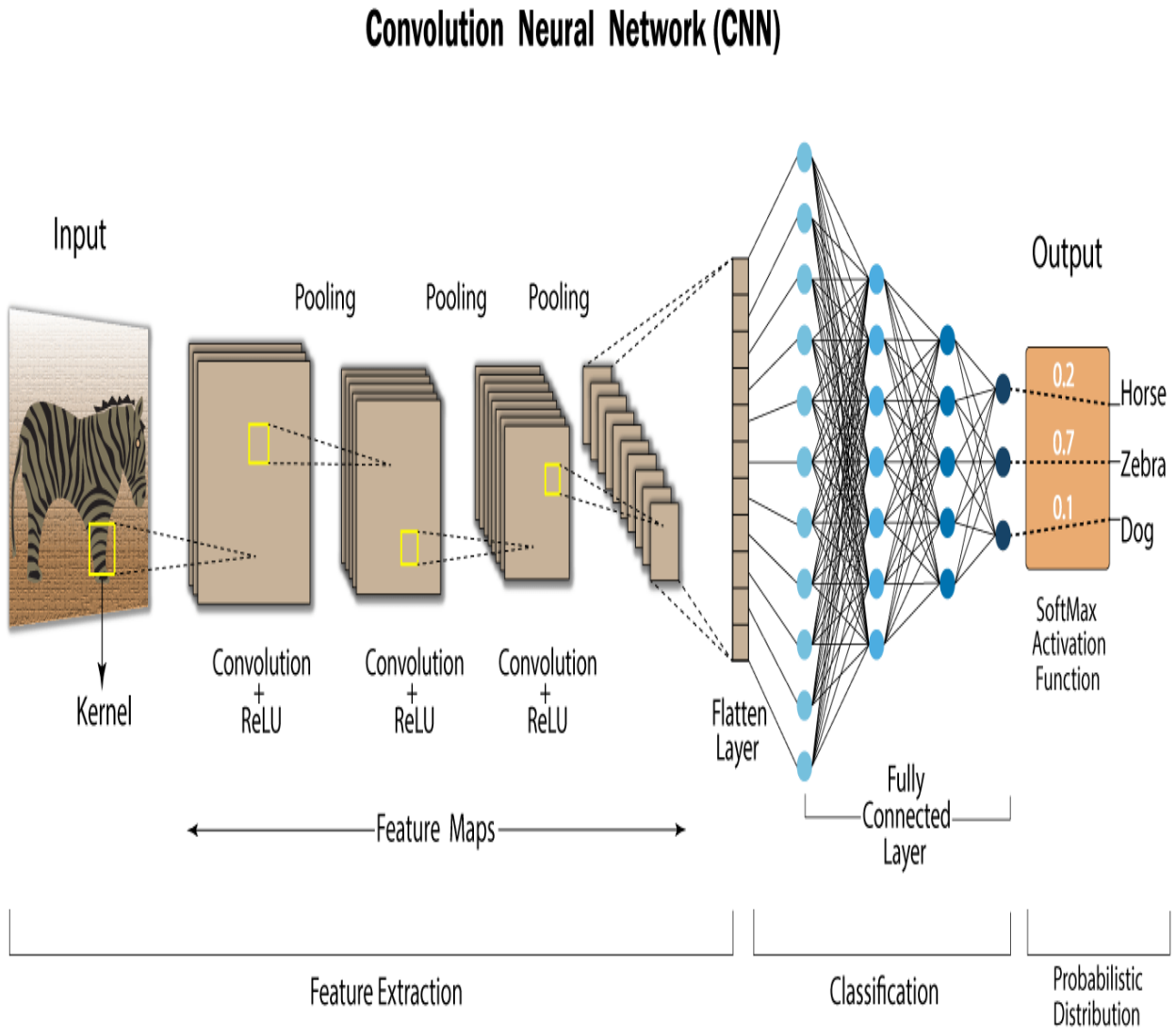
Convolution Kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

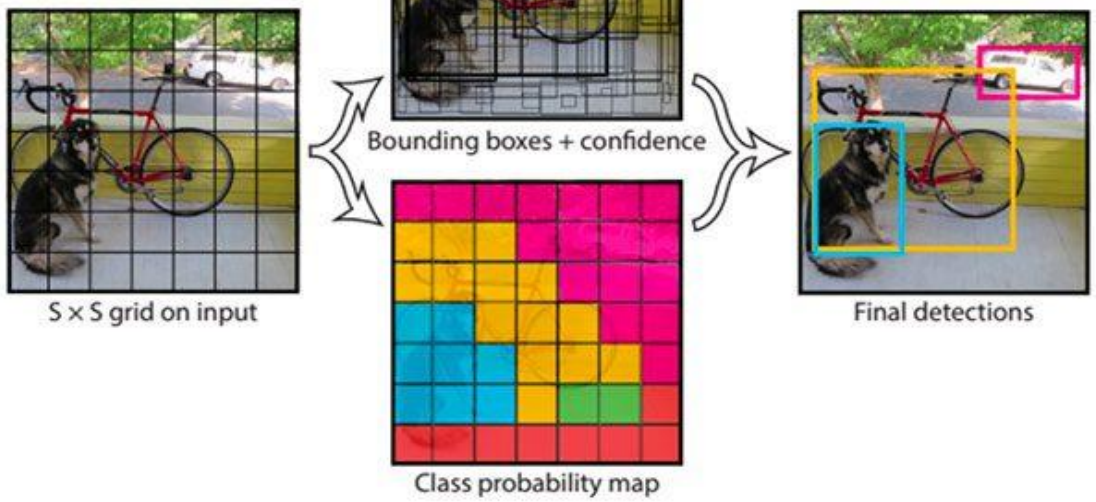
- Can be done with a specially structured neural network (repeating edge weights)

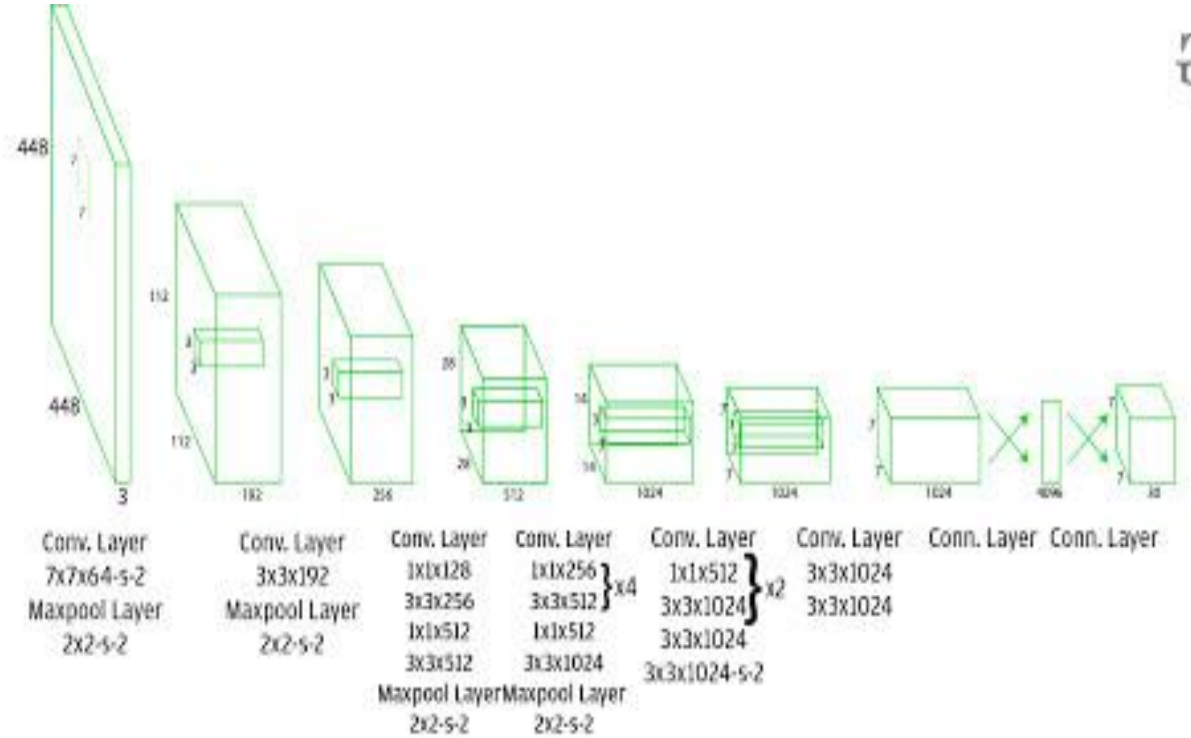
Deep Learning for Images





Object Detection with YOLO – You Only Look Once





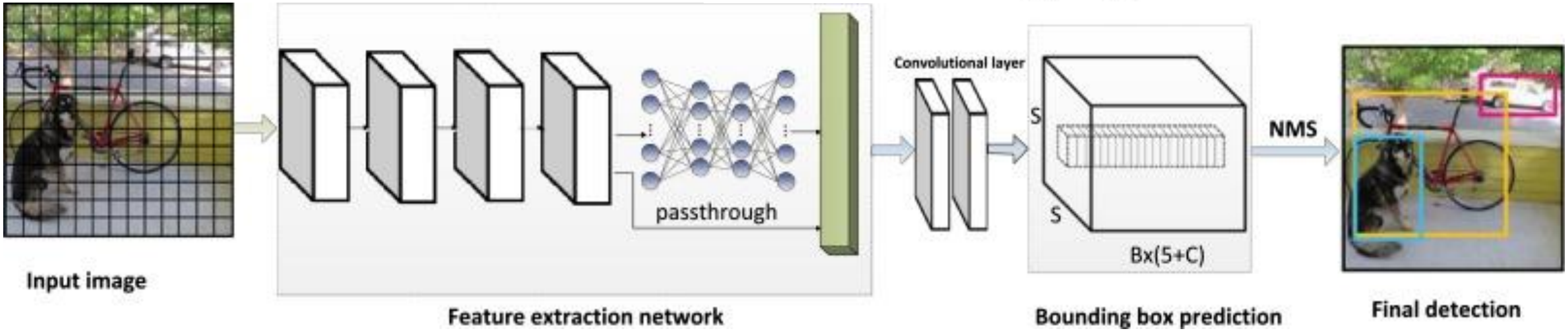
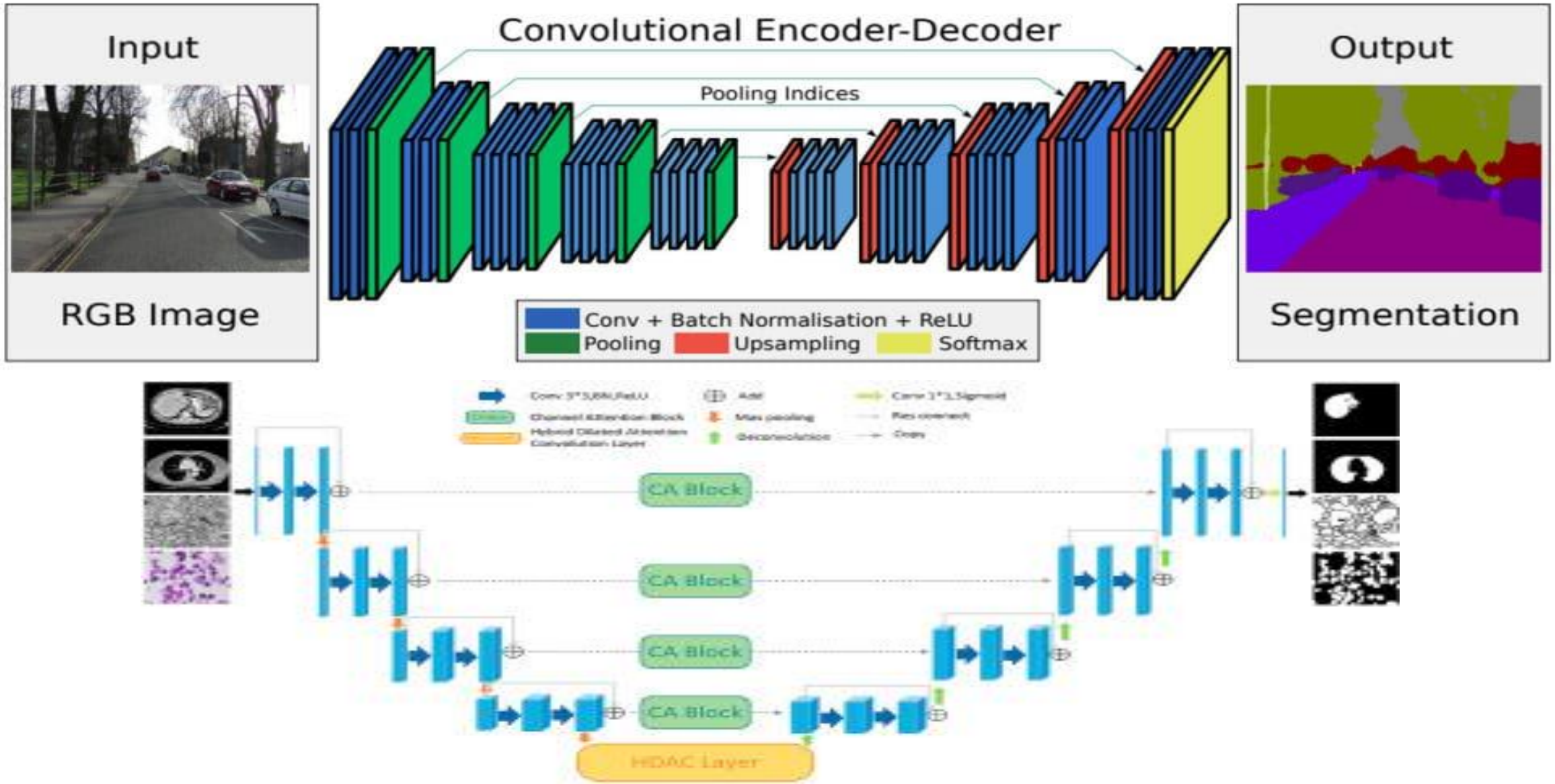


Image Semantic Segmentation



Object Detection in Satellite Images

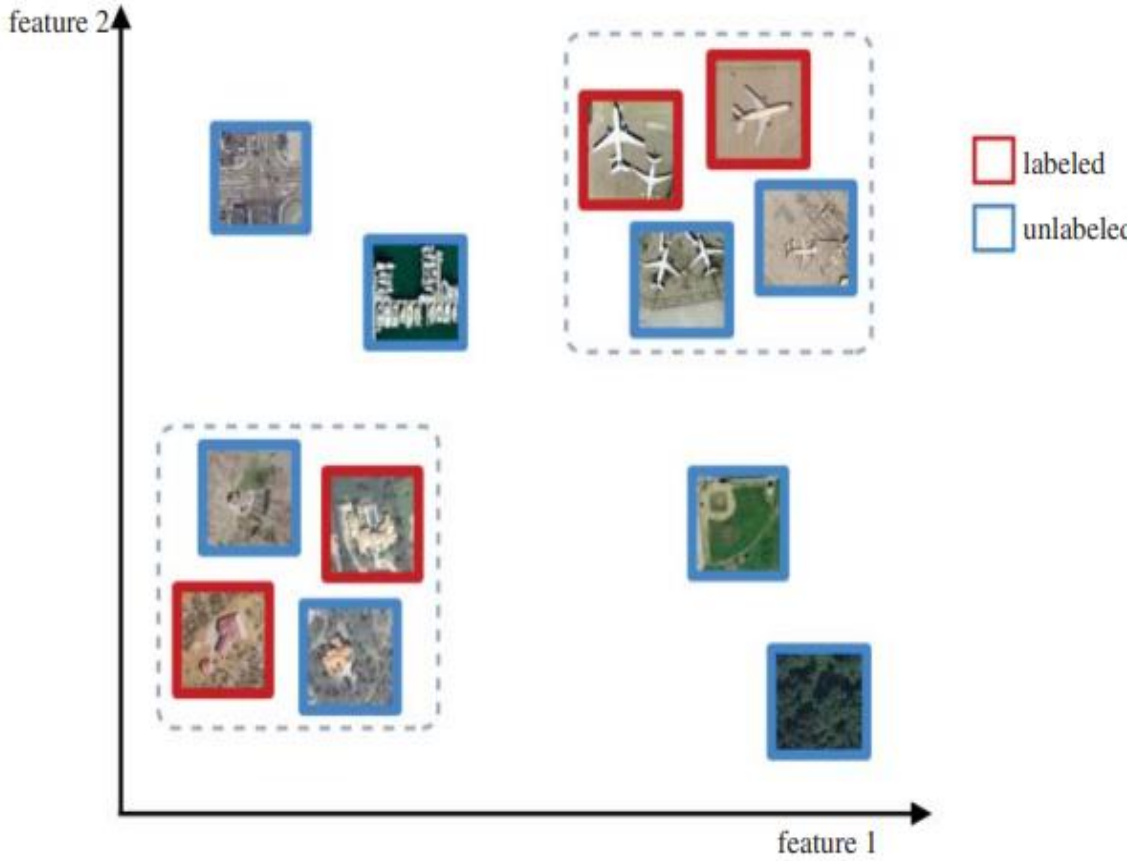


Figure 4.1 Schematic illustration of different learning paradigms and their use of labeled (red) and unlabeled (blue) data samples. In contrast to semi-supervised learning (data samples used shown in dotted boxes), self-taught learning also uses unlabeled data, which need not belong to the same classes as the labeled data. Images are from the UC Merced dataset (Yand and Newsam 2010).

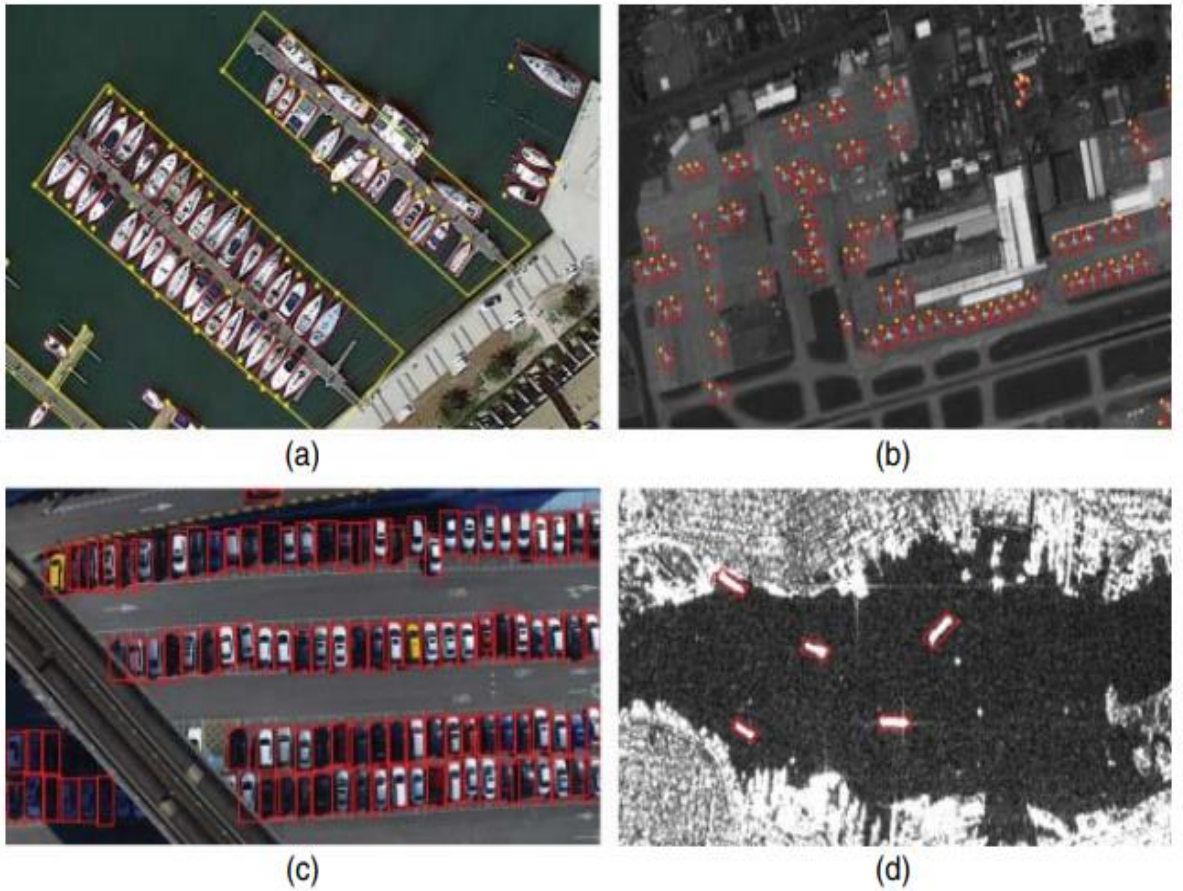


Figure 6.1 Examples of remote sensing images containing objects of interest. (a) An image from Google Earth, containing ships and harbors. (b) An image form JL-1 satellite, including planes. (c) An drone-based image containing many vehicles. (d) A SAR image, containing ships.

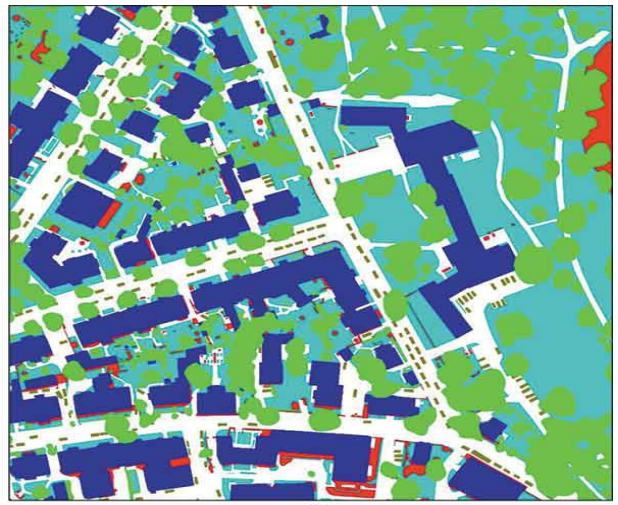
Segmentation of Satellite Images

(a) Original image

(b) Ground truth

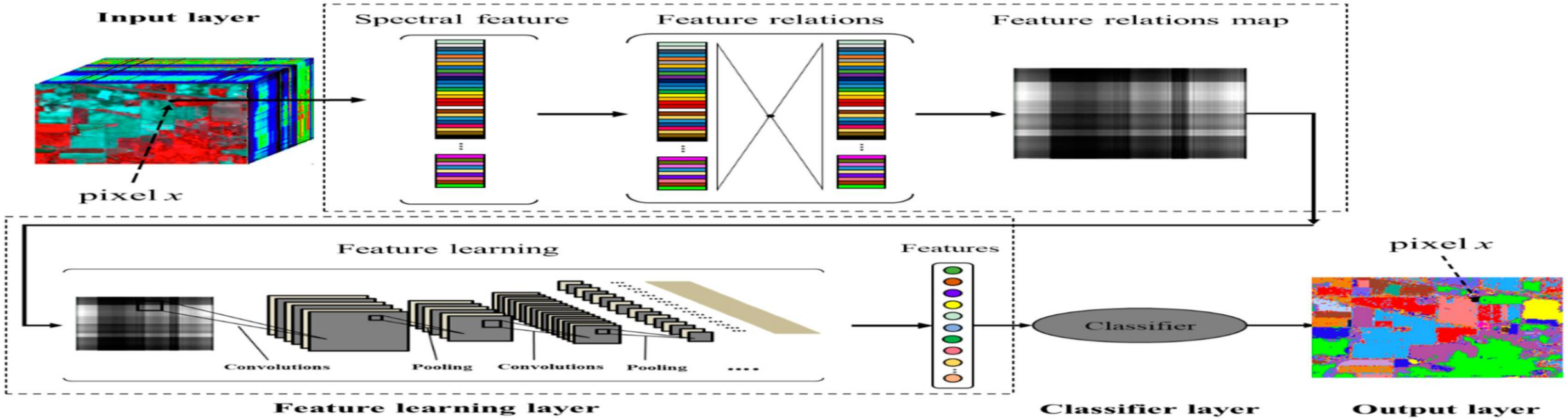
Land-Use Land Cover mapping

(a)

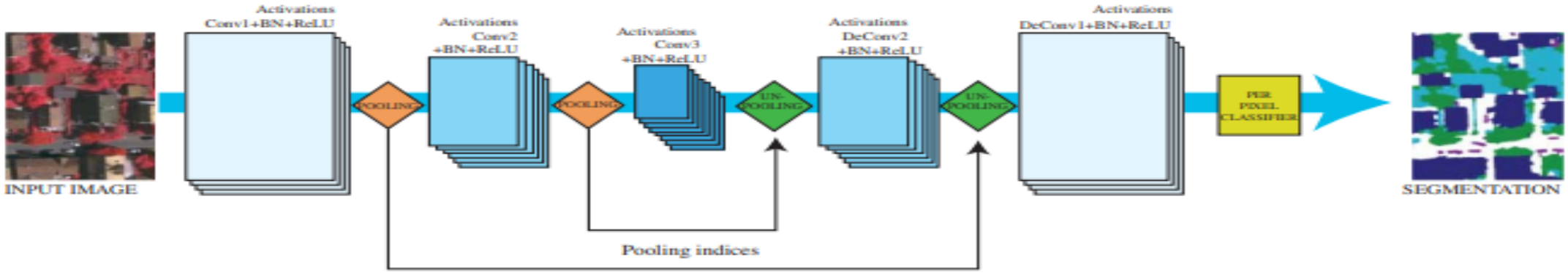


(b)

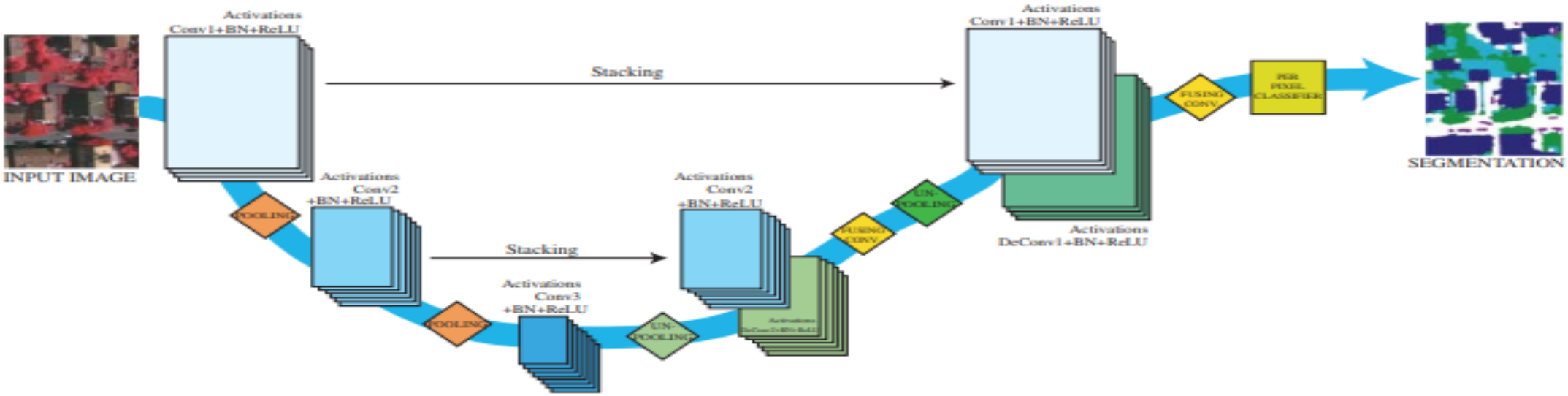
Road Network Extraction



Segmentation of Satellite Images



(a) SegNet (Badrinarayanan et al. 2017), propagating pooling indices.



(b) U-Net (Ronneberger et al. 2015a), propagating activation maps.

Figure 5.3 Semantic segmentation architectures learning the upsampling.

Deep Domain Adaptation

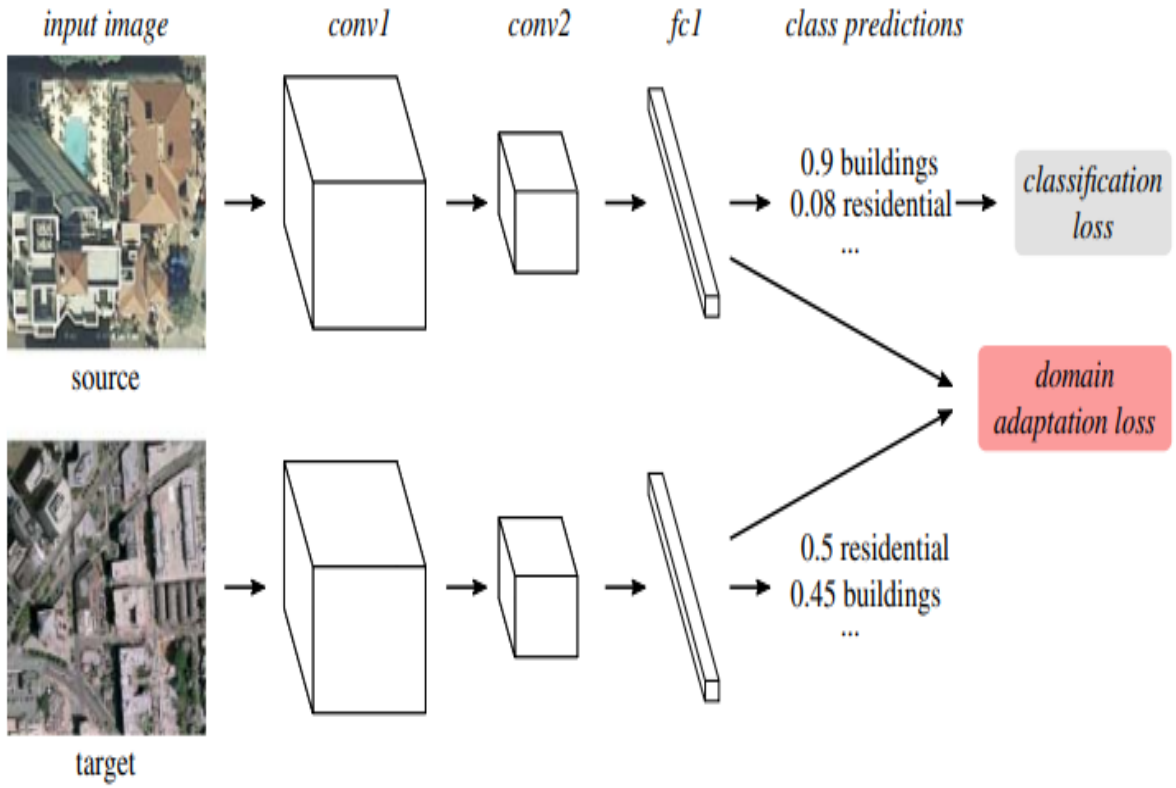


Figure 7.1 Domain adaptation loss (red) imposed on a CNN's feature vectors produced by the penultimate layer ("fc1").

Figure 7.2 Examples from the UC Merced (top) and WHU-RS19 (bottom) datasets.

Geophysical Networks

Adway Mitra

ML for Earth System Sciences (AI60002)

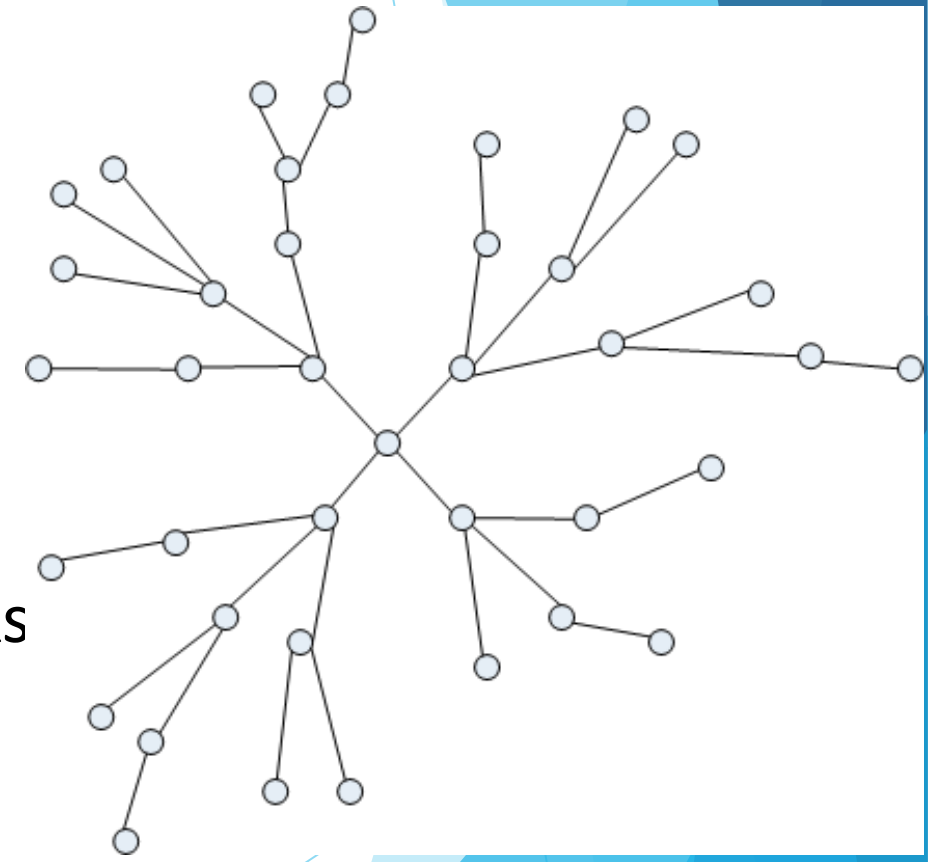
8th March 2021

Networks and Graphs

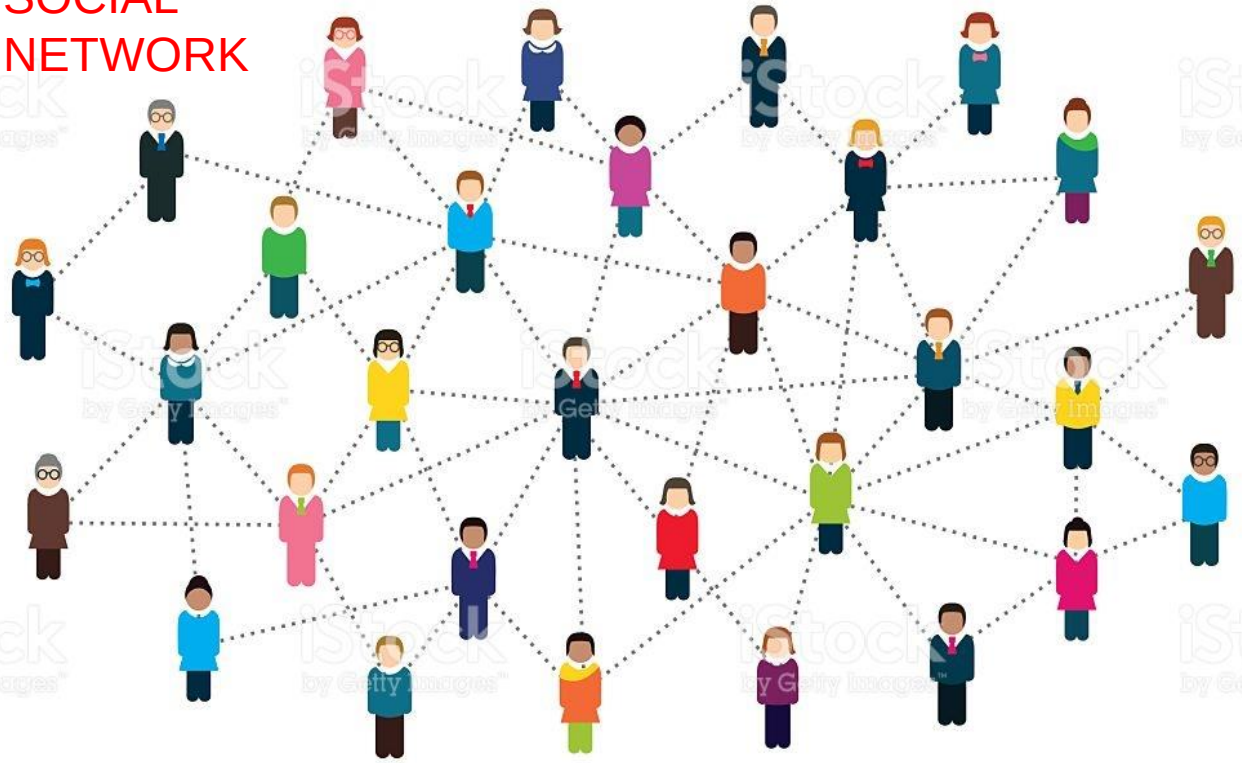
- V: set of nodes
- E: set of edges; an edge connects two nodes

- Each node represents an entity of some kind
- Edges represent interactions between them

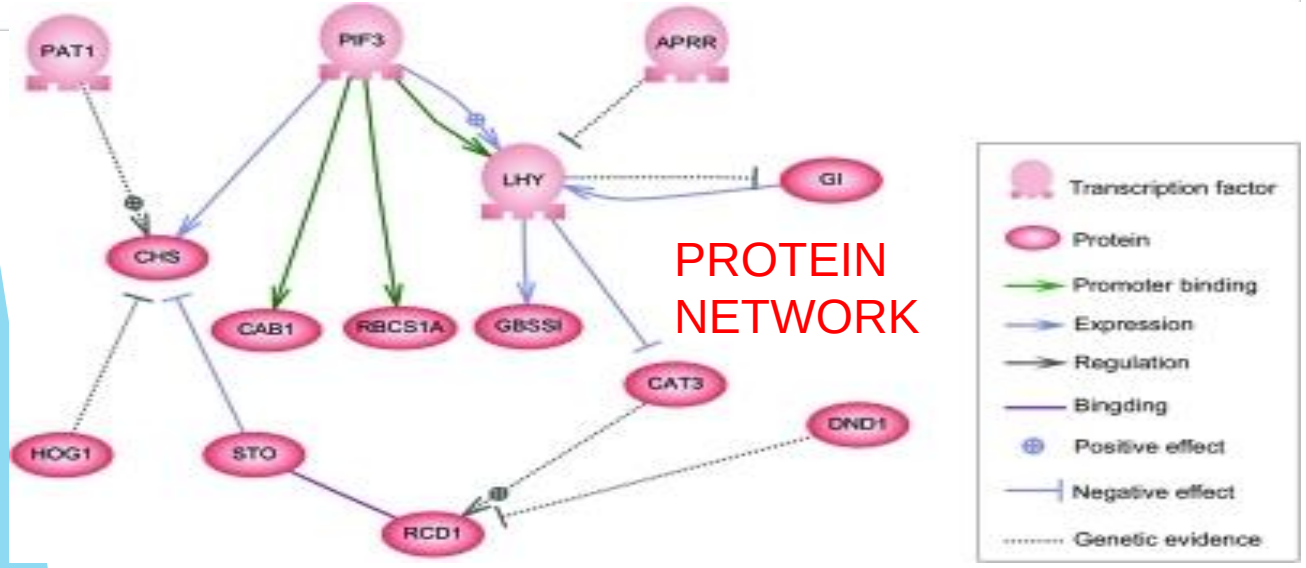
- Examples: computer networks, social networks biological networks, road/transport networks



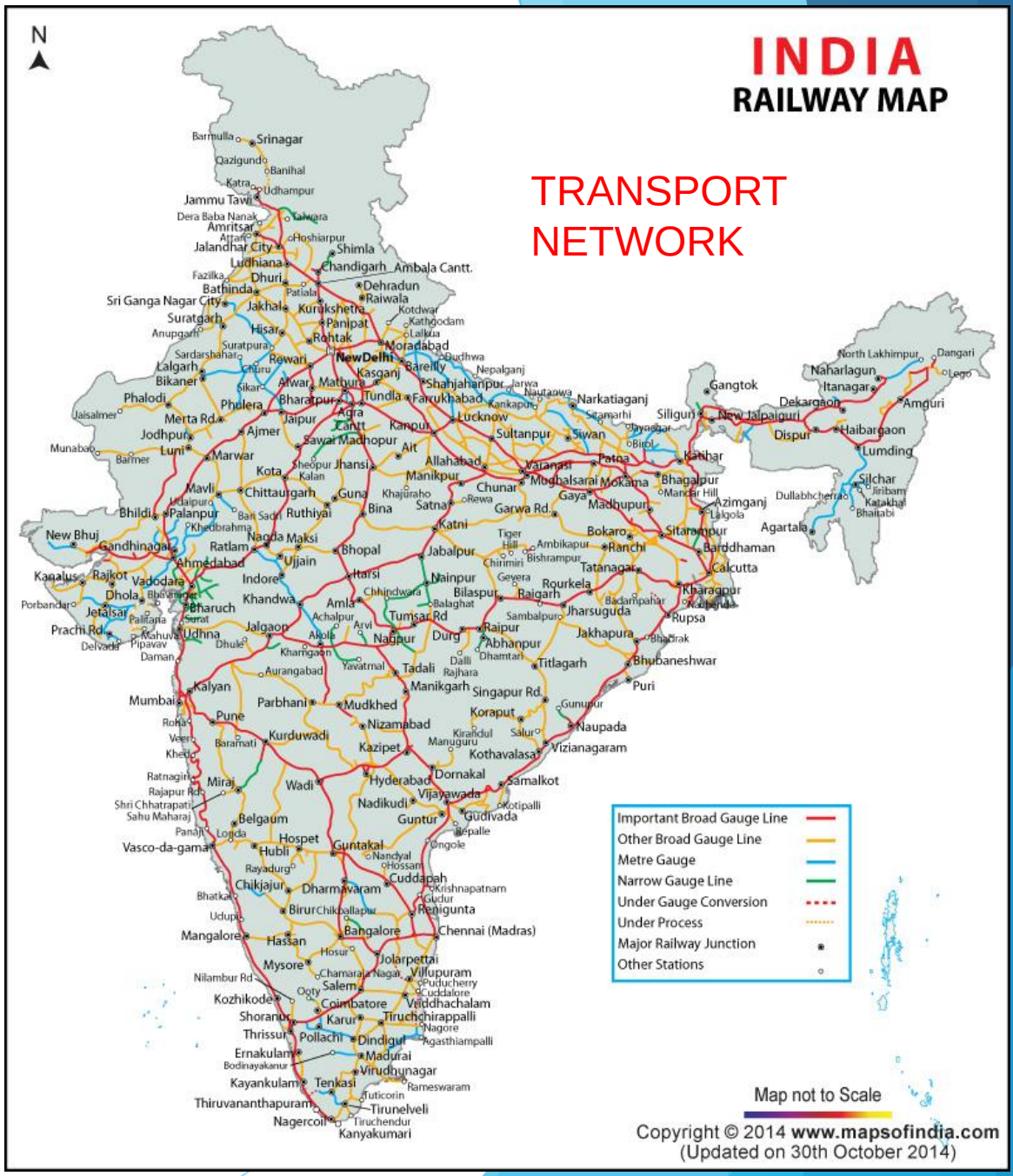
SOCIAL NETWORK



PROTEIN NETWORK



TRANSPORT NETWORK



Why use networks?

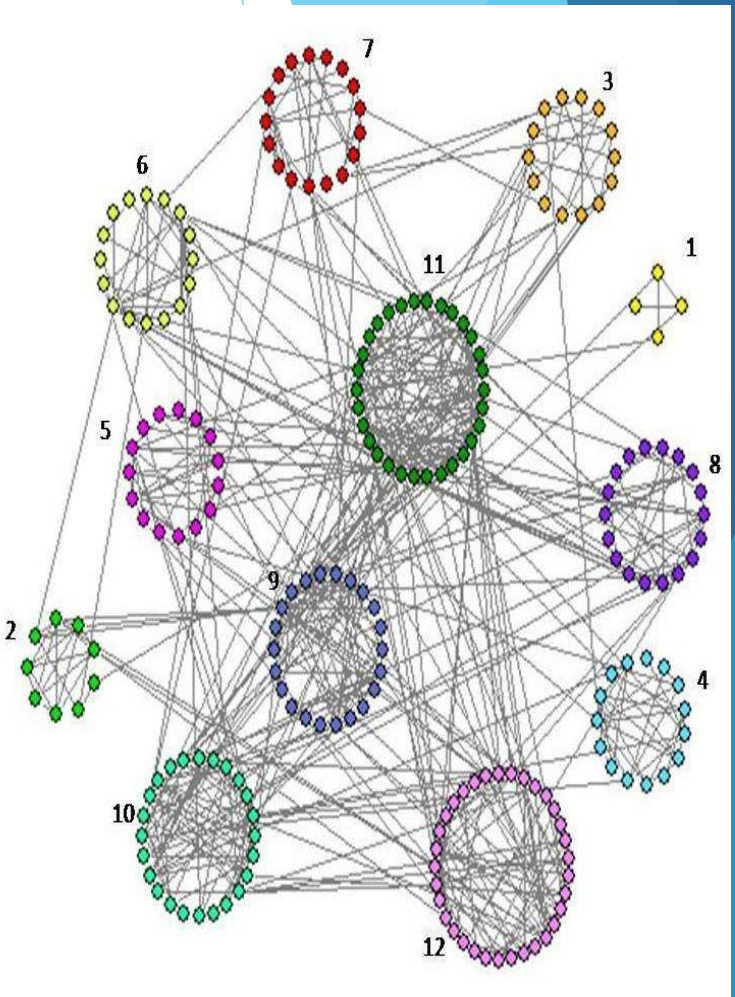
- Study the nature of interactions among entities and their dynamics
- Identify “communities” where members interact strongly,

  e.g. common interest groups in social networks

Model dynamic processes,

e.g. flow of information in social network

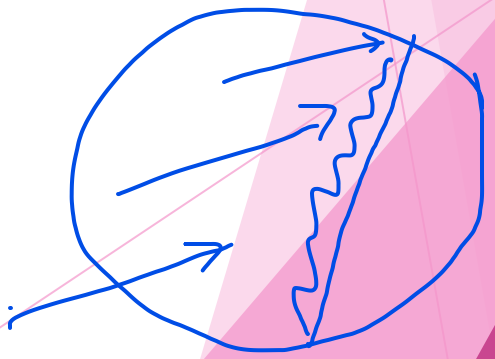
flow of infectious disease



Geophysical Networks

- Geophysical **networks to visualize and analyse spatio-temporal geophysical data**
- To identify regions whose geophysical conditions are strongly related
- To identify teleconnections
- To identify causal relationships between geophysical events
- To identify relationships among different geophysical variables

- But how to construct a geophysical network?



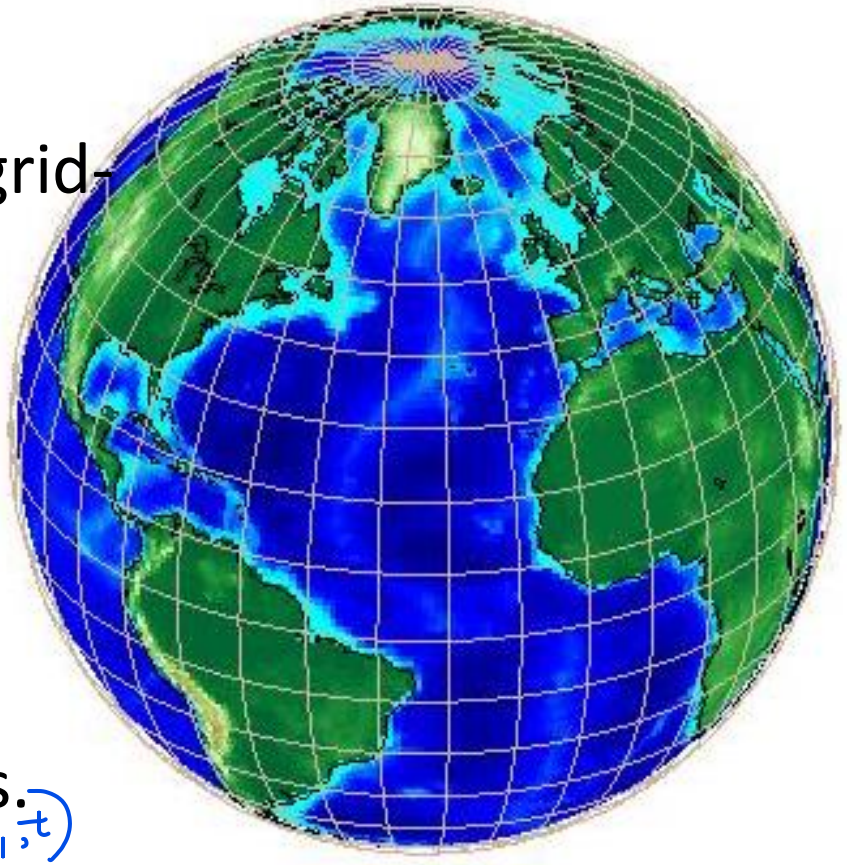
Networks and Graphs

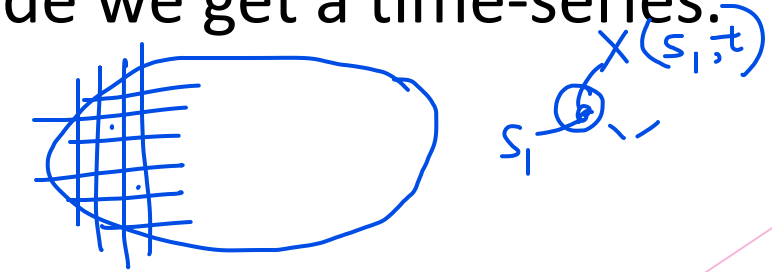
Design Issues –

- What does each node represent?
- Which nodes are connected by edges?
- Are edges directed?
- What does each edge represent?
- Is the graph fixed or does it evolve over time?

Geophysical Networks: Defining Nodes

- In most geophysical networks, each node represents a spatial location e.g. one node per grid-point.
- Each node has an attached variable, e.g. rainfall received by the corresponding location in any time-step (hour/day/month/year).
- At each time-step, we get an “instantiation” of the network; for each node we get a time-series.





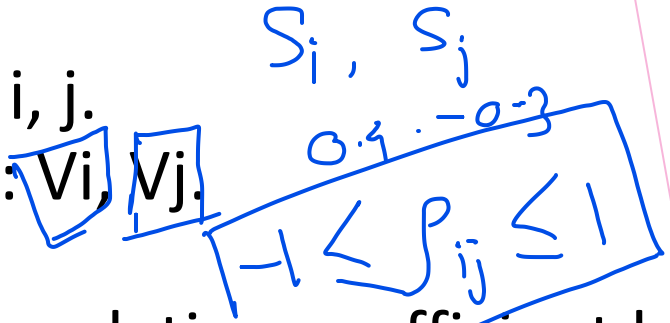
Geophysical Networks: Defining Edges

- Once the set of nodes has been chosen, we need to select how we define connections between pairs of nodes (edges).
- Edges can be weighted or unweighted; directed or undirected.
- Edge weight: a measure of “similarity” between the pair of nodes.

- Consider each pair of nodes in a network
- Measure their “similarity”
- Put an edge if similarity is above a threshold.

But what is a measure of “similarity”?

Geophysical Network Edges: Correlation



- Consider two nodes (two grid points): i, j.
- Their corresponding climatic variables: Vi, Vj.
- → two time series: {Vi(t)}, {Vj(t)}.
- Define edge weight W(i,j) = Pearson correlation coefficient between the two time-series.

- **Seminal paper by** Tsonis and Roebber **"climate networks" (2004)**
- Identify all **pairs** of grid points with correlation > 0.5.

→ "Correlation Network"

→ First type of climate network ever defined.

Tsonis, A. A., & Roebber, P. J. (2004). "The architecture of the climate network." *Physica A: Statistical Mechanics and its Applications*, *333*, 497-504.

Geophysical Network Edges: Mutual Information

- Another criterion: Mutual Information between the variables
- Measures the *statistical dependence* between them

$$M_{ij} = \sum_{\mu\nu} p_{ij}(\mu,\nu) \log \frac{p_{ij}(\mu,\nu)}{p_i(\mu)p_j(\nu)},$$

$$\frac{p(X_i=\mu, X_j=\nu)}{p(X_i=\mu)\,p(X_j=\nu)} \qquad \left[\begin{matrix} X_j(t)=\nu \\ X_j(t)=\nu \end{matrix}\right]$$

- The excess amount of information generated by falsely assuming the two time-series at nodes i and j to be independent.
- Able to detect nonlinear relationships.

$$\left[\begin{matrix} X_j(t)=\mu \\ X_j(t+\delta)=\nu \end{matrix}\right]$$

Donges, J. F.; Zou, Y.; Marwan, N.; Kurths, J. (2009). "Complex Networks in Climate Dynamics". *The European Physical Journal Special Topics*. Springer-Verlag. **174** (1): 157–179

Synchronization and Lag

- Often, time-series at two locations are not perfectly synchronized.
- Influence of one region may take time to reach another region.
- Especially true for spatially distant locations.
- Lag networks: compare {Vi(t)}, {Vj(t+Δ)}, where Δ is a suitable “lag”.

G. Tirabassi and C. Masoller, 2013. On the effects of lag-times in networks constructed from similarities of monthly fluctuations of climate fields. Europhysics Letters Vol 102 (2013).

- How to identify the “best” lag for any pair of nodes to maximize their correlation? Sequence Alignment/Dynamic Time Warping?

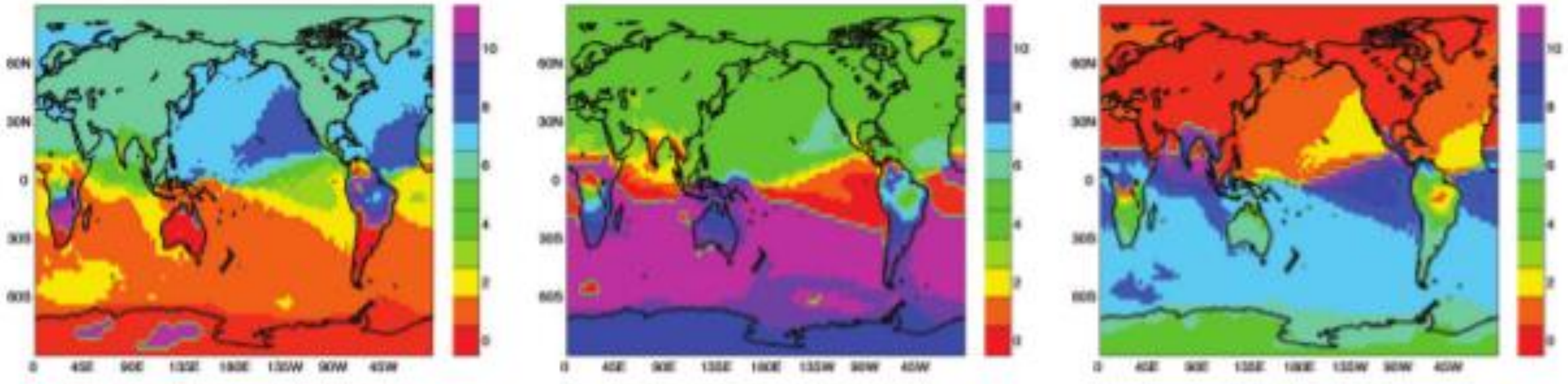


Fig. 2: (Color online) Lag-times of a node in Mongolia (left), in Australia (center) and in the El Niño basin (right).

PC: Tirabassi et al, EPL 102 (2013)

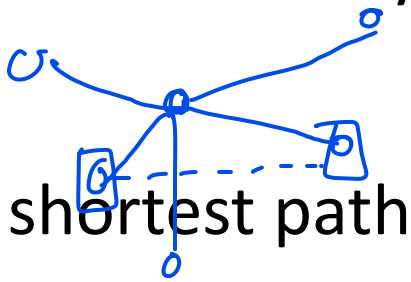
Geophysical Network Edges: Event Synchronization

- Define “events” for each time-series.
- E.g. annual rainfall at a location exceeding a threshold.
- Event a in time-series Vi, event b in time-series Vj are synchronized if |a-b|< threshold.
- How often are events of two time-series synchronized?
- Very relevant for extreme event analysis!

Geophysical Networks: Properties

Some networks are so large/dense that one can only print properties of the networks, not the connections themselves.

- **Degree distribution** - number of edges per node.
- **Local/global clustering coefficient** – probability that two randomly chosen neighbors of any node are themselves neighbors.
- **Centrality** – mean inverse shortest path to each node from a given node
- Number and size distribution of **connected components.**
- **Area-weighted Connectivity** – earth area covered by neighbors of each node.
- **Diameter, path length** distribution.
- **Small-world property** – distribution of shortest path length between pairs of nodes.



Geophysical Networks: Degree Distribution

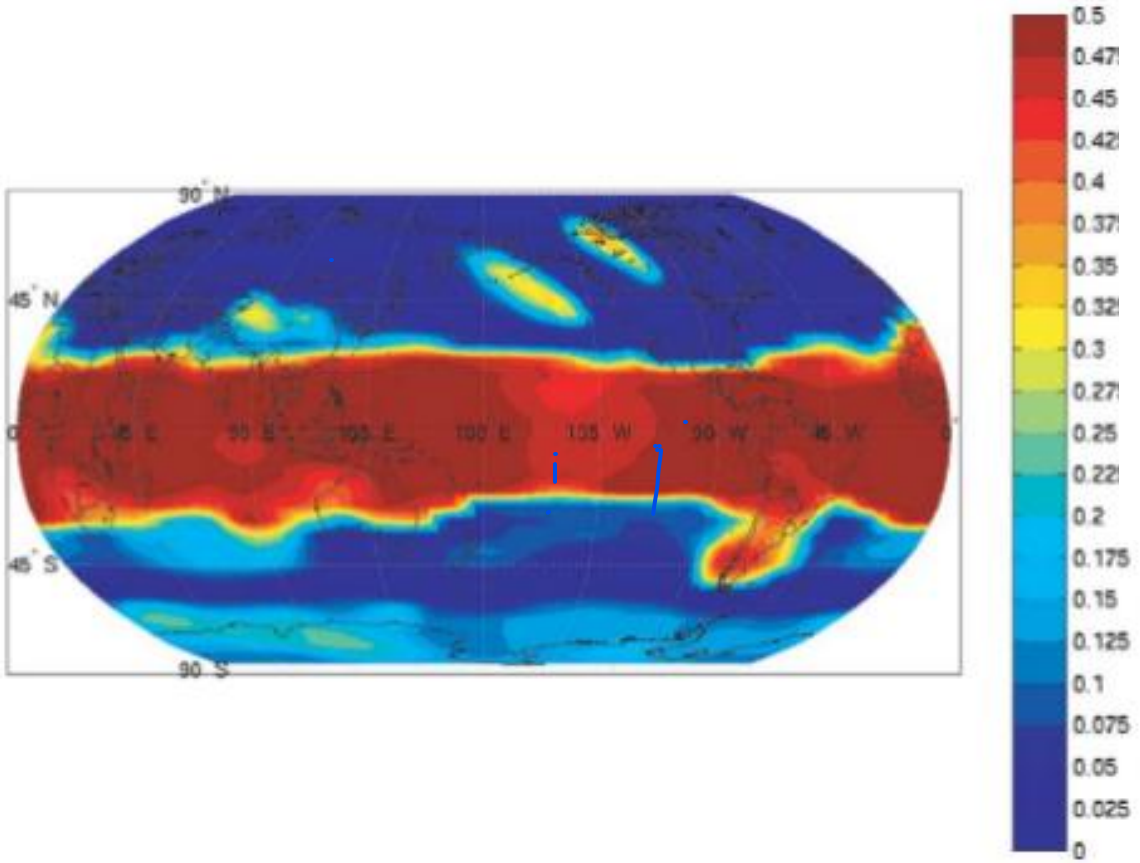
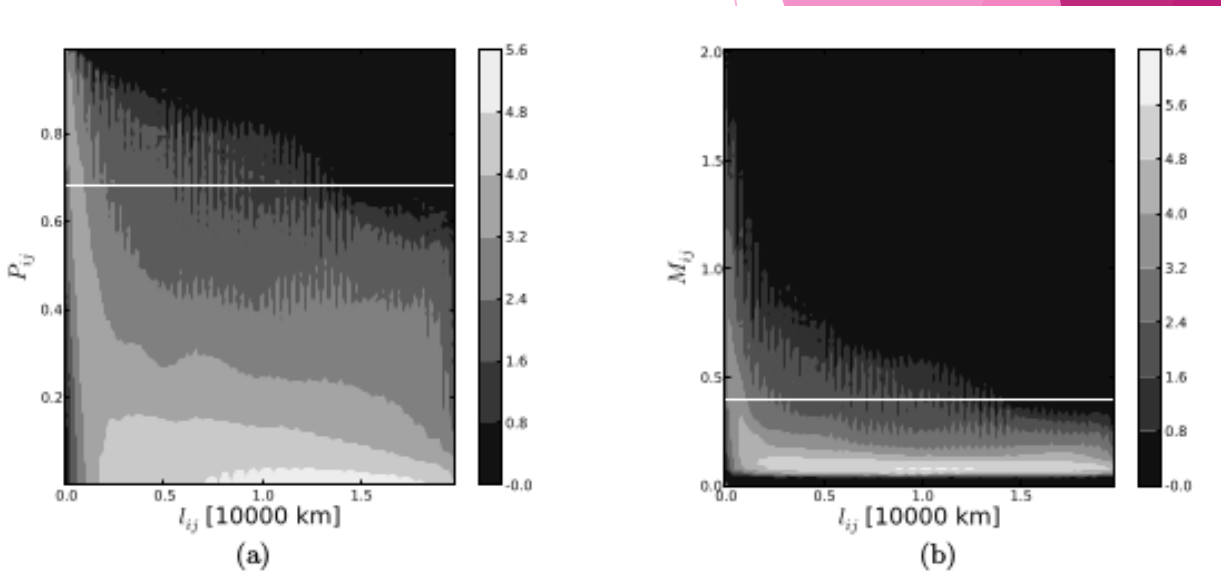


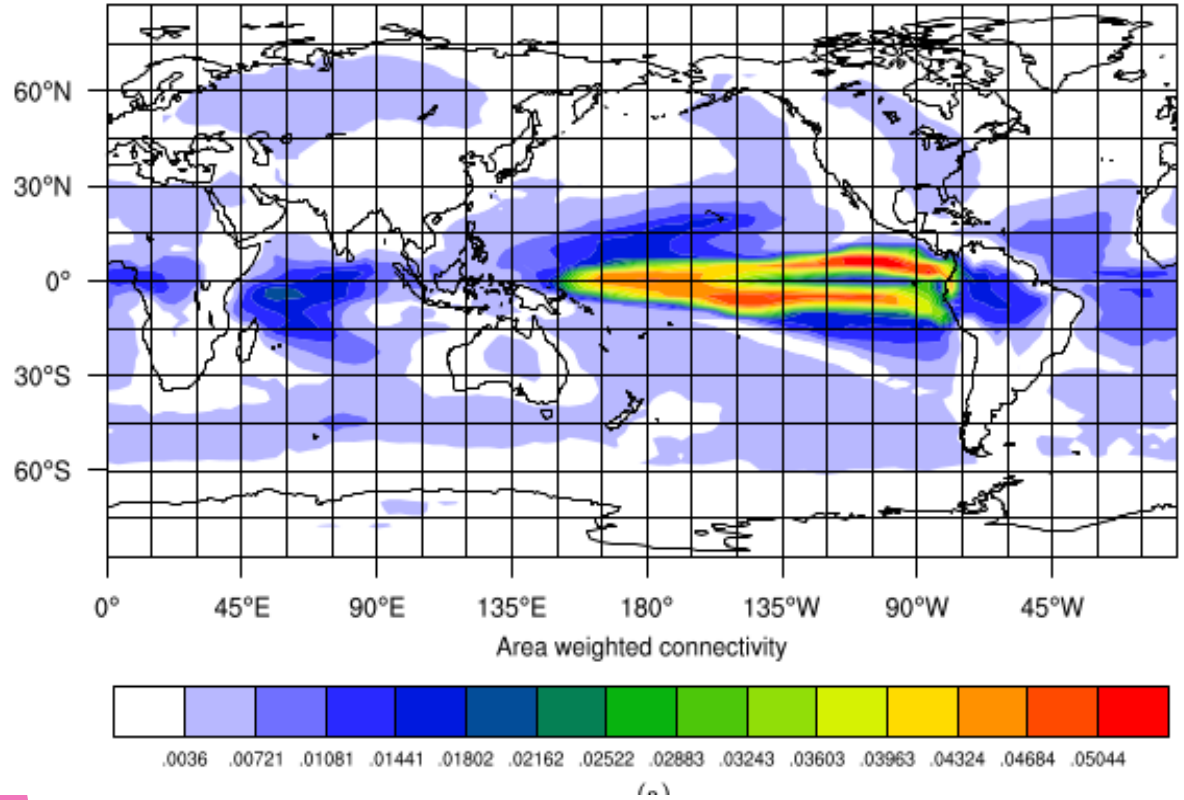
FIG. 6. Total number of links (connections) at each geographic location. The uniformity observed in the Tropics indicates that each node possesses the same number of connections. This is not the case in the extratropics where certain nodes possess more links than the rest.



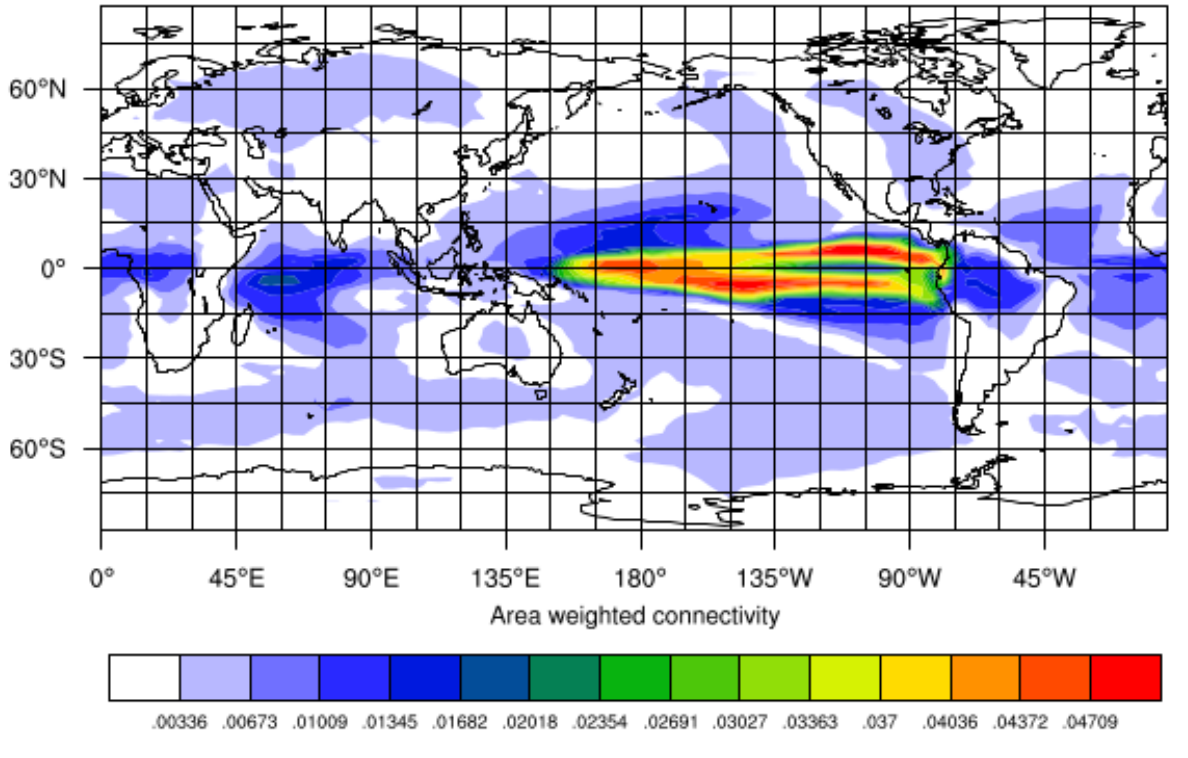
PC: Donges et al, EPJST 174 (2009)

PC: Tsonis et al,
BAMS May 2015

Geophysical Networks: Area Weighted Connectivity



Correlation



Mutual Information

PC: Donges et al, EPJST 174 (2009)

Teleconnections

- How does climatic influence pass over long distances?

- Zhou, Dong; Gozolchiani, Avi; Ashkenazy, Yosef; Havlin, Shlomo (2015). "Teleconnection Paths via Climate Network Direct Link Detection". *Physical Review Letters*. **115** (26).

- Considers lag-corrected correlations between pairs of nodes
- Separates “indirect” effects from each node’s time-series and extracts “pure” time-series for each node
- Constructs “direct” correlation network based these “pure” time-series between pairs of distant nodes
- Also identifies optimal path of influence between pairs of strongly correlated distant nodes

Teleconnections

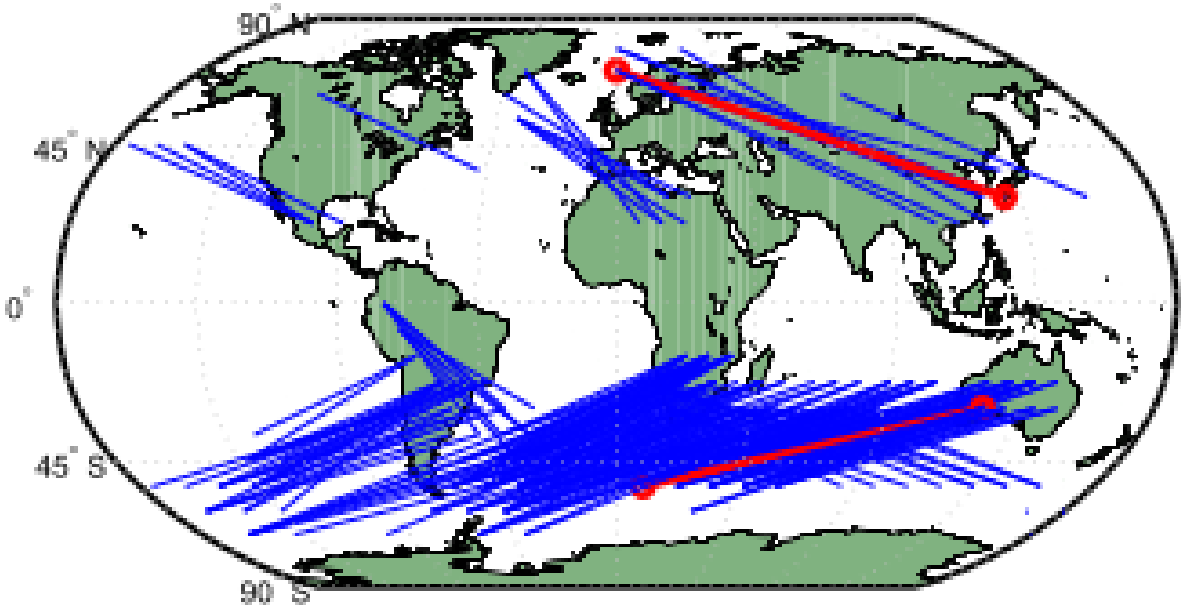
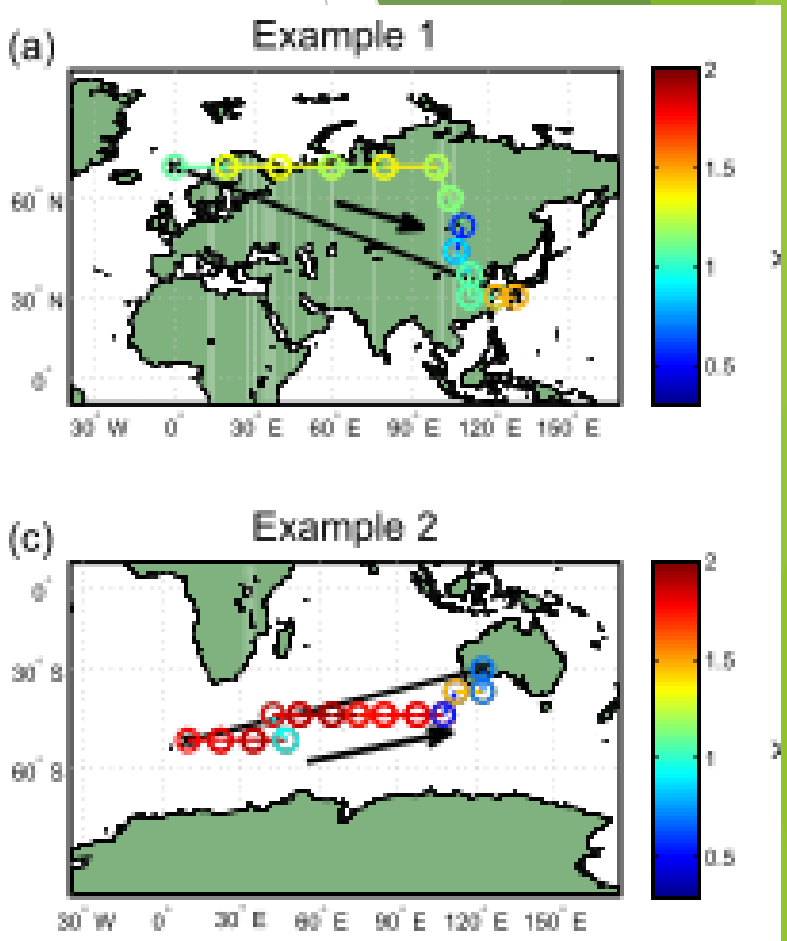


FIG. 3 (color online). 226 strong and long observed positive links which have (i) distance larger than 5000 km, (ii) $W_{i,j}^{obs,pos} \geq 9$, and (iii) latitude difference above 20°. The red circles and lines indicate the two examples considered in the text and in Fig. 4.



PC: Zhou et al, PRL 115 (2015)

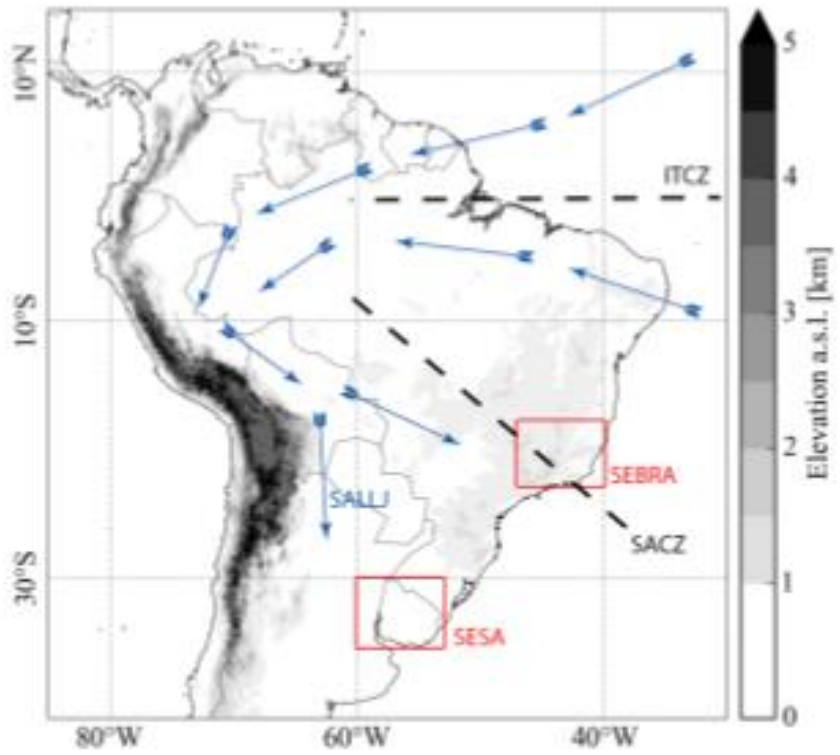
Extreme Event Synchronization

- Extreme Events at individual locations identified.
- How synchronized are such extreme events in two locations?

- N. Boers, A. Rheinwalt, B. Bookhagen, Henrique M. J. Barbosa, N. Marwan, J. Marengo and J. Kurths, 2014. “The South American rainfall dipole: A complex network analysis of extreme events”. Geophysical Research Letters, 2014

- Analyze rainfall extremes (above 90-th percentile) of a dipole over South America in Brazil and Argentina.
- Separate networks for the two phases of the dipole.
- Strength of each edge equal to number of synchronized rainfall. extremes (upto 3 days) at the connected nodes.
- Degree of each node studied during both phases of dipole.

Extreme Event Synchronization



PC: Boers et al, GRL 41 (2014)

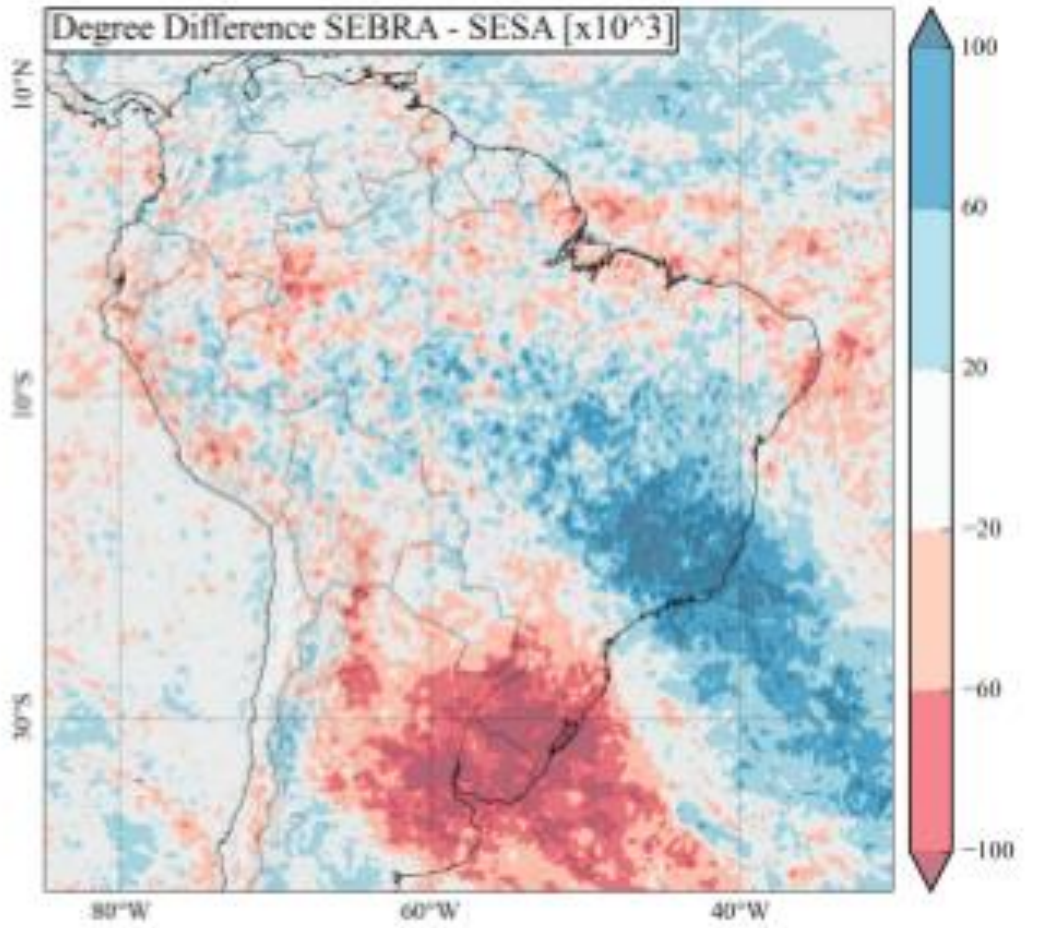


Figure 4. Difference between degree fields for the SEBRA and for the SESA phase. Note the oscillation between positive and negative values extending over the entire continent beyond the dipole between the SESA and SEBRA regions.

Other Geophysical network types exist

Other networks:

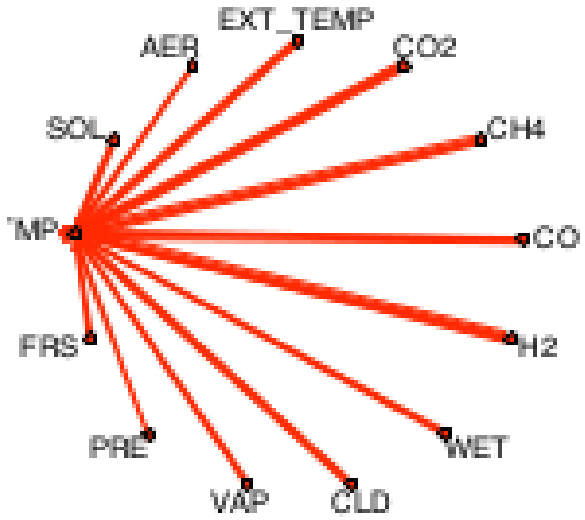
- Phase synchronization networks.
- Variety of causality-based networks.

Optional: We can discuss causal networks in more detail next time, if there is interest.

Geophysical Networks: Nodes

- Sometimes, each node can represent (location, time)
- E.g. Rainfall received by a location in the month of July of a year.

- Each node may also represent a climatic variable.
- E.g. one node for temperature, one for rainfall, one for humidity, etc.
- One instantiation of the network per location and time-step; one time-series per location.



PC: Lozano et al, KDD 2009

Time-evolving Network

- Divide the total data duration into time-slices.
- Compute the edge-weights and build the network for each time-slice
- Edge between any node-pair may be present in some slices, absent in others
- How do the network properties change across the slices?

Stochastic Rainfall Generators

16th March, 2020

AI60002

Simulation: Recap

- Process-based Models

  Try to create a mathematical model for the entire process

  Requires detailed knowledge about physics of the process

- Statistical Models

  Try to reproduce only the observable part of the process irrespective of the physics behind it

  Requires detailed knowledge about the statistical properties of the observables

Stochastic Rainfall Generator: Recap

- Create a model to generate/simulate synthetic values of rainfall
- Simulation is not same as prediction: point-wise matching is not required
- Statistical properties of the simulated values should match the corresponding properties of observed/historical values
- Stochastic: to utilize the uncertainty/noise inherent to the process
- General approach:

  Build a probabilistic model

  Create “synthetic” observations by sampling repeatedly from it

Key Statistics to be evaluated

- Proportion of wet days
- Intensity of rainfall during wet days
- Mean/max length of wet and dry spells
- Mean intensity of rainfall during wet and dry spells
- *Number of extreme rainfall events*
- The above statistics separately for each month

Key Statistics to be evaluated

- Proportion of wet days
- Intensity of rainfall during wet days
- Mean/max length of wet and dry spells
- Mean intensity of rainfall during wet and dry spells
- *Number of extreme rainfall events*
- The above statistics separately for each month
- *The above statistics separately for each location*
- *Spatial correlation across locations*

Journal of Hydrology 574 (2019) 181–192

Contents lists available at ScienceDirect

Journal of Hydrology

journal homepage: www.elsevier.com/locate/jhydrol

Research papers

A daily spatially explicit stochastic rainfall generator for a semi-arid climate

Ying Zhao[a,*], Mark A. Nearing[b], D. Phillip Guertin[a]

[a] School of Natural Resources and the Environment, University of Arizona, Tucson, AZ 85719, USA
[b] USDA-Agricultural Research Service, Southwest Watershed Research Center, Tucson, AZ 85719, USA

ARTICLE INFO

This manuscript was handled by A. Bardossy, Editor-in-Chief, with the assistance of Uwe Haberlandt, Associate Editor

Keywords:
Rainfall generator
Spatial
Semi-arid
Convective storm
Markov chain

ABSTRACT

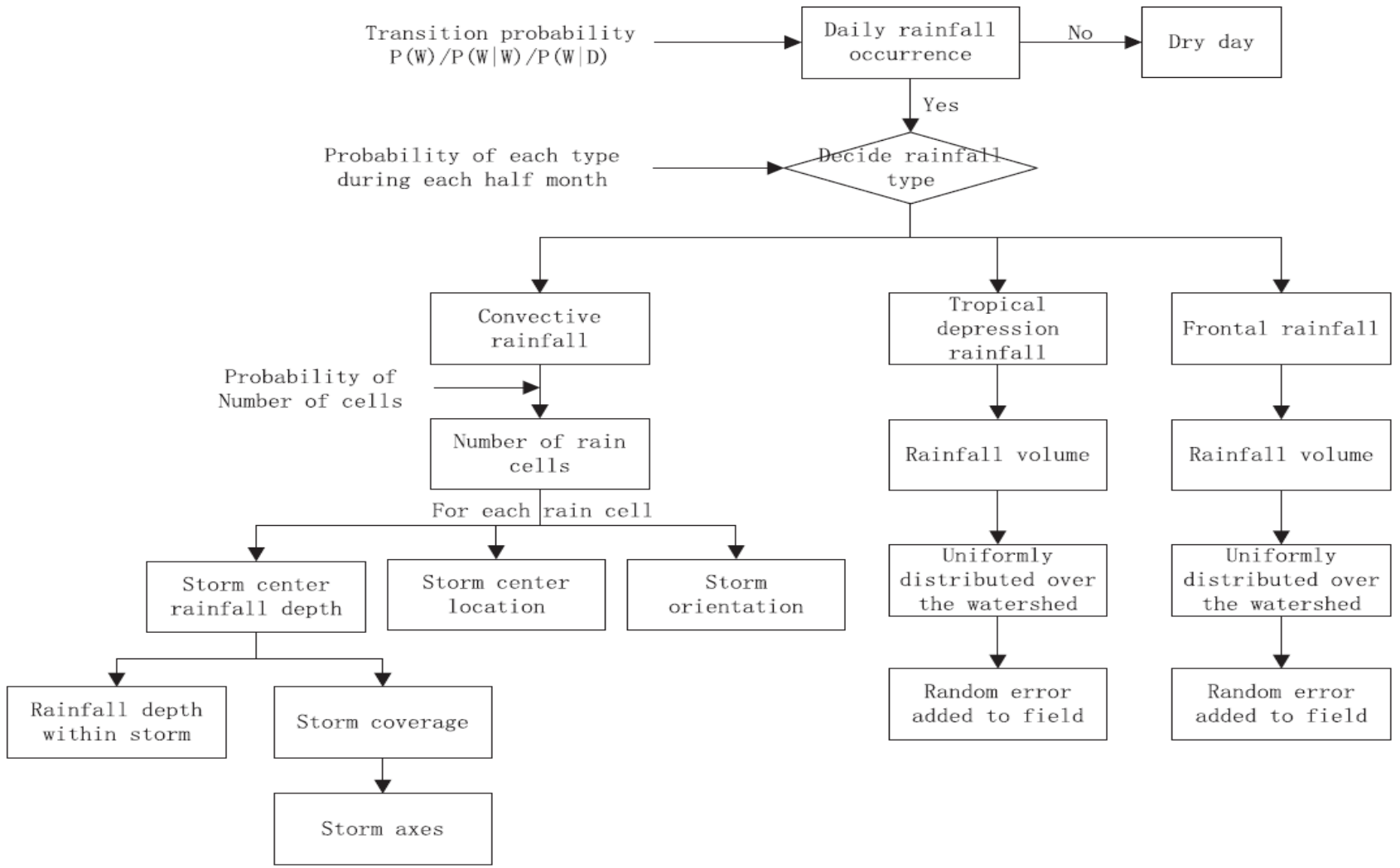
Many semi-arid regions of the world experience rainfall patterns characterized by high spatial variability. Accurate spatial representation of different types of rainfall will facilitate the application of distributed hydrological models in these areas. This study presents a daily, spatially distributed, stochastic rainfall generator based on a first-order Markov chain model, calibrated using 50 years of rainfall observations at 88 gages from 1967 through 2016 in the 148-km^2 Walnut Gulch Experimental Watershed. Three types of rainfall, including convective, frontal, and tropical depression storms, were simulated separately in the generator using biweekly parameterization. Convective storms were simulated based on an elliptical shape rain cell conceptual model, whereas frontal and tropical depression storms were simulated as uniform rainfall fields over the whole watershed with introduced random variability. The rainfall generator was evaluated by comparing the mean statistics of 30 sets of 50-year simulated data versus the 50-year rain gage observed data. Most individual storm statistics and aggregated seasonal rainfall statistics were similar to the measured rainfall observations. The long-term mean values of both summer and winter rainfall amount were statistically satisfactory. This model can serve as a guide for application in areas with convective, frontal, and tropical depression storms.

Simulation Field

- A relatively small semi-arid region in USA at altitude
- Months of July, August, and September, accounts for approximately 60% of the total annual amount of rainfall
- Frontal storms during the non-summer months account for approximately 35% of the annual precipitation.
- The remaining 5% of the annual rainfall falls in the form of tropical depression storms
- The summer rain often forms as convective storms, with relatively short duration but high intensity, and cover a limited spatial extent
- The winter frontal storms are, however, usually of long duration but low intensity, and usually cover the whole watershed more uniformly

Simulation Model

Journal of Hydrology 574 (2019) 181–192



Model Parameters

Table 1
Transition probabilities, probabilities for three types of rainfall, and the probabilities for multiple events in all 24 half month periods.

Half month		1	2	3	4	5	6	7	8	9	10	11	12
Transition probabilities	P(W)	0.2053	0.2225	0.2547	0.1914	0.1880	0.1338	0.1147	0.0960	0.0960	0.1175	0.1240	0.2640
	P(W|W)	0.4740	0.5000	0.5602	0.4776	0.4752	0.3738	0.4186	0.4167	0.3750	0.5106	0.4624	0.6111
	P(W|D)	0.1359	0.1431	0.1503	0.1237	0.1215	0.0952	0.0753	0.0619	0.0664	0.0652	0.0761	0.1377
Probabilities for types of rainfall	Convective	0	0	0	0	0	0	0	0	0	0	0	0
	Frontal	1	1	1	1	1	1	1	1	1	1	1	1
	Tropical	0	0	0	0	0	0	0	0	0	0	0	0
Probabilities for multiple events	1	0.7309	0.7155	0.6589	0.7629	0.6751	0.7312	0.6690	0.8808	0.8217	0.7066	0.7868	0.8031
	2	0.1651	0.1626	0.2321	0.1626	0.2188	0.1680	0.2242	0.1022	0.1426	0.1726	0.1745	0.1663
	3	0.1040	0.1219	0.1089	0.0745	0.1061	0.1008	0.1068	0.0170	0.0357	0.1208	0.0388	0.0306
	4												
	5												
Parameter of distributions*	μ (mm or 10^5 m^3)	2.6923	2.8794	2.1721	1.8628	2.6981	2.0318	1.5667	1.6052	1.4837	1.6541	2.3326	2.1434
	σ (mm)												

Half month		13	14	15	16	17	18	19	20	21	22	23	24
Transition probabilities	P(W)	0.6213	0.7738	0.7547	0.6438	0.5320	0.2733	0.2160	0.1950	0.1440	0.1613	0.2173	0.2288
	P(W|W)	0.7854	0.8336	0.8269	0.7592	0.7118	0.5561	0.5185	0.4615	0.4444	0.3884	0.5215	0.5191
	P(W|D)	0.3521	0.5635	0.5326	0.4316	0.3276	0.1651	0.1327	0.1304	0.0935	0.1176	0.1329	0.1410
Probabilities for types of rainfall	Convective	1	1	1	1	0.9876	0.9876	0	0	0	0	0	0
	Frontal	0	0	0	0	0	0	0.9876	0.9876	0.9876	0.9876	1	1
	Tropical	0	0	0	0	0.0124	0.0124	0.0124	0.0124	0.0124	0.0124	0	0
Probabilities for multiple events	1	0.6449	0.6499	0.6450	0.6842	0.7133	0.6966	0.7409	0.7515	0.7496	0.7596	0.7801	0.6929
	2	0.2187	0.2227	0.2171	0.2178	0.1633	0.1678	0.1766	0.1792	0.2019	0.1492	0.1367	0.1906
	3	0.0961	0.0911	0.0953	0.0713	0.0867	0.0807	0.0824	0.0694	0.0485	0.0912	0.0832	0.1165
	4	0.0362	0.0300	0.0310	0.0255	0.0250	0.0323						
	5	0.0042	0.0062	0.0115	0.0013	0.0117	0.0226						
Parameter of distributions*	μ (mm or 10^5 m^3)	1.5314	1.7461	1.6551	1.6105	1.6272	1.3189	2.4079	2.4877	2.5405	2.0375	3.2548	2.3647
	σ (mm)	1.4235	1.4680	1.4851	1.4827	1.4576	1.5344						

* (1) July–September (13–18): lognormal distribution for convective rainfall maximum depth, unit: mm. (2) Other months (1–12, 19–24): exponential distribution for frontal rainfall volume, unit: 10^5 m^3. (3) September–November (17–22): μ of exponential distribution for tropical depression rainfall is $4.2643*10^6$ m^3.

Comparisons

Table 4
Observed and simulated rainfall totals for summer months of six gages (mm).

Gage ID	Observed						Simulated					
	13	34	44	46	62	80	13	34	44	46	62	80
Mean	186.7	192.2	194.6	199.5	194.5	189.7	196.3	187.1	185.8	191.8	186.5	195.0
Std. dev.	60.4	63.7	58.6	61.4	52.2	66.8	100.4	99.7	101.6	102.6	96.3	100.7
Max	336.6	345.7	345.9	410.5	327.3	380.0	508.3	561.7	623.5	617.3	534.2	511.9
Min	89.8	70.2	81.0	77.7	88.8	75.4	3.6	1.5	7.2	8.2	10.4	9.1
Range	246.8	275.5	264.9	332.7	238.5	304.5	504.8	560.2	616.3	609.1	523.9	502.8
Skewness	0.5	0.4	0.4	0.5	−0.2	0.6	0.5	0.6	1.0	0.9	0.6	0.6

Table 5
Observed and simulated rainfall totals for non-summer months of six gages (mm).

Gage ID	Observed						Simulated					
	13	34	44	46	62	80	13	34	44	46	62	80
Mean	122.7	120.7	121.6	132.8	116.9	120.9	122.6	122.0	122.0	122.7	121.6	122.1
Std. dev.	59.7	65.8	61.1	65.1	64.3	59.8	34.2	34.1	33.7	34.7	34.0	34.3
Max	266.4	308.9	295.0	318.8	300.4	282.8	265.0	305.8	282.0	280.8	264.7	256.8
Min	19.8	18.0	13.2	16.3	10.7	12.2	42.3	41.9	42.3	33.2	44.7	30.3
Range	246.6	290.8	281.8	302.5	289.7	270.6	222.7	263.9	239.8	247.6	219.9	226.5
Skewness	0.6	0.9	0.9	0.8	1.2	0.7	0.5	0.6	0.4	0.5	0.6	0.4

Observed and simulated median length of dry and wet spells (day).

	Annual	Summer	Non-summer		Annual	Summer	Non-summer
Dry_observed	4.2	2.2	7.2	Wet_observed	1.0	1.0	1.0
Dry_simulated	4.0	2.0	6.0	Wet_simulated	2.0	3.0	1.0

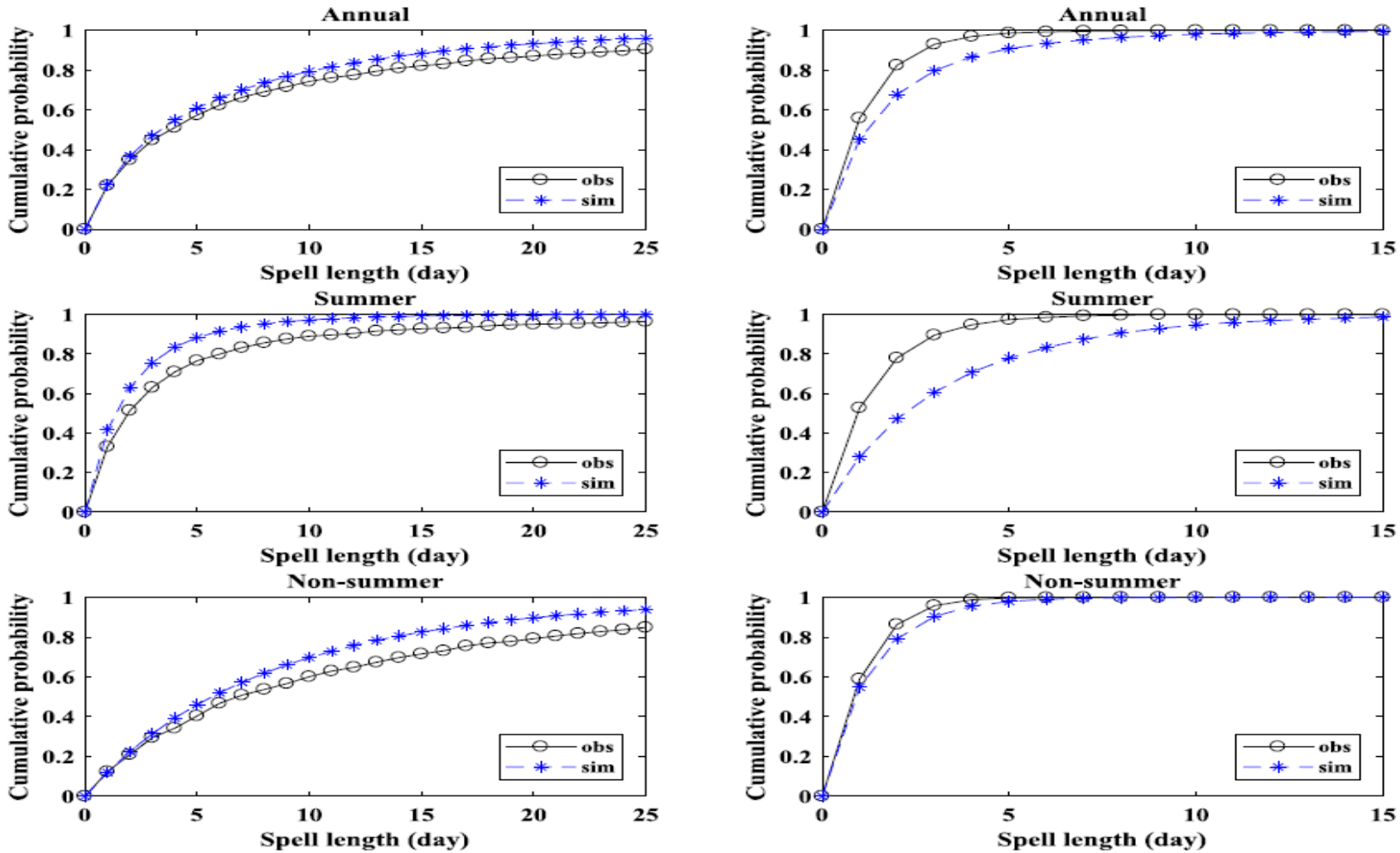


Fig. 7. CDFs of observed and simulated dry and wet spell length for annual, summer and non-summer periods, (1) first column: dry spell, (2) second column: wet spell.

Stoch Environ Res Risk Assess (2015) 29:347–356
DOI 10.1007/s00477-014-0911-6

ORIGINAL PAPER

Coupled stochastic weather generation using spatial and generalized linear models

Andrew Verdin · Balaji Rajagopalan · William Kleiber · Richard W. Katz

Published online: 5 July 2014

Abstract We introduce a stochastic weather generator for the variables of minimum temperature, maximum temperature and precipitation occurrence. Temperature variables are modeled in vector autoregressive framework, conditional on precipitation occurrence. Precipitation occurrence arises via a probit model, and both temperature and occurrence are spatially correlated using spatial Gaussian processes. Additionally, local climate is included by spatially varying model coefficients, allowing spatially evolving relationships between variables. The method is illustrated on a network of stations in the Pampas region of Argentina where nonstationary relationships and historical spatial correlation challenge existing approaches.

Keywords Spatial correlation · Pampas · Precipitation · Temperature · Weather simulation

ensembles of input sequences, which are typically daily weather, resulting in ensembles of system variables and their probability density functions that provide estimates of risk that are useful for decision making. Historic data is often limited in space and time hence the risk estimates based solely on them do not accurately reflect the underlying variability. Therefore, robust generation of weather sequences that capture the underlying variability is essential. Generating random weather sequences that are statistically consistent with historical observations is known as stochastic weather generation.

Crop models for agriculture planning, hydrologic models for generating streamflow needed for water resources management, and erosion models for land erosion management (Wallis and Griffiths 1997; Richardson 1981; Richardson and Wright 1984; Wilks 1998; Wilks and Wilby 1999; Friend et al. 1997) have motivated the development of stochastic weather generators over the

Stochastic Model - temperature

- Condition the bivariate temperature process on precipitation occurrence.
- Precipitation largely occurs due to large scale atmospheric movement, while surface temperatures are highly controlled by local climate factors and by whether or not precipitation occurs

$$Z_N(s,t) = \beta_N(s)'\mathbf{X}_N(s,t) + W_N(s,t)$$
$$Z_X(s,t) = \beta_X(s)'\mathbf{X}_X(s,t) + W_X(s,t).$$

- The first component is a local regression on some covariate vector X
- Regression parameters β are specific to location
- The weather component (denoted by W for weather) generates variability and spatial correlation via a multivariate normal Gaussian process.

Stochastic Model - rainfall

- The precipitation process is broken into two components: the occurrence O(s,t), and the intensity or amount, A(s,t) at location s on day t.
- Occurrence process is modelled as a probit: $O(\boldsymbol{s},t) = \mathbb{1}_{[W_O(\boldsymbol{s},t)\geq 0]}$

  where the latent process Wo(s,t) is a Gaussian Process.

  If the latent process is positive, it rains at location s, else it doesn't
- The latent process has mean function that is a regression on some covariates
- Rainfall intensity is spatially correlated by imposing a zero-mean Gaussian process WA(s,t) with covariance function CA(h,t) $A(\boldsymbol{s},t) = G_{s,t}^{-1}(\Phi(W_A(\boldsymbol{s},t)))$

Modeling Choices

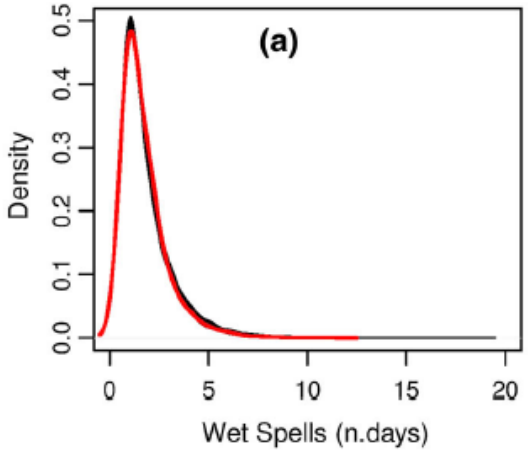
- For each of max, min temperature and precipitation intensity we have a covariate vector

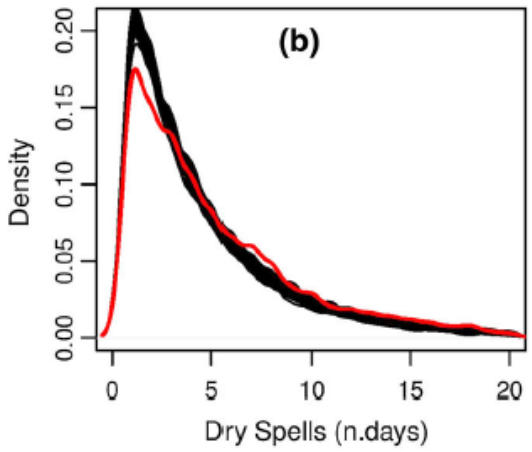
$$\mathbf{X}_N(\boldsymbol{s},t) = (1, \cos(2\pi t/365), \sin(2\pi t/365), r(t), Z_N(\boldsymbol{s},t-1), Z_X(\boldsymbol{s},t-1), O(\boldsymbol{s},t))'.$$

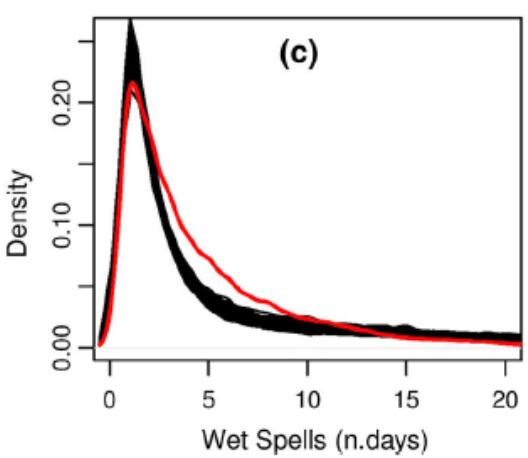
$$\mathbf{X}_O(\boldsymbol{s},t) = (1, \cos(2\pi t/365), \sin(2\pi t/365), O(\boldsymbol{s},t-1))',$$

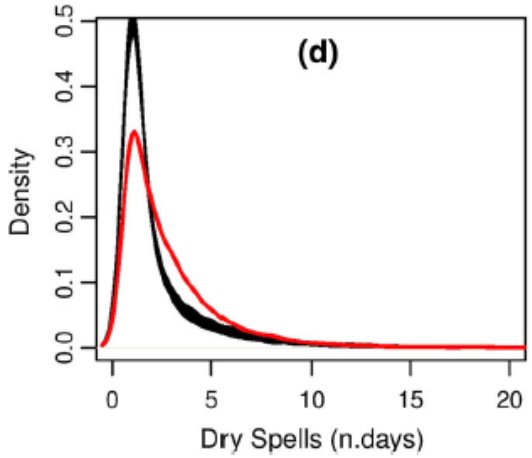
- The linear coefficients β are estimated through least-square regression
- Different covariance functions can be tried out for the Gaussian Processes

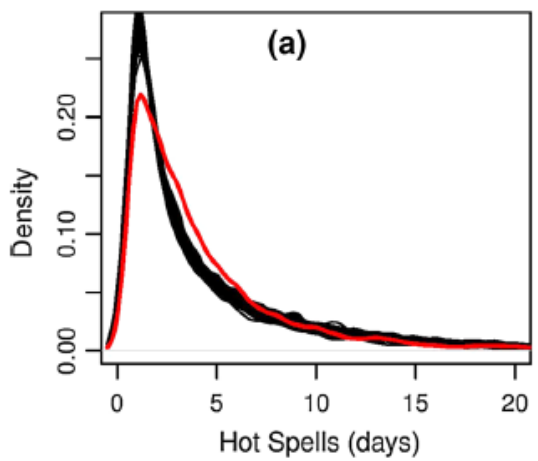
Results

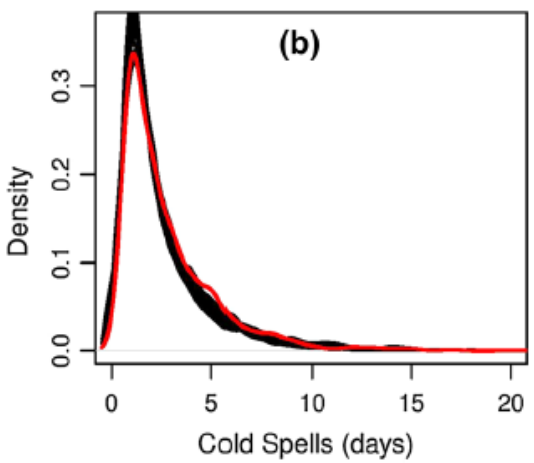

(a)
Density
Wet Spells (n.days)

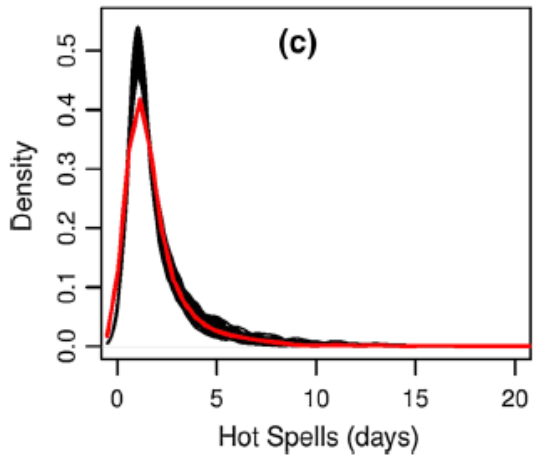

(b)
Density
Dry Spells (n.days)

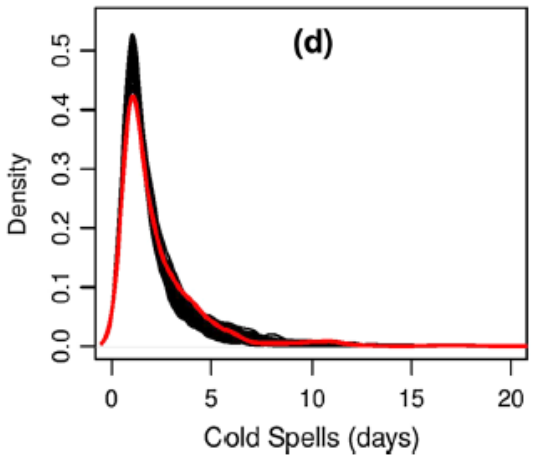

(c)
Density
Wet Spells (n.days)


(d)
Density
Dry Spells (n.days)

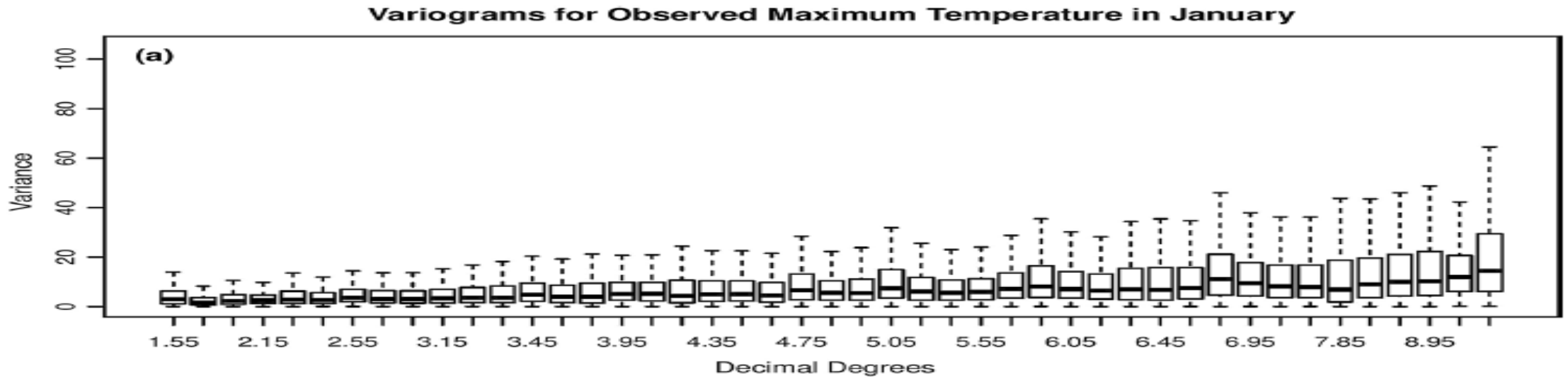

(a)
Density
Hot Spells (days)

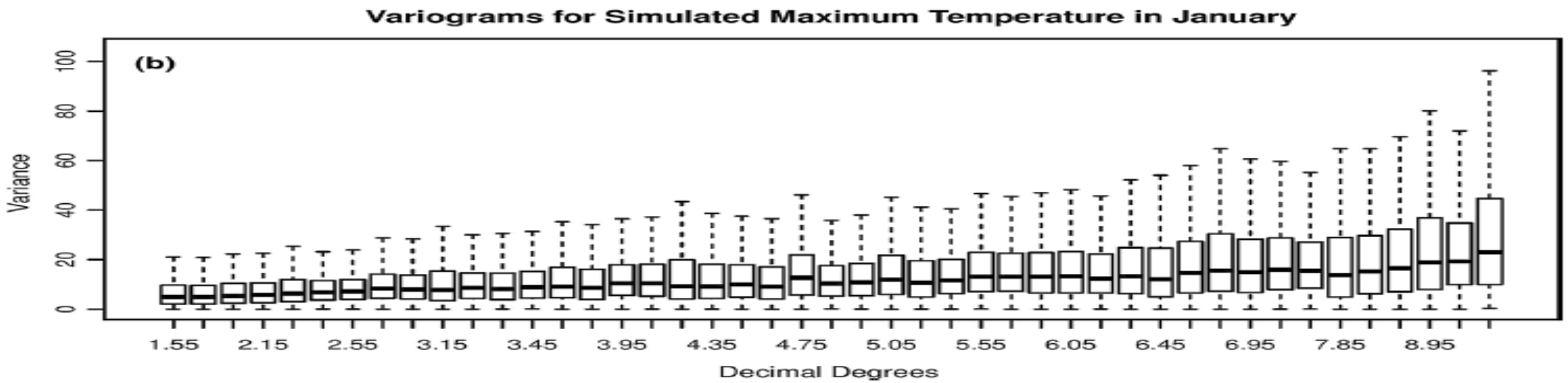

(b)
Density
Cold Spells (days)


(c)
Density
Hot Spells (days)


(d)
Density
Cold Spells (days)

Results


Variograms for Observed Maximum Temperature in January
(a)
Variance
0
20
40
60
80
100
1.55
2.15
2.55
3.15
3.45
3.95
4.35
4.75
5.05
5.55
6.05
6.45
6.95
7.85
8.95
Decimal Degrees


Variograms for Simulated Maximum Temperature in January
(b)
Variance
0
20
40
60
80
100
1.55
2.15
2.55
3.15
3.45
3.95
4.35
4.75
5.05
5.55
6.05
6.45
6.95
7.85
8.95
Decimal Degrees

Neural Networks for Nowcasting of Weather

AI60002 Lecture 27 April

Nowcasting

- Forecasting at short time ranges (order of hours)
- Usually done for precipitation, thunderstorms, lightnings etc
- Real-time performance very important
- Spatial and temporal precision very important
- Usually based on observations from satellites, radars, automatic weather station
- Usually done with numerical models – slow and computationally expensive
- Can machine learning help?

Short-term Rainfall Forecasting Using Multi-layer Perceptron

Pengcheng Zhang, Yangyang Jia, Jerry Gao, Wei Song, Hareton Leung

Abstract—Rainfall forecasting is crucial in the field of meteorology and hydrology. However, existing solutions always achieve low prediction accuracy for short-term rainfall forecasting. Numerical forecasting models perform worse in many conditions. Machine learning approaches neglect the influences of physical factors in upstream or downstream regions, which make forecasting accuracy fluctuate in different areas. To improve the overall forecasting accuracy for short-term rainfall, this paper proposes a novel solution called Dynamic Regional Combined short-term rainfall Forecasting approach (DRCF) using Multi-layer Perceptron (MLP). First, Principal Component Analysis (PCA) is used to reduce the dimension of thirteen physical factors, which serves as the input of MLP. Second, a greedy algorithm is applied to determine the structure of MLP. The surrounding sites are perceived based on the forecasting site. Finally, to solve the clutter interference which is caused by the extension of the perception range, DRCF is enhanced with several dynamic strategies. Experiments are conducted on data from 56 real-world meteorology sites in China, and we compare DRCF with atmospheric models and other machine learning approaches. The experimental results show that DRCF outperforms existing approaches in both threat score (TS) and root mean square error (RMSE).

Index Terms—Rainfall forecast, deep neural network, multi-layer perceptron, short-term, atmospheric models

Multi-layer Perceptron for Rainfall Nowcasting

3.3 Adjustment and Constructive Algorithm of MLP

MLP is a kind of feed-forward network. It is calculated from the input layer to the output layer successively. Each node at the same level is calculated at the same time, and it does not interfere with each other [42]. The value of every node is equal to the weighted summation of all nodes in the previous layer. This calculation process is called the feed-forward process of MLP. If there is a MLP which contains m hidden layers, its input and output dimensions are respectively equal to n_1 and n_{m+2}. The number of nodes in each hidden layer is n_2, n_3, ..., n_{m+1}, respectively. In the feed-forward process of this MLP, each node value is calculated using the following formula:

$$x_{ij} = f\left(W_i X_{i-1} + b_{i-1}\right) \tag{2}$$

where X_{ij} represents the value of the j neuron in the i layer. W_i represents the weight vector of the j neurons in layer $i-1$ to layer i. X_{i-1} represents the value vector of all neurons in layer $i-1$. b_{i-1} represents the bias of the $i-1$ layer, and f is the activation function.

MLP is a supervised learning algorithm. There is an ideal output corresponding to any input. The loss function of the ideal output and the actual value is defined as:

$$J\left(W, b; x, y\right) = \frac{1}{2}\left\|h_{W,b(x)} - y\right\|^2 \tag{4}$$

where h represents the output value, y represents the actual value, and $||.||$ represents any distance norm.

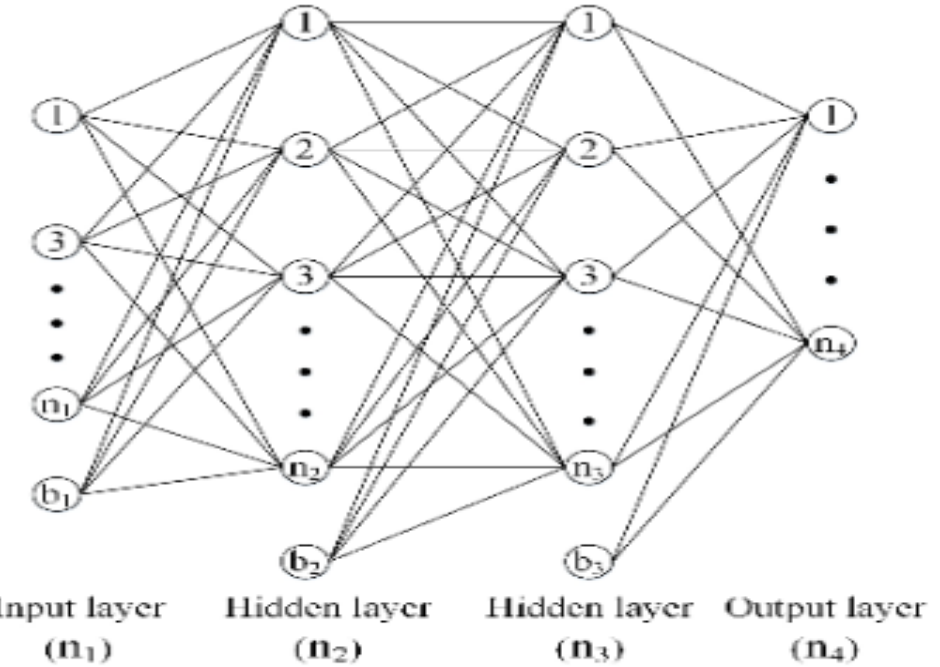


Fig. 1. Structure of MLP

we use Stochastic gradient descent (SGD) with momentum to adjust the parameters of MLP. The calculation method of gradient and SGD with momentum are described as follows:

$$\nabla W = -\frac{\partial J(W, b; x, y)}{\partial W} \tag{5}$$

$$\delta W_t = \alpha \nabla W_t + \beta \delta W_{t-1} \tag{6}$$

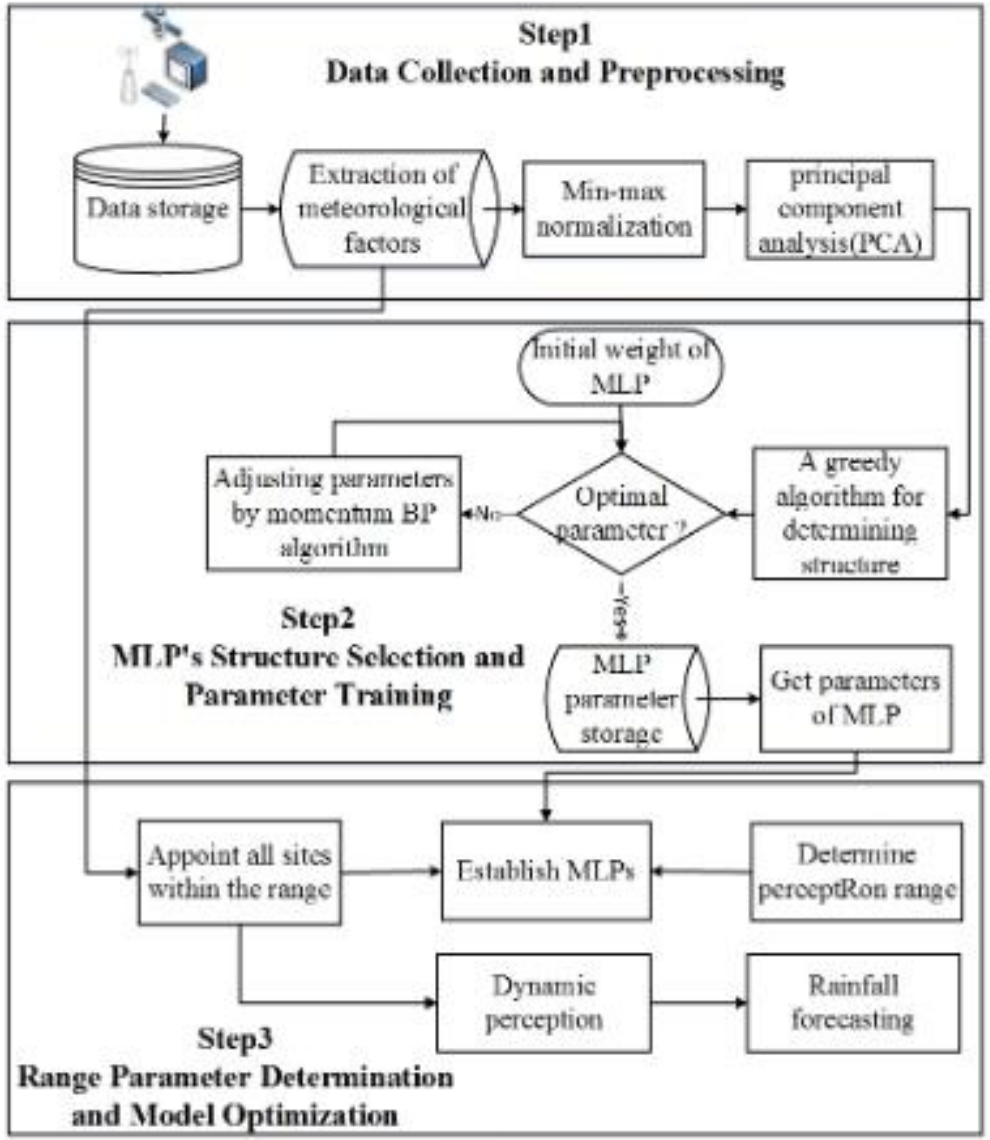


TABLE 1
A list of factors used in the model

Factor	Input value
500hPa height(X_1)	Forecast area value
500hPa temperature(X_2)	Forecast area value
500hPa temperature dew point difference(X_3)	Forecast area value
500hPa wind direction(X_4)	Forecast area value
500hPa wind speed(X_5)	Forecast area value
Total cloud amount(X_6)	Forecast area value – neighbouring area value
Surface wind speed(X_7)	Forecast area value – neighbouring area value
Surface wind direction(X_8)	Forecast area value – neighbouring area value
Surface air pressure(X_9)	Forecast area value – neighbouring area value
Surface 3 hour pressure change(X_{10})	Forecast area value – neighbouring area value
Surface temperature dew point difference(X_{11})	Forecast area value – neighbouring area value
Surface temperature(X_{12})	Forecast area value – neighbouring area value
Rainfall over past 3 hours(X_{13})	Neighbouring area value

Nowcasting based on Satellite and Radar Data

- Satellites and Radars do not directly measure atmospheric variables
- These variables need to be “derived” from their outputs
- The “derivation” is usually a complex mapping
- There may be disagreements between the measurements from different sources
- Can machine learning help us to estimate the correct values from multiple sources?

Geophysical Research Letters

RESEARCH LETTER
10.1029/2019GL084771

Key Points:

- Conventional parametric relationships between radar reflectivity Z and rain rate R are not sufficient to capture precipitation variabilities
- A hybrid deep neural network system is designed for improved space radar rainfall estimation

Supporting Information:

- Supporting Information S1

Correspondence to:
H. Chen,
haonan.chen@noaa.gov

Citation:
Chen, H., Chandrasekar, V., Tan, H., & Cifelli, R. (2019). Rainfall estimation

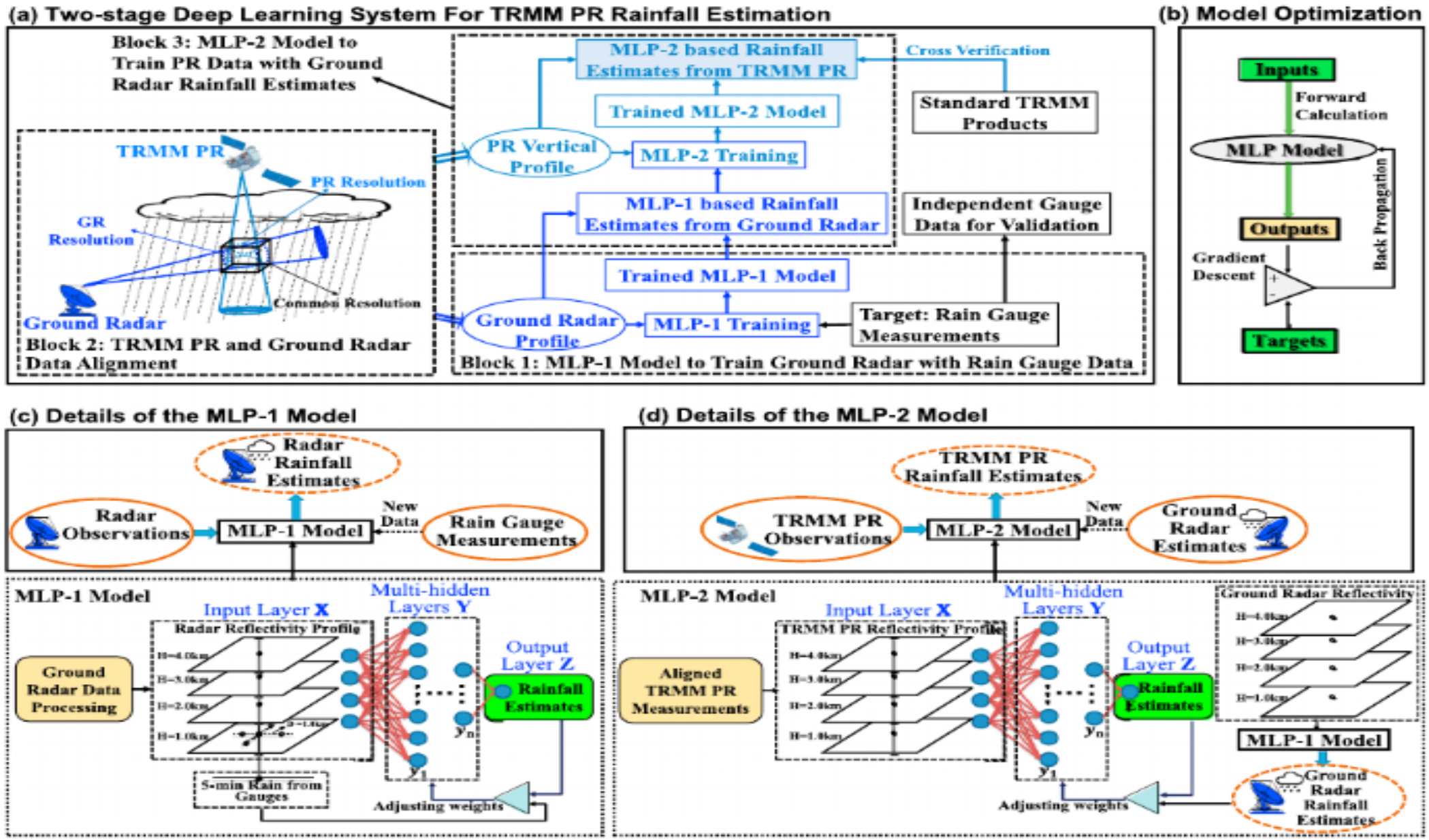
Rainfall Estimation From Ground Radar and TRMM Precipitation Radar Using Hybrid Deep Neural Networks

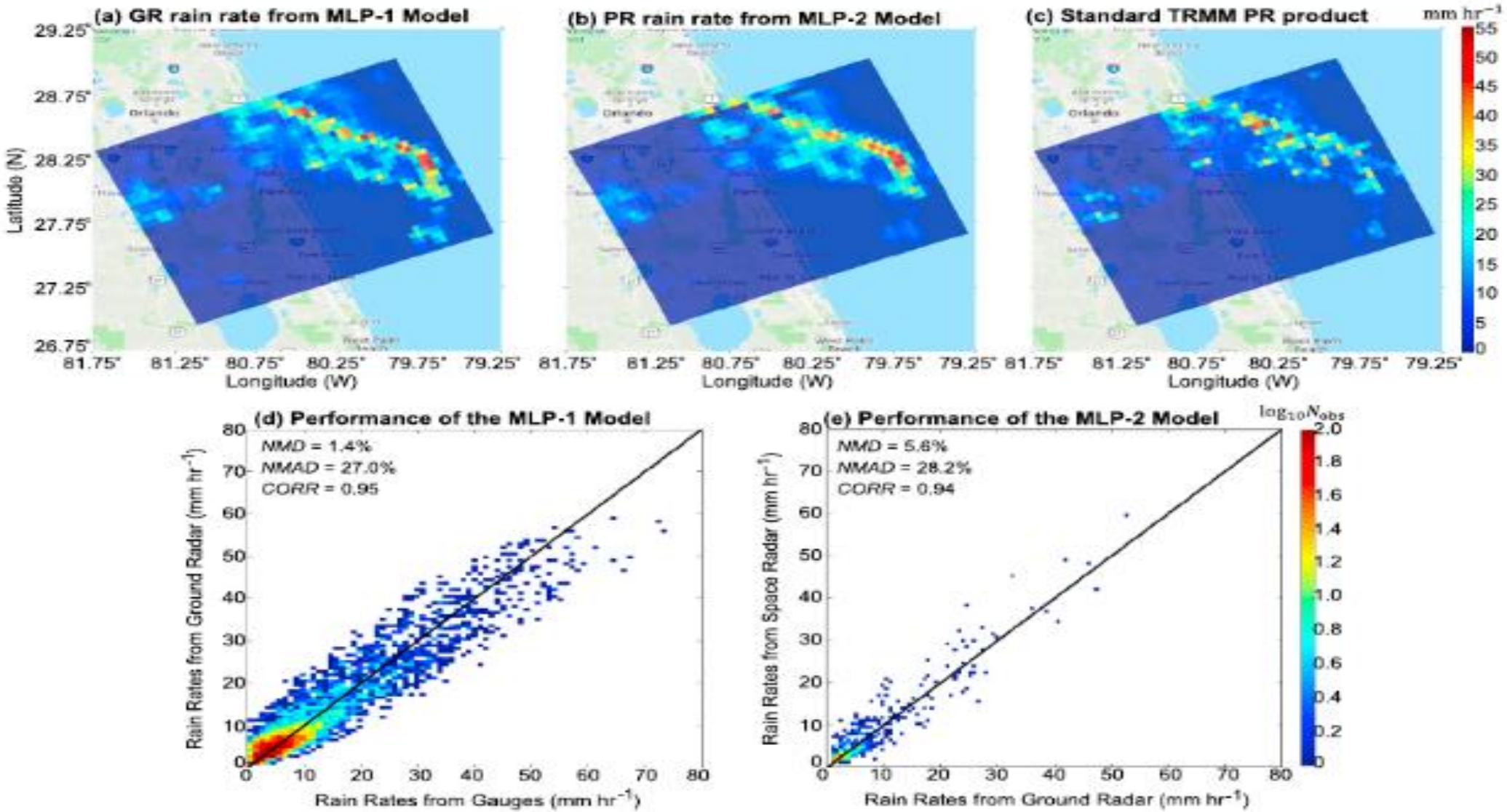
Haonan Chen[1,2], V. Chandrasekar[1], Haiming Tan[1], and Robert Cifelli[2]

[1]Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO, USA, [2]NOAA/Earth System Research Laboratory, Boulder, CO, USA

Abstract Remote sensing of precipitation is critical for regional, continental, and global water and climate research. This study develops a deep learning mechanism to link between point-wise rain gauge measurements, ground-based, and spaceborne radar reflectivity observations. Two neural network models are designed to construct a hybrid rainfall system, where the ground radar is used to bridge the scale gaps between rain gauge and satellite. The first model is trained for ground radar using rain gauge data as target labels, whereas the second model is for spaceborne Tropical Rainfall Measuring Mission (TRMM) Precipitation Radar (PR) using ground radar estimates as training labels. Data from 1 year of observations in Florida during 2009 are utilized to illustrate the application of this hybrid rainfall system. Validation using independent data in 2009, as well as 2-year comparison against the standard PR products, demonstrates the promising performance and generality of this innovative rainfall algorithm.

Plain Language Summary The Tropical Rainfall Measuring Mission (TRMM) Precipitation Radar (PR) was the first spaceborne active sensor for observing precipitation over the tropics and subtropics. During its 17 years (1997–2014) in orbit and beyond, PR has been an important tool to characterize tropical precipitation microphysics and quantify rainfall rate over the globe. Ground validation is a critical component in the development of TRMM products. However, the ground-based sensors have different characteristics from PR in terms of resolution, viewing angle, and uncertainties in the sensing environments, which are not taken into account in the operational parametric rainfall relations applied to PR measurements. This study develops a nonparametric machine learning technique for PR rainfall estimation. In the regions where substantial gauge and ground radar data are available, this approach can produce better rainfall estimates compared to the standard PR algorithm. In areas such as ocean and remote regions where no gauge or radar available, the proposed rainfall algorithm is easy to implement, and it can still produce reasonable estimates. With more and more gauges and radars being deployed and many of them become operational, this algorithm can be trained at different locations represented by different atmosphere properties to further improve the performance and generality.


(a) Two-stage Deep Learning System For TRMM PR Rainfall Estimation
Block 3: MLP-2 Model to Train PR Data with Ground Radar Rainfall Estimates
MLP-2 based Rainfall Estimates from TRMM PR
Cross Verification
Trained MLP-2 Model
Standard TRMM Products
PR Vertical Profile
MLP-2 Training
MLP-1 based Rainfall Estimates from Ground Radar
Independent Gauge Data for Validation
Trained MLP-1 Model
Ground Radar Profile
MLP-1 Training
Target: Rain Gauge Measurements
TRMM PR
PR Resolution
GR Resolution
Common Resolution
Ground Radar
Block 2: TRMM PR and Ground Radar Data Alignment
Block 1: MLP-1 Model to Train Ground Radar with Rain Gauge Data
(b) Model Optimization
Inputs
Forward Calculation
MLP Model
Outputs
Back Propagation
Gradient Descent
Targets
(c) Details of the MLP-1 Model
Radar Rainfall Estimates
Radar Observations
MLP-1 Model
New Data
Rain Gauge Measurements
MLP-1 Model
Input Layer X
Radar Reflectivity Profile
Multi-hidden Layers Y
Output Layer Z
Rainfall Estimates
Ground Radar Data Processing
S-min Rain from Gauges
Adjusting weights
(d) Details of the MLP-2 Model
TRMM PR Rainfall Estimates
TRMM PR Observations
MLP-2 Model
New Data
Ground Radar Estimates
MLP-2 Model
Input Layer X
TRMM PR Reflectivity Profile
Multi-hidden Layers Y
Output Layer Z
Rainfall Estimates
Aligned TRMM PR Measurements
Adjusting weights
Ground Radar Reflectivity
MLP-1 Model
Ground Radar Rainfall Estimates


(a) GR rain rate from MLP-1 Model
(b) PR rain rate from MLP-2 Model
(c) Standard TRMM PR product
mm hr^{-1}
55
50
45
40
35
30
25
20
15
10
5
0
Latitude (N)
29.25°
28.75°
28.25°
27.75°
27.25°
26.75°
81.75° 81.25° 80.75° 80.25° 79.75° 79.25°
Longitude (W)
Orlando
(d) Performance of the MLP-1 Model
NMD = 1.4%
NMAD = 27.0%
CORR = 0.95
Rain Rates from Ground Radar (mm hr^{-1})
Rain Rates from Gauges (mm hr^{-1})
(e) Performance of the MLP-2 Model
NMD = 5.6%
NMAD = 28.2%
CORR = 0.94
Rain Rates from Space Radar (mm hr^{-1})
Rain Rates from Ground Radar (mm hr^{-1})
$\log_{10}N_{obs}$
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0

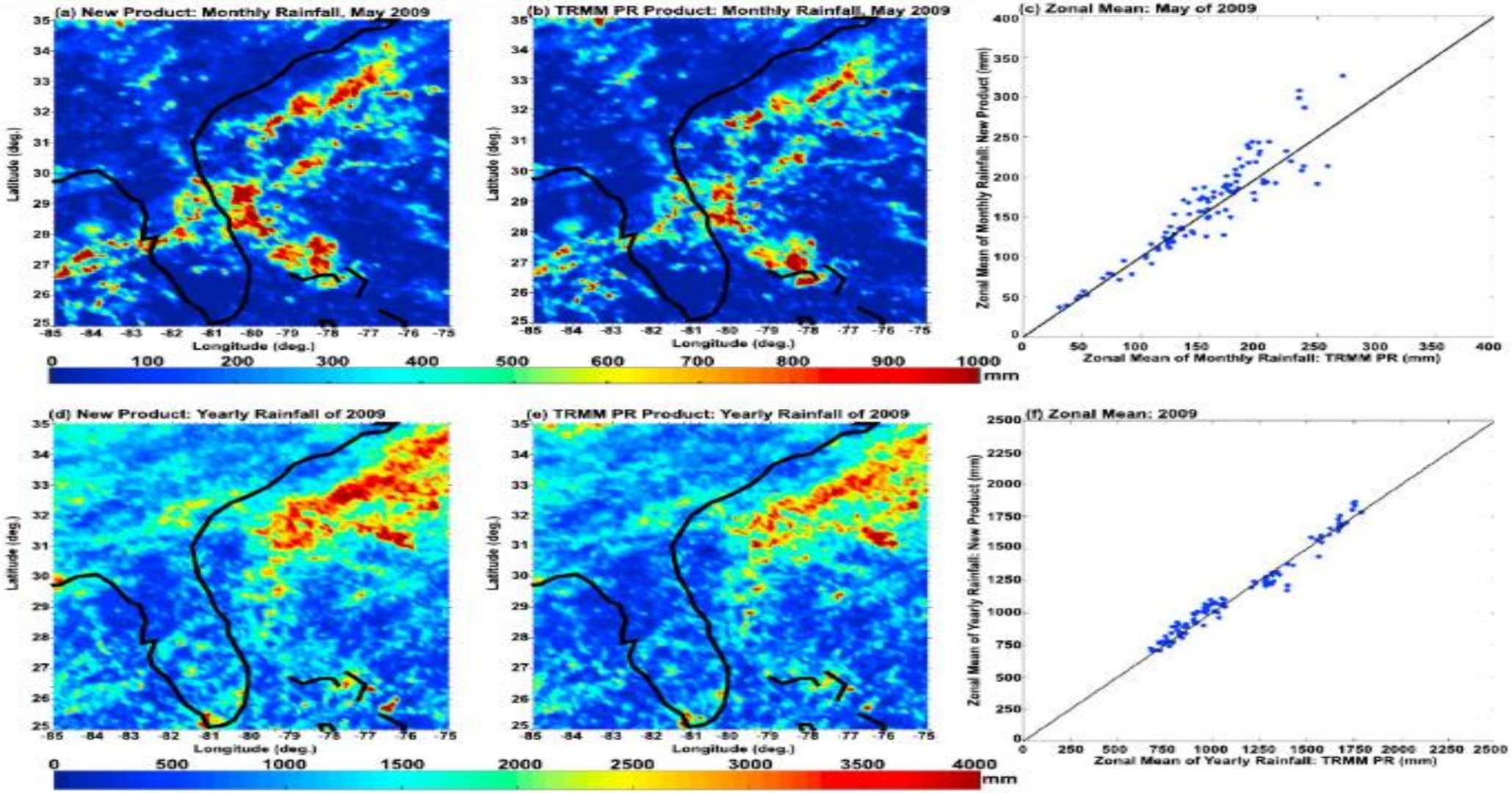


Figure 3. Sample monthly and yearly rainfall product for the region near Melbourne, FL, in 2009. Monthly rainfall map (May) derived using (a) the hybrid neural network system and (b) the standard Tropical Rainfall Measuring Mission (TRMM) Precipitation Radar (PR)product (i.e., 3A26); yearly rainfall derived using (d) the hybrid neural network system and (e) the standard TRMM PR product; (c) and (f) are the scatter plots of zonal means of rainfall estimates in (a) and (b) and (d) and (e), respectively.

Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting

Xingjian Shi **Zhourong Chen** **Hao Wang** **Dit-Yan Yeung**
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
{xshiab,zchenbb,hwangaz,dyyeung}@cse.ust.hk

Wai-kin Wong **Wang-chun Woo**
Hong Kong Observatory
Hong Kong, China
{wkwong,wcwoo}@hko.gov.hk

Abstract

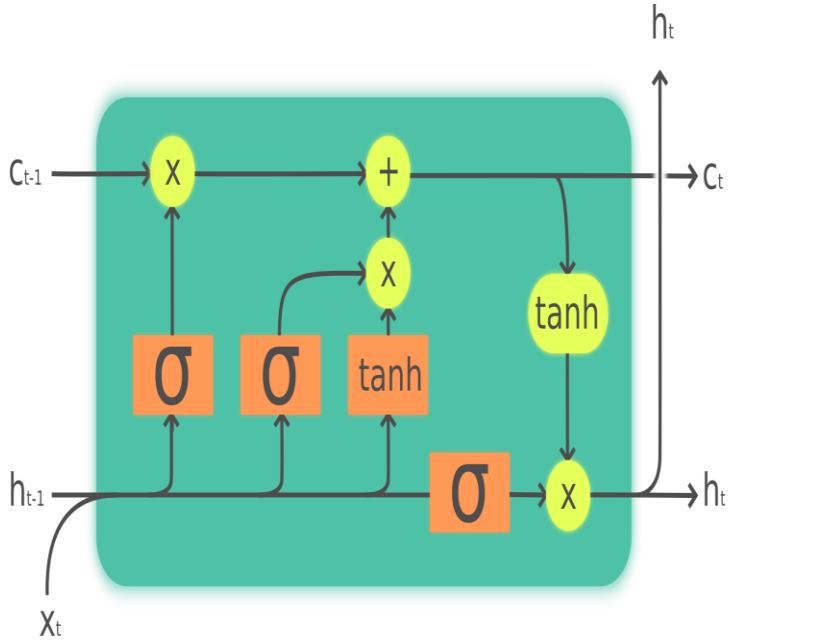
The goal of precipitation nowcasting is to predict the future rainfall intensity in a local region over a relatively short period of time. Very few previous studies have examined this crucial and challenging weather forecasting problem from the machine learning perspective. In this paper, we formulate precipitation nowcasting as a spatiotemporal sequence forecasting problem in which both the input and the prediction target are spatiotemporal sequences. By extending the *fully connected LSTM* (FC-LSTM) to have convolutional structures in both the input-to-state and state-to-state transitions, we propose the *convolutional LSTM* (ConvLSTM) and use it to build an end-to-end trainable model for the precipitation nowcasting problem. Experiments show that our ConvLSTM network captures spatiotemporal correlations better and consistently outperforms FC-LSTM and the state-of-the-art operational ROVER algorithm for precipitation nowcasting.

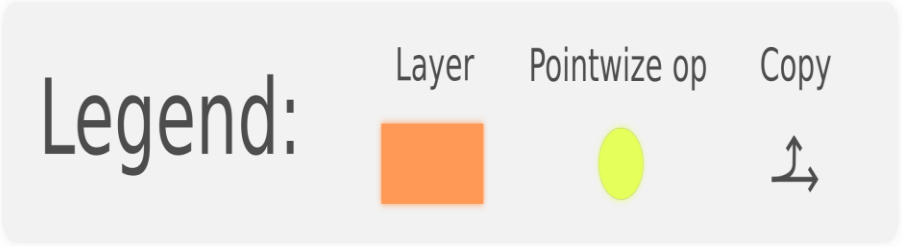
Suppose we observe a dynamical system over a spatial region represented by an $M \times N$ grid which consists of M rows and N columns. Inside each cell in the grid, there are P measurements which vary over time. Thus, the observation at any time can be represented by a tensor $\mathcal{X} \in \mathbb{R}^{P \times M \times N}$, where $\mathbb{R}$ denotes the domain of the observed features. If we record the observations periodically, we will get a sequence of tensors $\hat{\mathcal{X}}_1, \hat{\mathcal{X}}_2, \ldots, \hat{\mathcal{X}}_t$. The spatiotemporal sequence forecasting problem is to predict the most likely length-K sequence in the future given the previous J observations which include the current one:

$$\tilde{\mathcal{X}}_{t+1}, \ldots, \tilde{\mathcal{X}}_{t+K} = \underset{\mathcal{X}_{t+1}, \ldots, \mathcal{X}_{t+K}}{\arg\max} \; p(\mathcal{X}_{t+1}, \ldots, \mathcal{X}_{t+K} \mid \hat{\mathcal{X}}_{t-J+1}, \hat{\mathcal{X}}_{t-J+2}, \ldots, \hat{\mathcal{X}}_t) \qquad (1)$$

For precipitation nowcasting, the observation at every timestamp is a 2D radar echo map. If we divide the map into tiled non-overlapping patches and view the pixels inside a patch as its measurements (see Fig. 1), the nowcasting problem naturally becomes a spatiotemporal sequence forecasting problem.

We note that our spatiotemporal sequence forecasting problem is different from the one-step time series forecasting problem because the prediction target of our problem is a sequence which contains both spatial and temporal structures. Although the number of free variables in a length-K sequence can be up to $O(M^K N^K P^K)$, in practice we may exploit the structure of the space of possible predictions to reduce the dimensionality and hence make the problem tractable.





LSTM for Sequence-to-sequence prediction

- Seq2Seq: takes in a sequence as input, converts it into an “intermediate representation”, generates output sequence from it
- Input Sequence -> Encoder Network -> Intermediate Representation -> Output Network -> Output sequence
- Recurrent Neural Network: maintains a “hidden state” as memory unit, which is updated with every step in the input sequence
- Older values get updated with newer values, long sequences not handles well
- LSTM: has “hidden state” as short-term memory, and “Cell state” as long-term memory

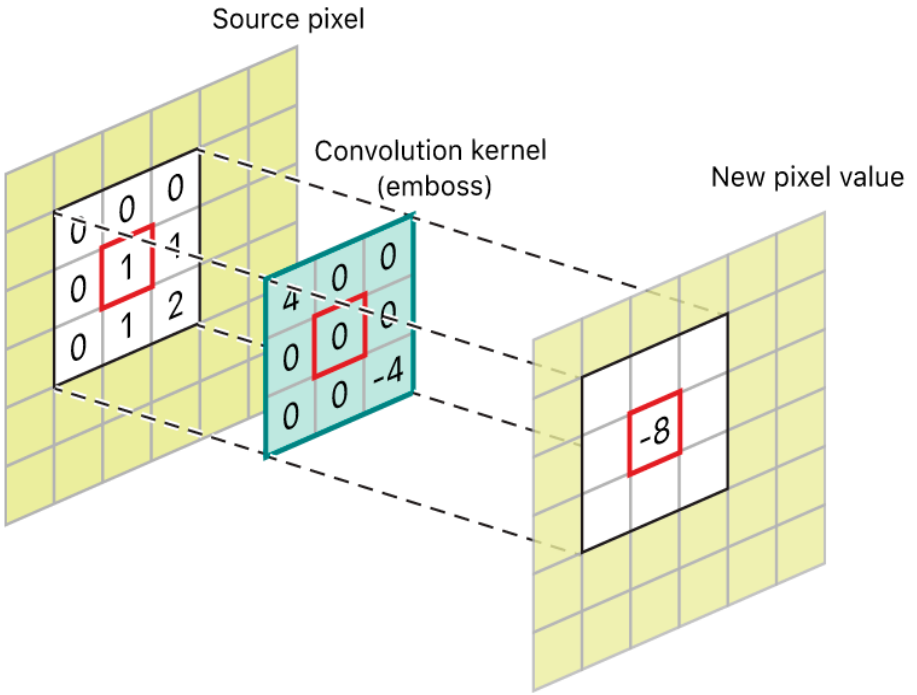
LSTM for spatio-temporal prediction

- All input, output and intermediate representations in RNNs and LSTMs: vectors
- Obtained by transformations by matrices
- Spatio-temporal sequence: each input is a matrix or tensor
- LSTM “vectorizes” the inputs, loses out spatial information!
- Solution: make the output and intermediate representations as tensors of input’s dimensions
- All the matrix-based transformations to be replaced by “Convolutions”
- LSTM -> Conv-LSTM!

3.1 Convolutional LSTM

The major drawback of FC-LSTM in handling spatiotemporal data is its usage of full connections in input-to-state and state-to-state transitions in which no spatial information is encoded. To overcome this problem, a distinguishing feature of our design is that all the inputs $\mathcal{X}_1, \ldots, \mathcal{X}_t$, cell outputs $\mathcal{C}_1, \ldots, \mathcal{C}_t$, hidden states $\mathcal{H}_1, \ldots, \mathcal{H}_t$, and gates i_t, f_t, o_t of the ConvLSTM are 3D tensors whose last two dimensions are spatial dimensions (rows and columns). To get a better picture of the inputs and states, we may imagine them as vectors standing on a spatial grid. The ConvLSTM determines the future state of a certain cell in the grid by the inputs and past states of its local neighbors. This can easily be achieved by using a convolution operator in the state-to-state and input-to-state transitions (see Fig. 2). The key equations of ConvLSTM are shown in (3) below, where '$*$' denotes the convolution operator and '$\circ$', as before, denotes the Hadamard product:

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o) \\
\mathcal{H}_t &= o_t \circ \tanh(\mathcal{C}_t)
\end{aligned}
\tag{3}
$$

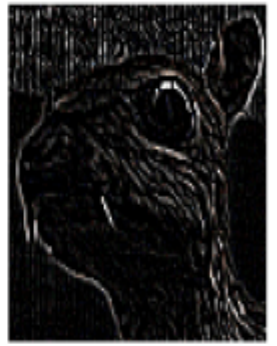


Input image

Convolution Kernel

$$
\begin{bmatrix}
-1 & -1 & -1 \\
-1 & 8 & -1 \\
-1 & -1 & -1
\end{bmatrix}
$$

Feature map



Sequence Prediction with ConvLSTM

- (Hidden State, Cell State): tensors that are updated at each input step
- Two or more LSTMs may be stacked together to encode long sequences
- Multiple LSTMS layers can “remember” more values
- (H_T, C_T) from each LSTM layer: intermediate representation! (T is the length of input sequence)
- The intermediate representation is copied into the decoder for output generation
- Outputs generated one step-by- one step
- Output of one step becomes input to next step

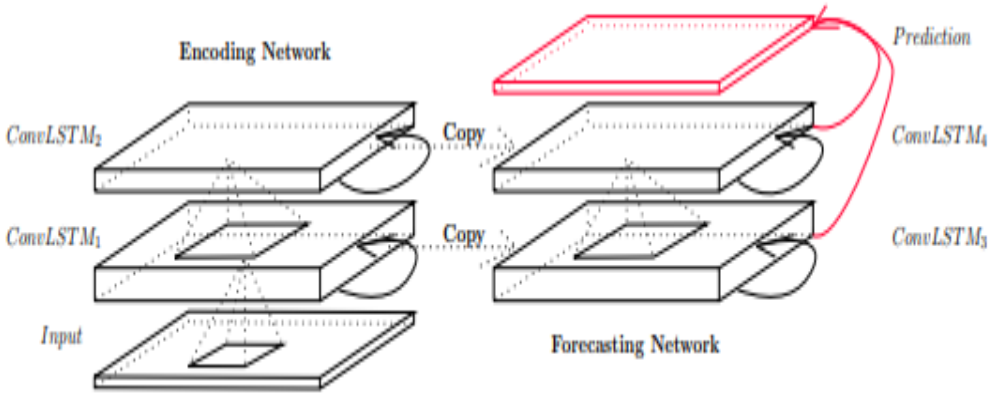


Figure 3: Encoding-forecasting ConvLSTM network for precipitation nowcasting

state to give the final prediction:

$$\begin{aligned}\tilde{\mathcal{X}}_{t+1},\ldots,\tilde{\mathcal{X}}_{t+K} &= \underset{\mathcal{X}_{t+1},\ldots,\mathcal{X}_{t+K}}{\arg\max}\ p(\mathcal{X}_{t+1},\ldots,\mathcal{X}_{t+K}\mid \hat{\mathcal{X}}_{t-J+1},\hat{\mathcal{X}}_{t-J+2},\ldots,\hat{\mathcal{X}}_t)\\ &\approx \underset{\mathcal{X}_{t+1},\ldots,\mathcal{X}_{t+K}}{\arg\max}\ p(\mathcal{X}_{t+1},\ldots,\mathcal{X}_{t+K}\mid f_{encoding}(\hat{\mathcal{X}}_{t-J+1},\hat{\mathcal{X}}_{t-J+2},\ldots,\hat{\mathcal{X}}_t)) \quad (4)\\ &\approx g_{forecasting}(f_{encoding}(\hat{\mathcal{X}}_{t-J+1},\hat{\mathcal{X}}_{t-J+2},\ldots,\hat{\mathcal{X}}_t))\end{aligned}$$

Table 2: Comparison of the average scores of different models over 15 prediction steps.

Model	Rainfall-MSE	CSI	FAR	POD	Correlation
ConvLSTM(3x3)-3x3-64-3x3-64	**1.420**	**0.577**	**0.195**	**0.660**	**0.908**
Rover1	1.712	0.516	0.308	0.636	0.843
Rover2	1.684	0.522	0.301	0.642	0.850
Rover3	1.685	0.522	0.301	0.642	0.849
FC-LSTM-2000-2000	1.865	0.286	0.335	0.351	0.774

4.2 Radar Echo Dataset

The radar echo dataset used in this paper is a subset of the three-year weather radar intensities collected in Hong Kong from 2011 to 2013. Since not every day is rainy and our nowcasting target is precipitation, we select the top 97 rainy days to form our dataset. For preprocessing, we first transform the intensity values Z to gray-level pixels P by setting $P = \frac{Z-\min\{Z\}}{\max\{Z\}-\min\{Z\}}$ and crop the radar maps in the central 330×330 region. After that, we apply the disk filter[5] with radius 10 and resize the radar maps to 100×100. To reduce the noise caused by measuring instruments, we further remove the pixel values of some noisy regions which are determined by applying K-means clustering to the monthly pixel average. The weather radar data is recorded every 6 minutes, so there are 240 frames per day. To get disjoint subsets for training, testing and validation, we partition each daily sequence into 40 non-overlapping frame blocks and randomly assign 4 blocks for training, 1 block for testing and 1 block for validation. The data instances are sliced from these blocks using a 20-frame-wide sliding window. Thus our radar echo dataset contains 8148 training sequences, 2037 testing sequences and 2037 validation sequences and all the sequences are 20 frames long (5 for the input and 15 for the prediction). Although the training and testing instances sliced from the same day may have some dependencies, this splitting strategy is still reasonable because in real-life nowcasting, we do have access to all previous data, including data from the same day, which allows us to apply online fine-tuning of the model. Such data splitting may be viewed as an approximation of the real-life "fine-tuning-enabled" setting for this application.

We set the patch size to 2 and train a 2-layer ConvLSTM network with each layer containing 64 hidden states and 3×3 kernels. For the ROVER algorithm, we tune the parameters of the optical

Missing Value Interpolation

Machine Learning for Earth System Sciences

2nd February 2021

Anomaly

- X(s,t) : measurement of quantity ‘X’ at location ‘s’ and time ‘t’
- μ(s): local mean, or “climatology”,

  eg. Mean daily temperature in Kharagpur
- μ(t): temporal mean, or “climatology”,

  eg. Mean aggregate rainfall in August
- μ(s,t): local mean with respect to time

  eg. Mean daily temperature for February in Kharagpur
- Anomaly = observation - climatology

Anomaly

- Anomaly $Y(s,t) = X(s,t) - \mu$ (positive, negative, zero)
- μ can be $\mu(s)$, $\mu(t)$ or $\mu(s,t)$ depending on context
- Anomaly Y likely to be spatially spatially coherent
- Many neighboring locations simultaneously have anomaly with large magnitude: anomaly event
- Anomaly event can be positive or negative!
- Eg. Heat wave: positive anomaly event w.r.t. temperature

  Drought: negaitive anomaly event w.r.t. rainfall

Any random data matrix

	T1	T2	T3	T4	T5
S1	32	30	19	23	32
S2	33	16	28	35	26
S3	25	33	13	15	33
S4	27	14	34	29	15
S5	38	26	32	35	18
S6	15	18	24	26	28

Can you predict missing values?

	T1	T2	T3	T4	T5
S1	32	30	19	x	32
S2	33	x	28	35	26
S3	25	33	x	15	33
S4	27	x	34	x	15
S5	x	26	32	35	18
S6	15	18	24	26	x

Can you predict missing values?

	T1	T2	T3	T4	T5
S1	32	30	32	x	32
S2	33	x	34	35	33
S3	25	23	x	26	26
S4	27	x	28	x	28
S5	x	18	19	23	18
S6	15	14	13	15	x

Geophysical Data Matrix

	T1	T2	T3	T4	T5
S1	32	30	32	35	32
S2	33	32	34	35	33
S3	25	23	24	26	26
S4	27	26	28	29	28
S5	18	18	19	23	18
S6	15	14	13	15	15

Location-wise interpolation

	T1	T2	T3	T4	T5	Local Mean
S1	32	30	32	x	32	31
S2	33	x	34	35	33	34
S3	25	23	x	26	26	25
S4	27	x	28	x	28	28
S5	x	18	19	23	18	19
S6	15	14	13	15	x	14

Time-wise Interpolation (makes no sense here)

	T1	T2	T3	T4	T5
S1	32	30	32	x	32
S2	33	x	34	35	33
S3	25	23	x	26	26
S4	27	x	28	x	28
S5	x	18	19	23	18
S6	15	14	13	15	x
Daily Mean	26.4	21.25	25.2	24.75	27.4

Spatio-temporal Interpolation

- $X(s,t) = \mu(s) + \eta(s,t) + e(s,t)$
- $\mu(s)$ = local mean/climatology
- $\eta(s,t) + e(s,t)$ = anomaly
- $\eta(s,t)$: predictable part of anomaly, $e(s,t)$: unpredictable
- Missing value prediction = climatology + anomaly prediction
- Climatology: estimate from past data or recent values!

Location-wise interpolation

	T1	T2	T3	T4	T5	Estimated Climatology
S1	32	30	32	x	32	31.5
S2	33	x	34	35	33	33.75
S3	25	23	x	26	26	25
S4	27	x	28	x	28	27.67
S5	x	18	19	23	18	19.5
S6	15	14	13	15	x	14.25

Anomaly Prediction

$$X_{s,t_1} = a \cdot X_{s,t_2} + b\, X_{s,t_3} \quad \forall$$

$$X_{s_1,t} = a \cdot X_{s_2,t} + b\, X_{s_3,t}$$

- Calculate anomaly at neighboring locations!
- Estimate of η(s,t) = Mean of neighboring anomalies!

- Basically a simplified model of autoregression!
- Only “neighboring” locations used as predictors with equal coefficients
- Climatology: bias term of regression!

Anomaly Calculation

	T1	T2	T3	T4	T5	Estimated Climatology
S1	0.5	-1.5	0.5	x	0.5	31.5
S2	-0.75	x	0.25	1.25	-0.75	33.75
S3	0	-2	x	1	1	25
S4	-0.67	x	0.33	x	0.33	27.67
S5	x	-1.5	-0.5	3.5	-1.5	19.5
S6	0.75	-0.25	-1.25	0.75	x	14.25

$X(s,t) - \mu(s) \neq \eta(s,t) + \cancel{\epsilon(s,t)}$

ANOMALY

GLOBAL

SPATIO-TEMP CORR.

Anomaly Calculation

	T1	T2	T3	T4	T5	Estimated Climatology
S1	0.5	-1.5	0.5	x	0.5	31.5
S2	-0.75	x	0.25	1.25	-0.75	33.75
S3	0	-2	x	1	1	25
S4	-0.67	x	0.33	x	0.33	27.67
S5	x	-1.5	-0.5	3.5	-1.5	19.5
S6	0.75	-0.25	-1.25	0.75	x	14.25

$|x - 1.25| \approx 0.9$ $|x - 0.5| \approx 2.5$ $|x - 0.5| \approx 2$

Anomaly Interpolation from Neighbors

	T1	T2	T3	T4	T5	Estimated Climatology
S1	0.5	-1.5	0.5	1.25	0.5	31.5
S2	-0.75	-1.5	0.25	1.25	-0.75	33.75
S3	0	-2	0.33	1	1	25
S4	-0.67	-2	0.33	1	0.33	27.67
S5	0.75	-1.5	-0.5	3.5	-1.5	19.5
S6	0.75	-0.25	-1.25	0.75	-1.5	14.25

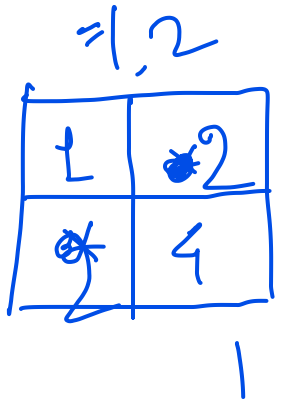
Missing value prediction

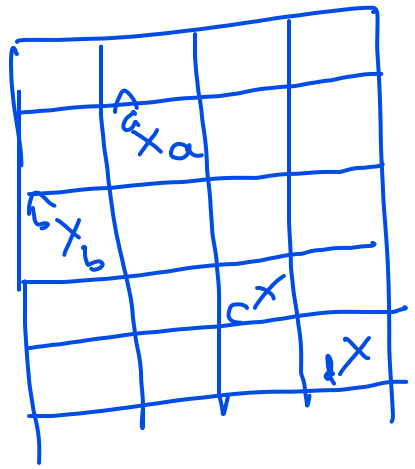
	T1	T2	T3	T4	T5	Estimated Climatology
S1	32	30	32	33	0.5	31.5
S2	33	-1.5	0.25	1.25	-0.75	33.75
S3	0	-2	0.33	1	1	25
S4	-0.67	-2	0.33	1	0.33	27.67
S5	0.75	-1.5	-0.5	3.5	-1.5	19.5
S6	0.75	-0.25	-1.25	0.75	-1.5	14.25

Missing value prediction

	T1	T2	T3	T4	T5	Estimated Climatology
S1	32	30	32	33	0.5	31.5
S2	33	-1.5	0.25	1.25	-0.75	33.75
S3	0	-2	0.33	1	1	25
S4	-0.67	-2	0.33	1	0.33	27.67
S5	0.75	-1.5	-0.5	3.5	-1.5	19.5
S6	0.75	-0.25	-1.25	0.75	-1.5	14.25

Alternative Approaches





- Low-rank matrix completion!
- Theoretical Result: If a limited fraction of entries in a matrix are missing, they can be estimated, provided the matrix rank is low!
- Rank of matrix = number of linearly independent rows/columns
- Spatio-temporal data matrices: “approximately” low-rank!

Not a low-rank matrix

	T1	T2	T3	T4	T5
S1	32	30	19	23	32
S2	33	16	28	35	26
S3	25	33	13	15	33
S4	27	14	34	29	15
S5	38	26	32	35	18
S6	15	18	24	26	28

“Approximately” low-rank matrix

	T1	T2	T3	T4	T5
S1	32	30	32	35	32
S2	33	32	34	35	33
S3	25	23	24	26	26
S4	27	26	28	29	28
S5	18	18	19	23	18
S6	15	14	13	15	15

Low-rank matrix Completion

- Frame it as an optimization problem
- Partially observed matrix M
- Set of observed entries: Ω

$$\textit{minimize} \quad \textit{rank}(X), \textit{ s.t. } X(\Omega) = M(\Omega)$$

- rank(X) is a non-convex function, difficult to optimize
- Relaxation: Nuclear Norm (sum of singular values)!
- Can be solved by specialized optimization algorithms (Candes and Recht, 2008)

Another Approach: Matrix Factorization

- Low rank matrix X = A.B
- A: (S x K) matrix, B: (K x T) matrix, K much lower than S,T
- Can we find suitable A and B matrices?
- Number of unknown values: SK + KT
- Each entry in X: X(s,t) = A(s,1)B(1,t) + A(s,2)B(2,t) + …… + A(s,k)B(k,t)
- Number of observed values in X >> SK + KT!
- Number of equations >> Number of variables (overdetermined system)
- (A*,B*) = least square solutions of A,B -> estimate X* = A*.B*!

Statistical and dynamical models of Spatio-temporal Processes

Adway Mitra

25 January 2021

Geo-statistical Equation

- $X(s,t) = \mu(s,t) + \eta(s,t) + \epsilon(s,t)$
- $\mu(s,t)$: local mean, spatially or temporally stationery
- $\eta(s,t)$: dynamic process model containing spatial or temporal correlations
- $\epsilon(s,t)$: random noise

Hierarchical model for Spatio-temporal Process

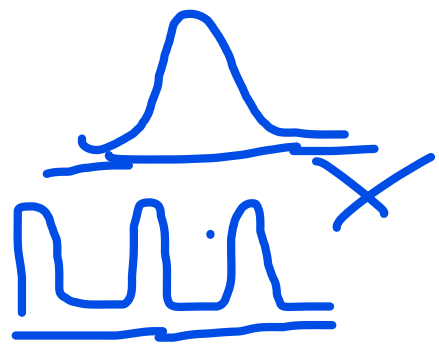
- Hierarchical model: Data model + Process model + Parameter model
- *Parameter model*: parameter values sampled from a prior distribution
- *Process model*: describes the process (including spatio-temporal dynamics) based on the parameters
- *Data model*: describes the observations, in terms of the process
- $Z(s,t)$ is the description of the process (latent variable)
- $X(s,t)$ are the observations

Template of a hierarchical model

$X(s,t) = g(Z(s,t), Y(s,t))$

- $\theta \sim p(\eta)$ [Parameter Model]
- $Z(s,t) \sim f(\theta)$ [Process Model]
- $X(s,t) \sim g(Z, Y)$ [Data/observation Model]; Y: co-variates
- Simulation/forward problem: Generate X by sampling in order
- Inverse problem: estimate Z, θ from X, Y using Bayes Theorem
- How to choose p, f, g?

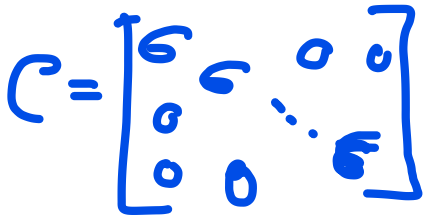
Designer's choice

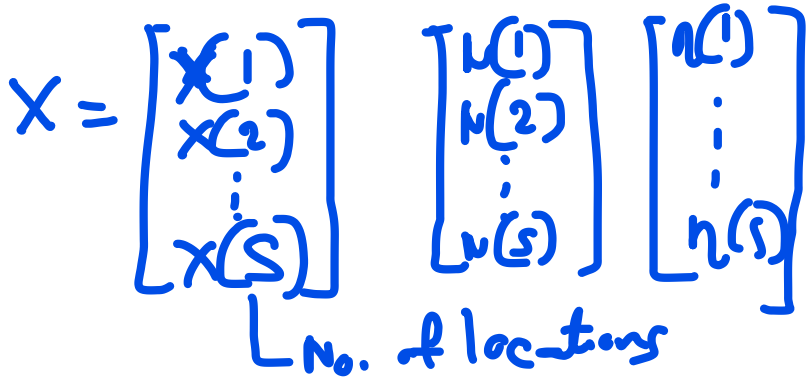


Spatial Process

- Data Model: $X(s) \sim \mathcal{N}(\mu(s) + \eta(s), \sigma)$
- Observations at each time-point is a realization from this model
- $\eta(s) = AZ(s) + BY(s)$
- ϵ is managed by σ
- $\mu(s), Z(s)$: contains covariance between different locations
- Y are co-variates which are extraneous to the model
- A, B represent transformation coefficients

Spatial Process- vectorized





- Data Model: $X \sim \mathcal{N}(\mu + \eta, \sigma I)$
- $\eta = AZ + BY$
- X, μ, η, Z, Y are vectors of length S
- A, B are transformation matrices
- Vectorization allows the influence of other locations
- If A, B diagonal matrices: back to the previous model

Interpretation

- μ: local effect (eg. all locations have local mean temperature)
- Z: global effect (eg. during a heat wave, temperatures at all locations are affected by different degrees)
- A: transfer the effect of heat wave on the local observations
- Y: covariates (eg. humidity, rainfall etc)
- B: effect of co-variates (eg. role of humidity on temperature)

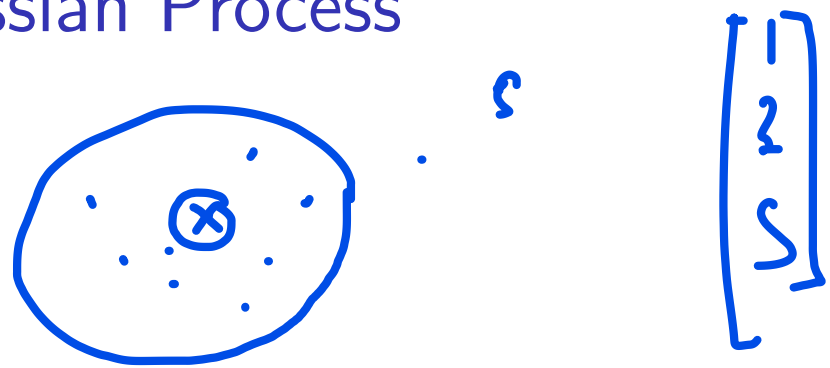
Temporal Process

- Data Model: $X(t) \sim \mathcal{N}(\mu(t) + \eta(t), \sigma)$
- $\eta(t) = AZ(t) + BY(t)$
- $\mu(t), Z(t)$: contains covariance between different time-points
- Y are co-variates which are extraneous to the model
- A, B represent transformation matrices

Interpretation

- μ: seasonal component (eg. all months have seasonal mean temperature)
- Z: trend component (eg. a heat wave that lasts for a few days, global warming)

Gaussian Process



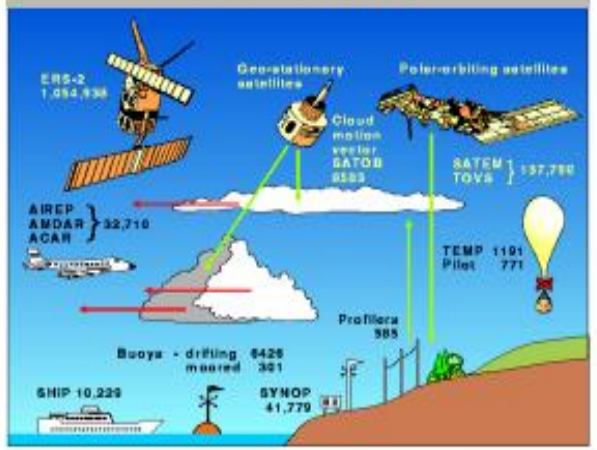
- Consider a (finite or infinite) set of random variables X_1, X_2, ...
- Consider any random finite subset $\{X_{i1}, X_{i2}, \ldots, X_{iN}\}$
- Then we have $(X_{i1}, X_{i2}, \ldots, X_{iN}) \sim \mathcal{N}(\mu, \Sigma)$
- $\mu(s)$: mean function (a function of s)
- $\Sigma(s, s')$: covariance function (a function of $||s - s'||$)

Data Assimilation with Kalman Filters

Machine Learning for Earth System Sciences

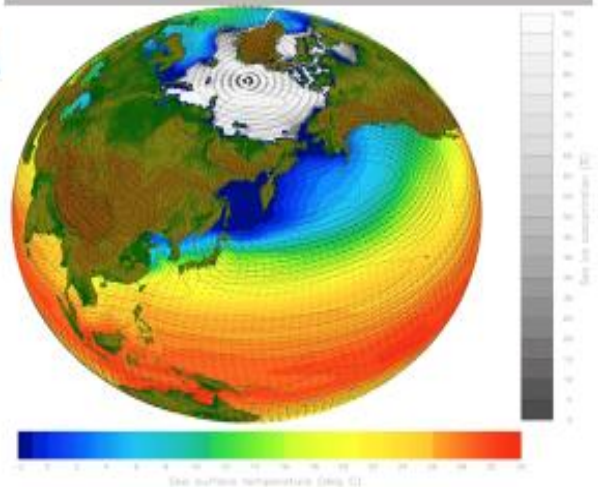
Carrassi et al

Observations



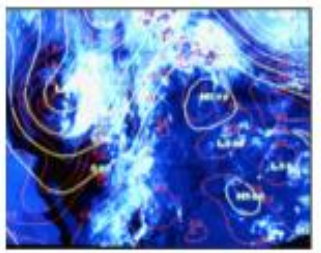
Data Assimilation best combines model and observations and brings synergy

Model Simulations

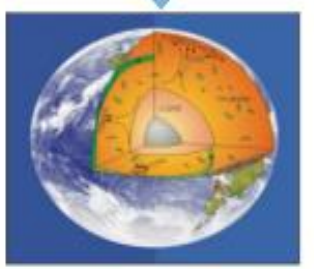


An on-going rapid expansion from **Weather Science (NWP)** into **Climate Science/Geophysics** in general:

- Oceanography
- Climate Prediction
- Climate Assessment
- Hydrology
- Geology
- Climatology
- Detection & Attribution
- ... and many more beyond geosciences ...

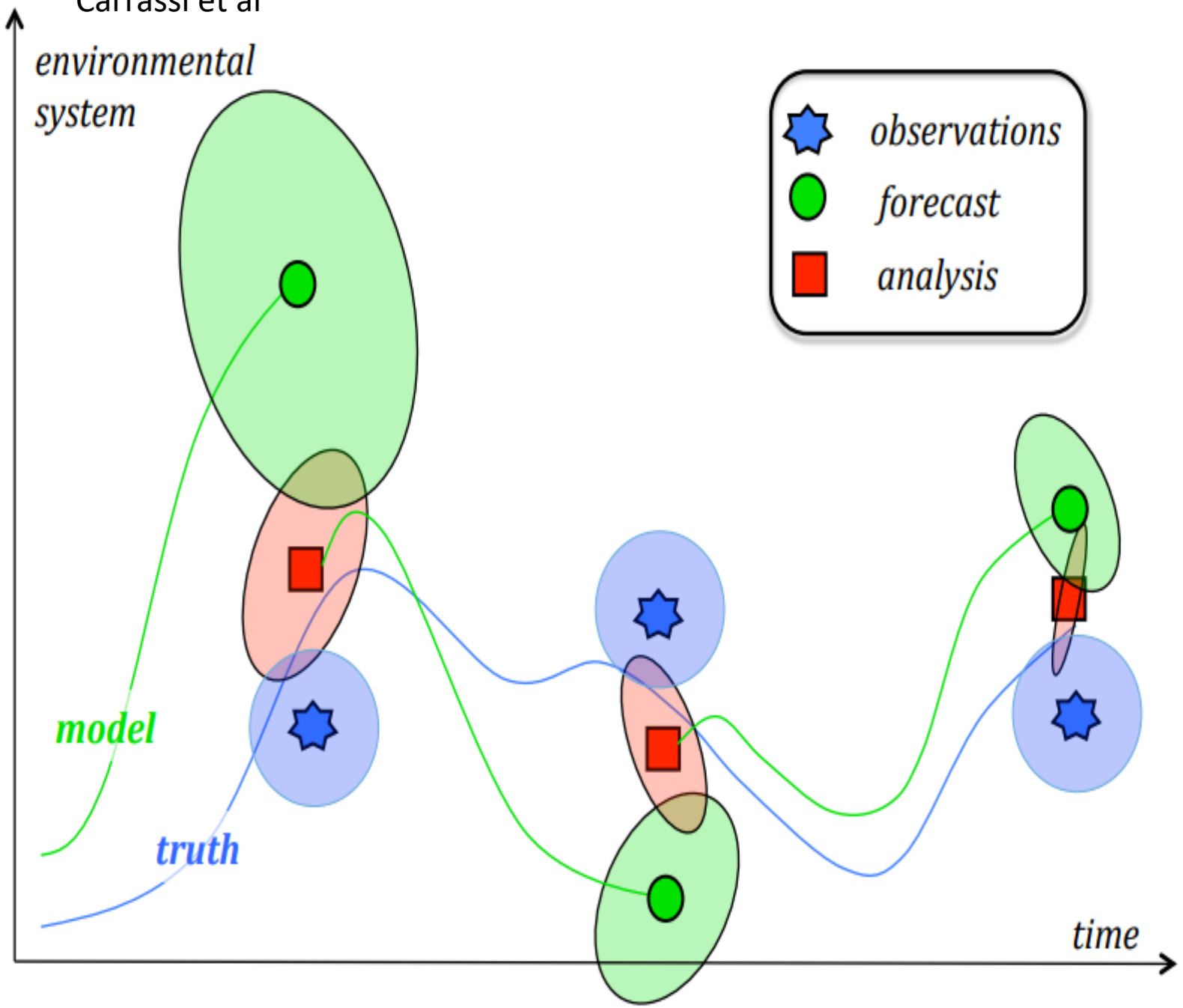


NWP



Geophysics

Carrassi et al



- The problem is in principle solved using a **probabilistic framework**
- The quantity of interest are the **probability density functions**, PDF
- The **PDFs are evolved in time** and updated at analysis times using **Bayes's rules**

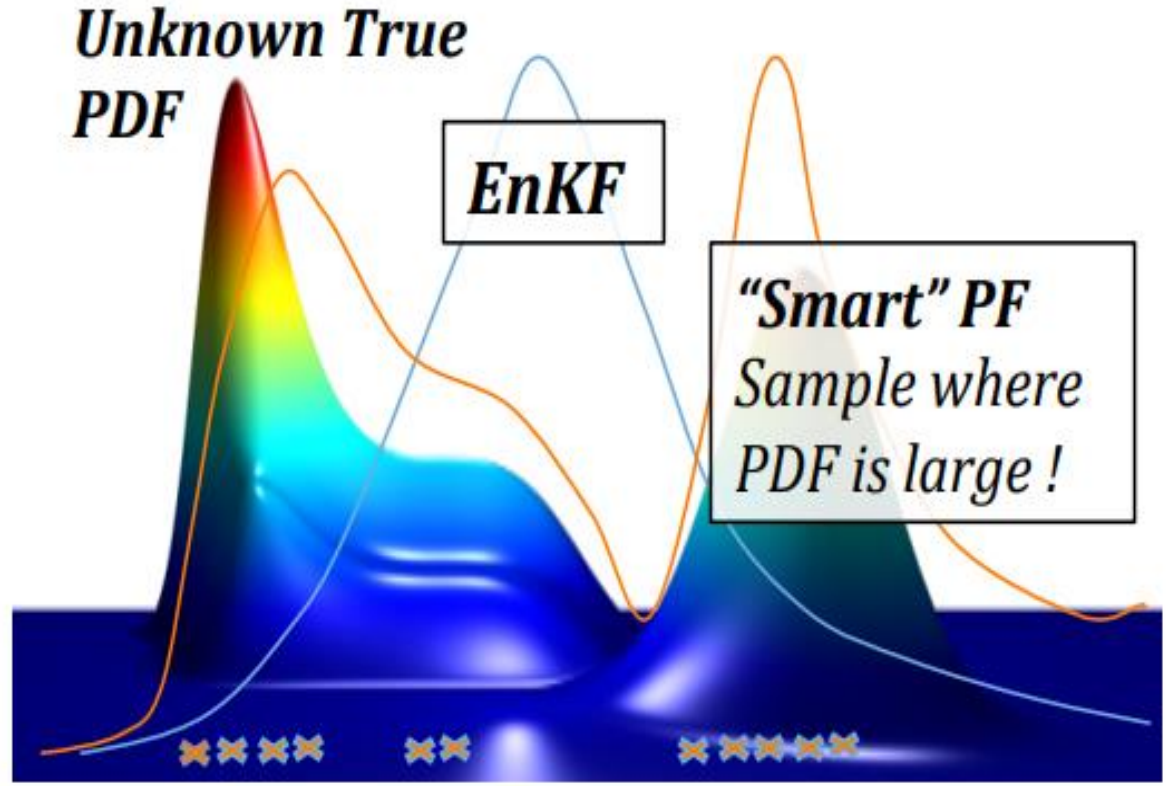
Carrassi et al

Efficient Bayesian Data Assimilation

- EnKF/4DVar fails in highly nonlinear/non-Gaussian situations
- Nonlinear Bayesian ***Particle Filter*** are required.

Key Scientific Issue

- **Curse of Dimensionality**
- **Big Data Problem** (model/obs $10^9/10^7$)
- Not **computational power alone**



- NERSC DA group is actively studying advanced formulations of Particle Filter and EnKF to deal with nonlinearity
- The main idea is to study PF which incorporates model dynamic's features in its design (see Raanes and Grudzien poster)

State Update Model: Kalman Filter

- State representation: X(1), X(2), ……., X(t), …. [latent variables]
- Observations: Z(1), Z(2), ……, Z(t), …….
- Input: U(1), U(2), ……. , U(t), …….

- Observation model: Z(t) = f(X(t), u(t)) [f: linear/non-linear]
- State updation model: X(t+1) = g(X(t), u(t))

- **Filtering problem**: Estimate X(t) based on Z(1), Z(2), ….. , Z(t)
- **Smoothing problem**: Estimate X(t) based on Z(1), Z(2), …., Z(t), Z(t+1), …. , Z(T)
- **Prediction problem**: Estimate X(t) based on Z(1), Z(2), ….., Z(t-1)

Kalmanfilter.net

$$\hat{\boldsymbol{x}}_{n+1,n} = \boldsymbol{F}\hat{\boldsymbol{x}}_{n,n} + \boldsymbol{G}\boldsymbol{u}_n + \boldsymbol{w}_n$$

Where:

$\hat{\boldsymbol{x}}_{n+1,n}$	is a predicted system state vector at time step $n+1$
$\hat{\boldsymbol{x}}_{n,n}$	is an estimated system state vector at time step n
$\boldsymbol{u}_n$	is a **control variable** or **input variable** - a measurable (deterministic) input to the system
$\boldsymbol{w}_n$	is a **process noise** or disturbance - an unmeasurable input that affects the state
$\boldsymbol{F}$	is a **state transition matrix**
$\boldsymbol{G}$	is a **control matrix** or **input transition matrix** (mapping control to state variables)

$$\boldsymbol{z}_n = \boldsymbol{H}\boldsymbol{x}_n + \boldsymbol{v}_n$$

Where:

$\boldsymbol{z}_n$	is a measurement vector
$\boldsymbol{x}_n$	is a true system state (hidden state)
$\boldsymbol{v}_n$	is a random noise vector
$\boldsymbol{H}$	is an **observation matrix**

$$\hat{\boldsymbol{x}}_{n,n} = \hat{\boldsymbol{x}}_{n,n-1} + \boldsymbol{K}_n(\boldsymbol{z}_n - \boldsymbol{H}\hat{\boldsymbol{x}}_{n,n-1})$$

where:

$\hat{\boldsymbol{x}}_{n,n}$	is a estimated system state vector at time step n
$\hat{\boldsymbol{x}}_{n,n-1}$	is a predicted system state vector at time step $n-1$
$\boldsymbol{K}_n$	is a Kalman Gain
$\boldsymbol{z}_n$	is a measurement
$\boldsymbol{H}$	is an observation matrix

Consider a free-falling object. The state vector includes the altitude h and the object's velocity $\dot{h}$:

$$\hat{\boldsymbol{x}}_{\boldsymbol{n}} = \begin{bmatrix} \hat{h}_n \\ \hat{\dot{h}}_n \end{bmatrix}$$

The state transition matrix $\boldsymbol{F}$ is:

$$\boldsymbol{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

The control matrix $\boldsymbol{G}$ is:

$$\boldsymbol{G} = \begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \end{bmatrix}$$

The input variable $\boldsymbol{u_n}$ is:

$$\boldsymbol{u_n} = [g]$$

where g is the gravitational acceleration.

We don't have a sensor that measures acceleration, but we know that for a falling object, the acceleration equals g.

The state extrapolation equation looks like:

$$\begin{bmatrix} \hat{h}_{n+1,n} \\ \hat{\dot{h}}_{n+1,n} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{h}_{n,n} \\ \hat{\dot{h}}_{n,n} \end{bmatrix} + \begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \end{bmatrix} [g]$$

Kalmanfilter.net

$$\boldsymbol{P_{n+1,n} = FP_{n,n}F^T + Q}$$

Where:

$\boldsymbol{P_{n,n}}$	is the uncertainty of an estimate - covariance matrix of the current state
$\boldsymbol{P_{n+1,n}}$	is the uncertainty of a prediction - covariance matrix for the next state
$\boldsymbol{F}$	is the state transition matrix that we derived in the "Modeling linear dynamic systems" section
$\boldsymbol{Q}$	is the process noise matrix

$$\boldsymbol{P_{n,n} = (I - K_n H)\, P_{n,n-1} (I - K_n H)^T + K_n R_n K_n^T}$$

where:

$\boldsymbol{P_{n,n}}$	is the estimate uncertainty (covariance) matrix of the current state
$\boldsymbol{P_{n,n-1}}$	is the prior estimate uncertainty (covariance) matrix of the current state (predicted at the previous state)
$\boldsymbol{K_n}$	is the Kalman Gain
$\boldsymbol{H}$	is the observation matrix
$\boldsymbol{R_n}$	is the Measurement Uncertainty (measurement noise covariance matrix)

$$\boldsymbol{K_n = P_{n,n-1} H^T \left(H P_{n,n-1} H^T + R_n\right)^{-1}}$$

where:

$\boldsymbol{K_n}$	is the Kalman Gain
$\boldsymbol{P_{n,n-1}}$	is the prior estimate uncertainty (covariance) matrix of the current state (predicted at the previous step)
$\boldsymbol{H}$	is the observation matrix
$\boldsymbol{R_n}$	is the Measurement Uncertainty (measurement noise covariance matrix)

Kalmanfilter.net

Time Update ("Predict")

1. Extrapolate the state

$$\hat{\boldsymbol{x}}_{n+1,n} = \boldsymbol{F}\hat{\boldsymbol{x}}_{n,n} + \boldsymbol{G}\boldsymbol{u}_n$$

2. Extrapolate uncertainty

$$\boldsymbol{P}_{n+1,n} = \boldsymbol{F}\boldsymbol{P}_{n,n}\boldsymbol{F}^T + \boldsymbol{Q}$$

Measurement Update ("Correct")

1. Compute the Kalman Gain

$$\boldsymbol{K}_n = \boldsymbol{P}_{n,n-1}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{P}_{n,n-1}\boldsymbol{H}^T + \boldsymbol{R}_n\right)^{-1}$$

2. Update estimate with measurement

$$\hat{\boldsymbol{x}}_{n,n} = \hat{\boldsymbol{x}}_{n,n-1} + \boldsymbol{K}_n\left(\boldsymbol{z}_n - \boldsymbol{H}\hat{\boldsymbol{x}}_{n,n-1}\right)$$

3. Update the estimate uncertainty

$$\boldsymbol{P}_{n,n} = (\boldsymbol{I} - \boldsymbol{K}_n\boldsymbol{H})\boldsymbol{P}_{n,n-1}(\boldsymbol{I} - \boldsymbol{K}_n\boldsymbol{H})^T + \boldsymbol{K}_n\boldsymbol{R}_n\boldsymbol{K}_n^T$$

Initial Estimate: $\hat{\boldsymbol{x}}_{0,0}$, $\boldsymbol{P}_{0,0}$

	Equation	Equation Name	Alternative names
Predict	$\hat{\boldsymbol{x}}_{\boldsymbol{n+1,n}} = \boldsymbol{F}\hat{\boldsymbol{x}}_{\boldsymbol{n,n}} + \boldsymbol{G}\boldsymbol{u}_{\boldsymbol{n}}$	State Extrapolation	Predictor Equation Transition Equation Prediction Equation Dynamic Model State Space Model
	$\boldsymbol{P}_{\boldsymbol{n+1,n}} = \boldsymbol{F}\boldsymbol{P}_{\boldsymbol{n,n}}\boldsymbol{F}^{\boldsymbol{T}} + \boldsymbol{Q}$	Covariance Extrapolation	Predictor Covariance Equation
Update (correction)	$\hat{\boldsymbol{x}}_{\boldsymbol{n,n}} = \hat{\boldsymbol{x}}_{\boldsymbol{n,n-1}} + \boldsymbol{K}_{\boldsymbol{n}}\left(\boldsymbol{z}_{\boldsymbol{n}} - \boldsymbol{H}\hat{\boldsymbol{x}}_{\boldsymbol{n,n-1}}\right)$	State Update	Filtering Equation
	$\boldsymbol{P}_{\boldsymbol{n,n}} = \left(\boldsymbol{I} - \boldsymbol{K}_{\boldsymbol{n}}\boldsymbol{H}\right)\boldsymbol{P}_{\boldsymbol{n,n-1}}\left(\boldsymbol{I} - \boldsymbol{K}_{\boldsymbol{n}}\boldsymbol{H}\right)^{\boldsymbol{T}} + \boldsymbol{K}_{\boldsymbol{n}}\boldsymbol{R}_{\boldsymbol{n}}\boldsymbol{K}_{\boldsymbol{n}}^{\boldsymbol{T}}$	Covariance Update	Corrector Equation
	$\boldsymbol{K}_{\boldsymbol{n}} = \boldsymbol{P}_{\boldsymbol{n,n-1}}\boldsymbol{H}^{\boldsymbol{T}}\left(\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{n,n-1}}\boldsymbol{H}^{\boldsymbol{T}} + \boldsymbol{R}_{\boldsymbol{n}}\right)^{-\boldsymbol{1}}$	Kalman Gain	Weight Equation
Auxiliary	$\boldsymbol{z}_{\boldsymbol{n}} = \boldsymbol{H}\boldsymbol{x}_{\boldsymbol{n}}$	Measurement Equation	
	$\boldsymbol{R}_{\boldsymbol{n}} = E\left(\boldsymbol{v}_{\boldsymbol{n}}\boldsymbol{v}_{\boldsymbol{n}}^{\boldsymbol{T}}\right)$	Measurement Uncertainty	Measurement Error
	$\boldsymbol{Q}_{\boldsymbol{n}} = E\left(\boldsymbol{w}_{\boldsymbol{n}}\boldsymbol{w}_{\boldsymbol{n}}^{\boldsymbol{T}}\right)$	Process Noise Uncertainty	Process Noise Error
	$\boldsymbol{P}_{\boldsymbol{n,n}} = E\left(\boldsymbol{e}_{\boldsymbol{n}}\boldsymbol{e}_{\boldsymbol{n}}^{\boldsymbol{T}}\right) = E\left(\left(\boldsymbol{x}_{\boldsymbol{n}} - \hat{\boldsymbol{x}}_{\boldsymbol{n,n}}\right)\left(\boldsymbol{x}_{\boldsymbol{n}} - \hat{\boldsymbol{x}}_{\boldsymbol{n,n}}\right)^{\boldsymbol{T}}\right)$	Estimation Uncertainty	Estimation Error

Bayesian Kalman Filtering

- Express the above equations as probabilistic models to account for the uncertainty
- Prediction Model: $P(X(t) \mid X(t-1))$
- Observation Model: $P(Y(t) \mid X(t))$
- Observation sequence: $P(Y(1:T) \mid X(1:T)) = \prod_t P(Y(t) \mid X(t))$
- State sequence: $P(X(1:T)) = P(X(1)) * \prod_t P(X(t) \mid X(t-1))$

- Combining: $P(X(1:T+1) \mid Y(1:T)) = P(X(1)) * \prod_t P(Y(t) \mid X(t)) * \prod_t P(X(t+1) \mid X(t))$

Causality in Earth Science

Adway Mitra

Machine Learning for Earth System Science

AI60002

Causality between two variables

- Consider two variables X and Y (may be spatio-temporal)
- “X causes Y” = Value of X influences value of Y
- Eg. i) Smoking causes cancer
  ii) Clouds cause rainfall
- Spatial causality: X(s) causes Y(s’) where s, s’ may be same
- Temporal causality: X(t) causes Y(t’) where t’>=t
- Controlled process: if we can externally change the value of X, value of Y will change accordingly.

Bi-directional Causality

- Bi-directional causality: “X causes Y” and “Y causes X”!
- Self-replenishing or self-destructive
- i) High temperature (X) causes water evaporation
  ii) Water evaporation creates clouds
  iii) Clouds cause rainfall (Y)
  iv) Rainfall (Y) brings down temperature! (X)

- i) High temperature (X) -> people use air conditioners
  ii) Air conditioners release CO2
  iii) CO2 (Y) causes higher temperature (X)!!!!

Correlation and Causation

X	Y
12	105
25	176
13	109
19	140
23	168
37	225
16	115

Whenever X increases, Y increases too.
Whenever X decreases, Y decreases too.

X	Y
12	153
25	105
13	176
19	109
23	140
37	168
16	225

Whenever X increases, Y increases in next step.
Whenever X decreases, Y decreases in next step

X	Y
12	153
25	105
13	176
19	109
15	125
17	120
16	135

Whenever X in/decreases, Y de/increases.
Whenever Y increases, X increases in next step!

Correlation and Causation

X	Y
12	105
25	176
13	109
19	140
23	168
37	225
16	115

Whenever X increases, Y increases too.
Whenever X decreases, Y decreases too.

High Correlation

X	Y
12	153
25	105
13	176
19	109
23	140
37	168
16	225

Whenever X increases, Y increases in next step.
Whenever X decreases, Y decreases in next step

High lagged Correlation

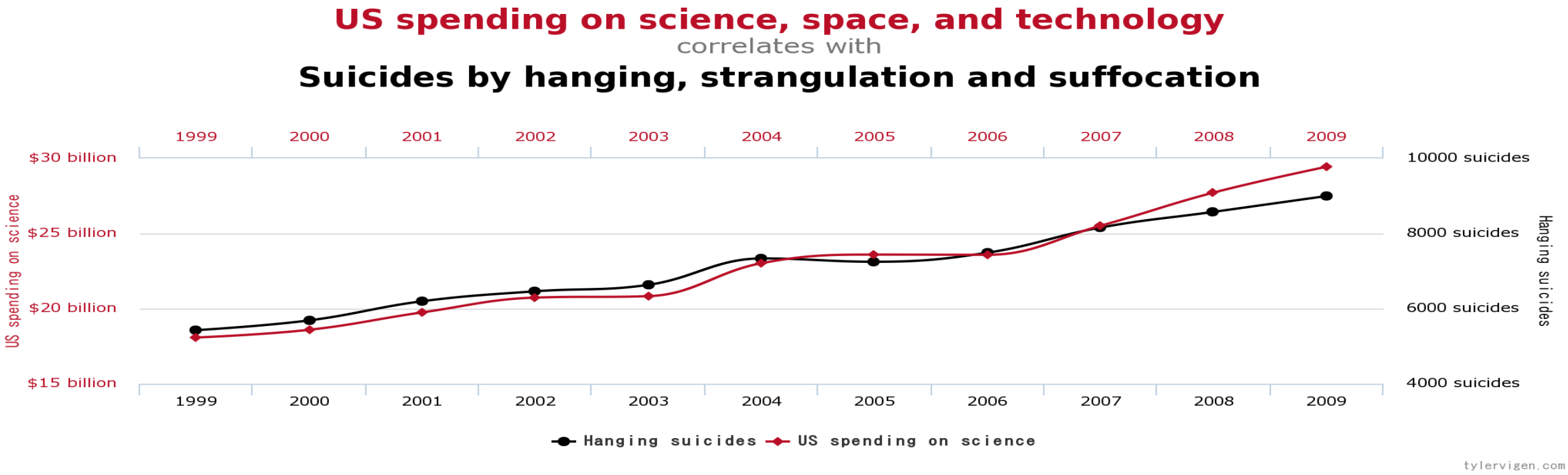
X	Y
12	153
25	105
13	176
19	109
15	125
17	120
16	135

Whenever X in/decreases, Y de/increases.
Whenever Y increases, X increases in next step!

High lagged Correlation, high anti-correlation!

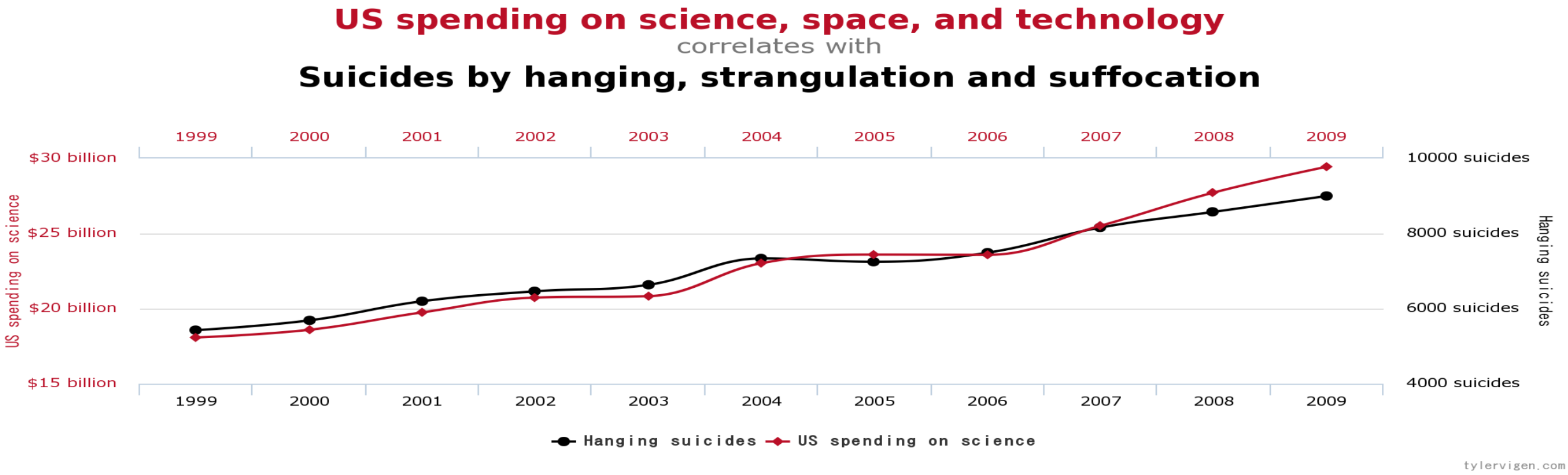
Correlation and Causation

- High correlation need not indicate causal relationship
- “Spurious correlation”: artificially enforced high correlation



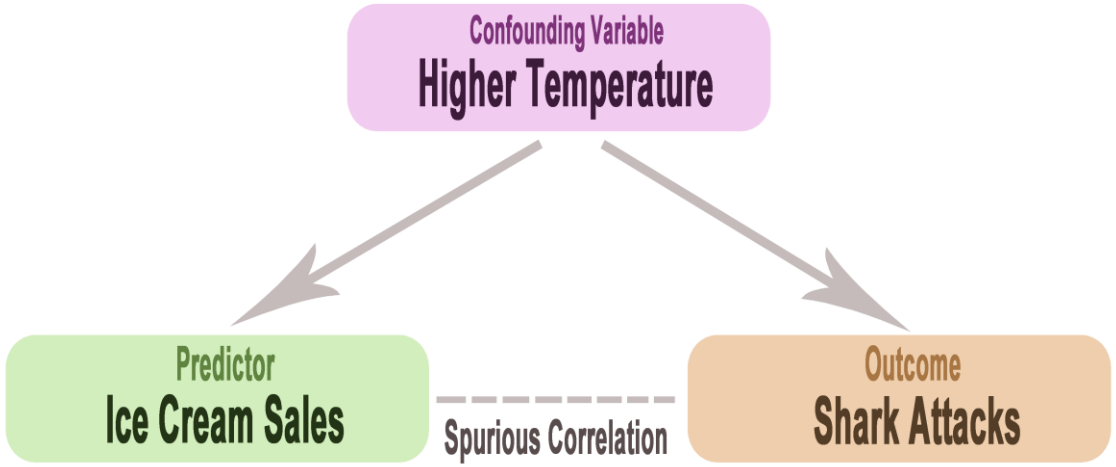
Correlation and Causation

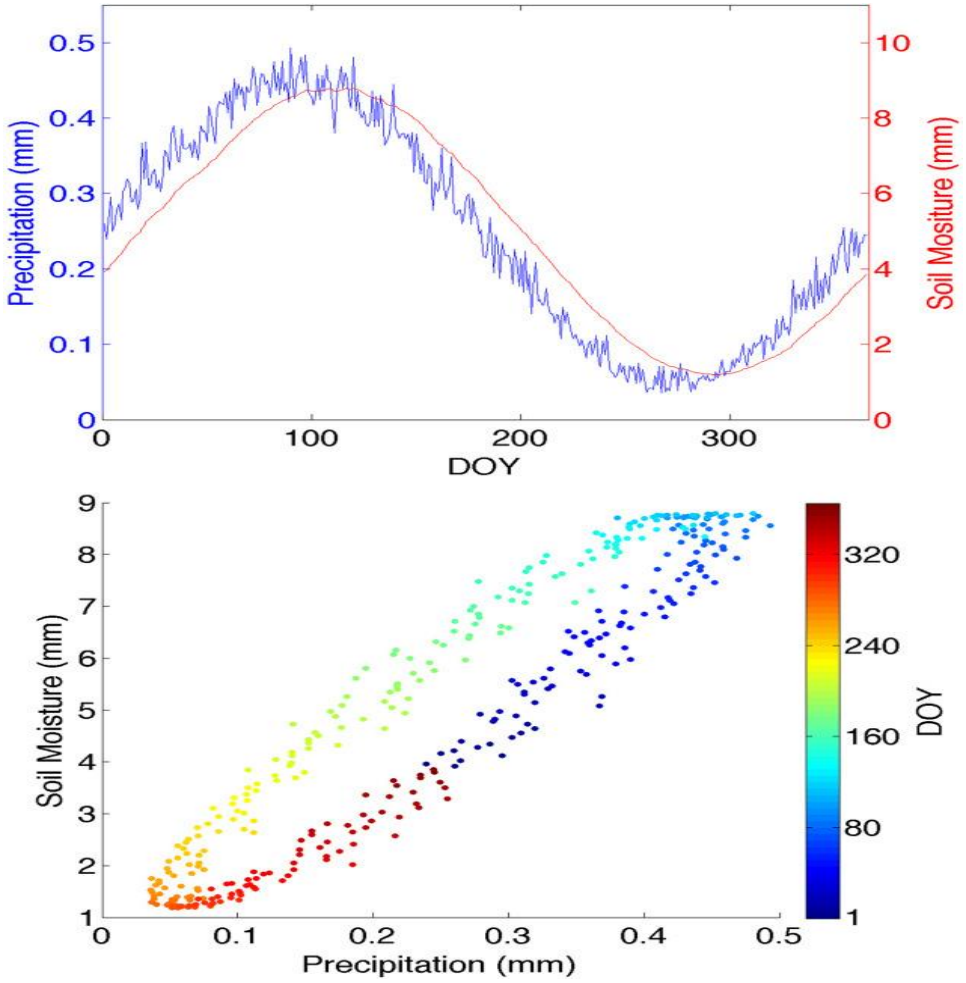
- High correlation need not indicate causal relationship
- “Spurious correlation”: artificially enforced high correlation
- System involves more “confounding” variables which are not observed



Correlation and Causation

- "Explaining away": identify new variable Z which has causal relationship with both X and Y!





Granger Causality in Time-series

- Express X as a linear function of past values of itself
- $X(t) \sim a_1X(t-1) + a_2X(t-2) + a_3X(t-3) + \ldots\ldots$
- Does the estimate improve if we include past values of Y?
- $X(t) \sim a_1X(t-1) + a_2X(t-2) + a_3X(t-3) + \ldots\ldots + b_1Y(t-1) + b_2Y(t-2) + b_3Y(t-3) + \ldots\ldots$
- If yes, then Y Granger-causes X.

Granger Causality in Time-series

- Express X as a linear function of past values of itself
- $X(t) \sim a_1X(t-1) + a_2X(t-2) + a_3X(t-3) + \ldots\ldots$
- Does the estimate improve if we include past values of Y?
- $X(t) \sim a_1X(t-1) + a_2X(t-2) + a_3X(t-3) + \ldots\ldots + b_1Y(t-1) + b_2Y(t-2) + b_3Y(t-3) + \ldots\ldots$
- If yes, then Y Granger-causes X.
- Does X Granger-cause Y?
- $Y(t) \sim c_1Y(t-1) + c_2Y(t-2) + c_3Y(t-3) + \ldots\ldots$
- $Y(t) \sim c_1Y(t-1) + c_2Y(t-2) + c_3Y(t-3) + \ldots\ldots + b_1X(t-1) + b_2X(t-2) + b_3X(t-3) + \ldots\ldots$
- If both yes, then bidirectional causality!

Pearl Causality

- Smoking (X) may cause cancer (Y)
- P(Y=1 | X=1) = 0.4 (relation between cancer and smoking)

- Cancer (Y) can be caused by smoking (X), or many other factors (Z)
- Prob(Y=1) = Prob(X=1,Y=1) + Prob(Y=1,Z=1) + Prob(Y=1,Z=2) + ….

- Prob(Y=1|X=1) helps us to estimate probability of cancer Prob(Y=1)
- But does it tell us anything about probability of smoking?

Pearl Causality

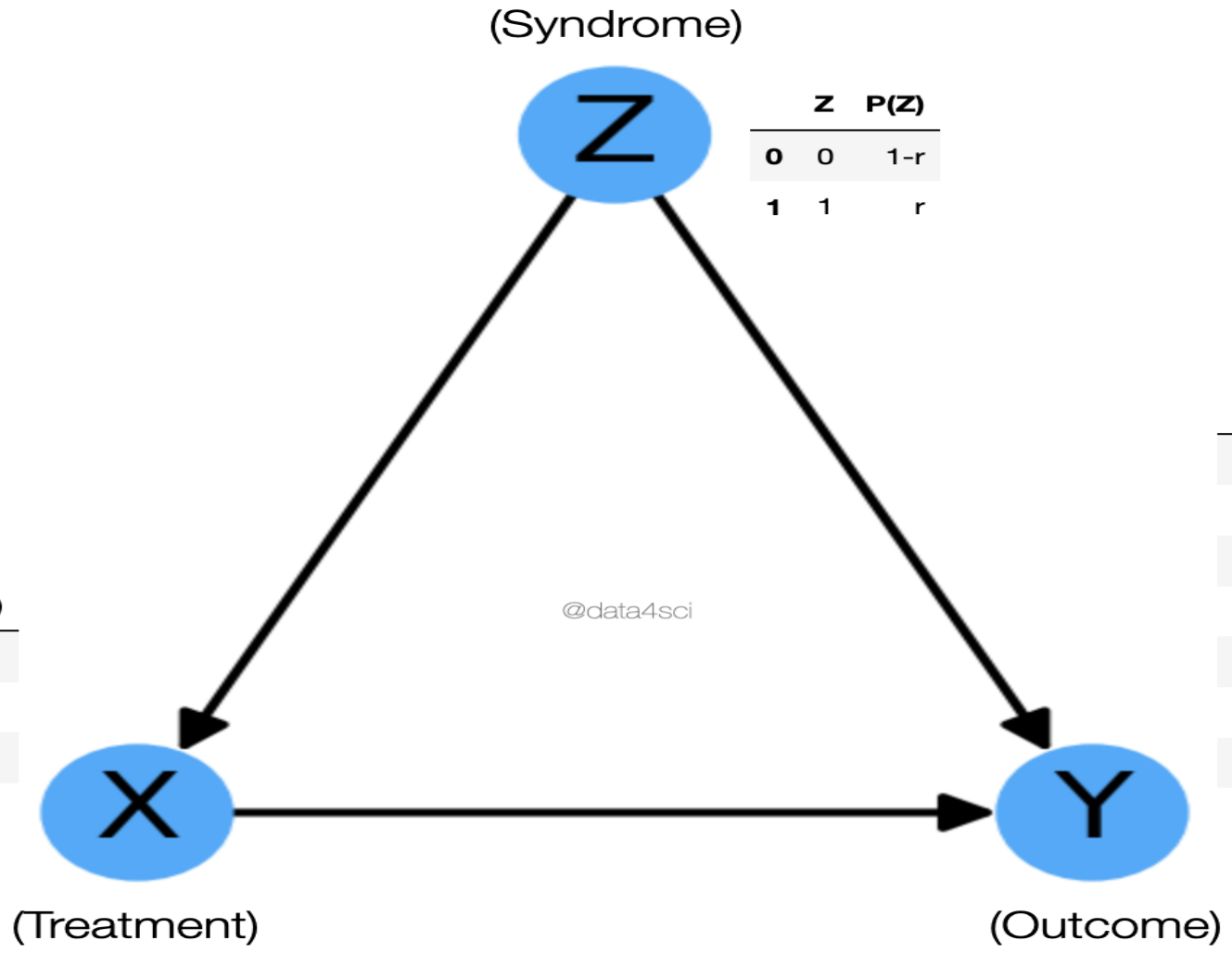
- Based on the notion of conditional independence
- X, Y are conditionally dependent on each other
- X->Y: Y may take certain values **if** X takes certain values

- Can we estimate Prob(Y) using Prob(X|Y)?
- Can we estimate Prob(X) using Prob(Y|X)?

- Prob(X|Y) helps us to estimate Prob(Y) but Prob(Y|X) doesn't help us to estimate Prob(X): X -> Y

- Prob(Y | X=x) is different from Prob(Y | do(X=x))

Structural Causal Model



	Z	P(Z)
0	0	1-r
1	1	r

	X	Z	P(X | Z)
0	1	0	q_0
1	1	1	q_1
2	0	0	$1 - q_0$
3	0	1	$1 - q_1$

	Y	X	Z	P(Y | X,Z)
0	1	1	0	p_{10}
1	1	0	0	p_{00}
2	1	0	1	p_{01}
3	1	1	1	p_{11}
4	0	1	0	$1 - p_{10}$
5	0	0	0	$1 - p_{00}$
6	0	0	1	$1 - p_{01}$
7	0	1	1	$1 - p_{11}$

Spatio-temporal Anomaly Detection

Adway Mitra

Machine Learning for Earth System Sciences (AI60002)

IIT Kharagpur

Definition of Anomaly

- Earth Science definition: deviation from past behavior

$$Y(s,t) = X(s,t) - \mu(s,t)$$

- Data Science definition: deviation from other values

- Spatio-temporal Anomaly: deviation of values from spatio-temporal neighbors
- Easy problem: isolated anomalies
- Hard problem: bulk anomalies (anomaly events)

Spatio-temporal Dataset

	T1	T2	T3	T4	T5	T6
S1	24	26	22	30	31	23
S2	28	27	29	26	32	33
S3	25	29	20	26	25	22
S4	35	33	29	33	34	31
S5	33	31	28	31	24	26
S6	32	37	29	34	33	29

- Spatial neighbors: (S1,S2,S3) and (S4,S5,S6)

Spatio-temporal Dataset

	T1	T2	T3	T4	T5	T6
S1	24	26	22	30	31	23
S2	28	27	29	26	32	33
S3	25	29	20	26	25	22
S4	35	33	29	33	34	31
S5	33	31	28	31	24	26
S6	32	37	29	34	33	29

- |X(S3,T3) - X(S3,T2) | > thres, |X(S3,T3) - X(S2,T3) | > thres

Spatio-temporal Dataset

	T1	T2	T3	T4	T5	T6
S1	24	26	22	30	31	23
S2	28	27	29	26	32	33
S3	25	29	20	26	25	22
S4	35	33	29	33	34	31
S5	33	31	28	31	24	26
S6	32	37	29	34	33	29

- $|X(S6,T2) - X(S6,T1)| > thres$, $|X(S6,T2) - X(S5,T2)| > thres$

Spatio-temporal Dataset

	T1	T2	T3	T4	T5	T6
S1	24	26	22	30	31	23
S2	28	27	29	26	32	33
S3	25	29	20	26	25	22
S4	35	33	29	33	34	31
S5	33	31	28	31	24	26
S6	32	37	29	34	33	29

- $|X(S1,T5) - X(S3,T5)| > thres$, $|X(S2,T6) - X(S3,T6)| > thres$

Spatio-temporal Dataset

	T1	T2	T3	T4	T5	T6
S1	24	26	22	30	31	23
S2	28	27	29	26	32	33
S3	25	29	20	26	25	22
S4	35	33	29	33	34	31
S5	33	31	28	31	24	26
S6	32	37	29	34	33	29

- $|X(S1,T5) - X(S3,T5)| > thres$, $|X(S2,T6) - X(S3,T6)| > thres$

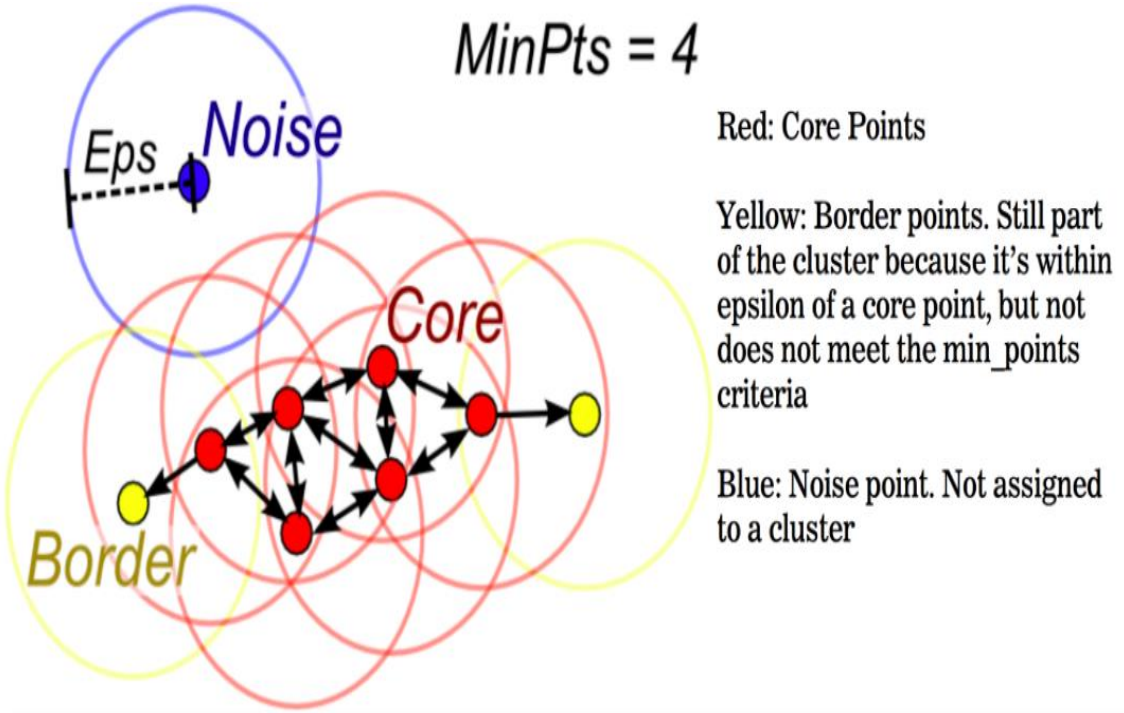
Anomaly Detection

- Simplest approach: compare with spatio-temporal neighbors or climatology using threshold
- Problems: i) Results totally sensitive on threshold
  ii) Comparison with neighbors can’t catch “bulk anomalies”
  iii) Climatology may not be available

- Alternatives: i) Clustering
  ii) Latent variable models

Clustering: DB-SCAN

- Density-based Spatial Clustering of Applications with Noise
- Idea: for each point, identify “neighbors” in feature space, and add them in cluster.
- Those points which could not be added to any cluster outlier!
- Need to specify distance threshold



Spatio-temporal clustering: DBSCAN

- Each data-point has spatial and temporal neighbors
- Each data-point can join only the clusters of its spatial or temporal neighbors
- Joining cluster on the basis of values
- If it cannot join any such cluster, then it is an outlier/anomaly!

	T1	T2	T3	T4	T5	T6
S1	24	26	22	30	31	23
S2	28	27	29	26	32	33
S3	25	29	20	26	25	22
S4	35	33	29	33	34	31
S5	33	31	28	31	24	26
S6	32	37	29	34	33	29

Spatio-temporal clustering: DBSCAN

- Each data-point has spatial and temporal neighbors
- Each data-point can join only the clusters of its spatial or temporal neighbors
- Joining cluster on the basis of values
- If it cannot join any such cluster, then it is an outlier/anomaly!

	T1	T2	T3	T4	T5	T6
S1	24	26	22	30	31	23
S2	28	27	29	26	32	33
S3	25	29	20	26	25	22
S4	35	33	29	33	34	31
S5	33	31	28	31	24	26
S6	32	37	29	34	33	29

Spatio-temporal clustering: DBSCAN

- Each data-point has spatial and temporal neighbors
- Each data-point can join only the clusters of its spatial or temporal neighbors
- Joining cluster on the basis of values
- If it cannot join any such cluster, then it is an outlier/anomaly!

	T1	T2	T3	T4	T5	T6
S1	24	26	22	30	31	23
S2	28	27	29	26	32	33
S3	25	29	20	26	25	22
S4	35	33	29	33	34	31
S5	33	31	28	31	24	26
S6	32	37	29	34	33	29

Can’t handle extended/bulk anomalies!

Alternative-1: Multiple scales

- Difficult to identify bulk anomalies at a single spatial/temporal scale
- Smoothen/coarsen the data at several levels
- Merge locations and consider their mean values
- Merge time-points and consider their mean values
- Repeat same process on coarsened dataset

	T1	T2	T3	T4	T5	T6
S1	24	26	22	30	31	23
S2	28	27	29	26	32	33
S3	25	29	20	26	25	22
S4	35	33	29	33	34	31
S5	33	31	28	31	24	26
S6	32	37	29	34	33	29

	T1	T2	T3	T4	T5	T6
S1-S2	26	27	26	28	32	28
S2-S3	26	28	25	26	29	28
S4-S5	34	32	29	32	29	29
S5-S6	33	34	29	33	29	28

Alternative-1: Multiple scales

- Difficult to identify bulk anomalies at a single spatial/temporal scale
- Smoothen/coarsen the data at several levels
- Merge locations and consider their mean values
- Merge time-points and consider their mean values
- Repeat same process on coarsened dataset

	T1	T2	T3	T4	T5	T6
S1	24	26	22	30	31	23
S2	28	27	29	26	32	33
S3	25	29	20	26	25	22
S4	35	33	29	33	34	31
S5	33	31	28	31	24	26
S6	32	37	29	34	33	29

	T1	T2	T3	T4	T5	T6
S1-S2	26	27	26	28	32	28
S2-S3	26	28	25	26	29	28
S4-S5	34	32	29	32	29	29
S5-S6	33	34	29	33	29	28

Anomalies may get smoothed out!

Alternative-2: latent variable models

- At each (s,t) define discrete latent variable Z(s,t)
- Z(s,t) can take two values (anomaly/ no anomaly) or three values (no anomaly/positive anomaly/negative anomaly)
- X(s,t) is a random variable with value known
- $X(s,t) \sim f(p_{s,t,k})$ where k = Z(s,t)
- Z(s,t) may also depend on Z(s,t-1) or Z(s’,t) where (s,s’) are neighbors
- Values of Z estimated by Gibbs Sampling

Alternative-2: latent variable models

Example:

Observation of X(s,t) = 34.3

f: Gaussian distribution

Ps,t,1 = [30,10], ps,t,2 = [8, 20]

prob(X(s,t)=34.3 | Z(s,t)=1) = N(34.3; [30,10])

prob(X(s,t)=34.3 | Z(s,t)=2) = N(34.3; [8,20])

prob(Z(s,t) = 1 | X(s,t)=34.3) = ???? (Bayes Theorem)

Z(s,t) should also depend on Z(s,t-1), Z(s’,t) etc for bulk anomalies

Alternative-2: latent variable models

- Can handle bulk anomalies
- Used to identify “anomaly events” like heat waves or droughts

	T1	T2	T3	T4	T5	T6
S1	24	26	22	30	31	23
S2	28	27	29	26	32	33
S3	25	29	20	26	25	22
S4	35	33	29	33	34	31
S5	33	31	28	31	24	26
S6	32	37	29	34	33	29

Z	T1	T2	T3	T4	T5	T6
S1	1	1	1	2	2	1
S2	1	1	1	1	2	2
S3	1	1	3	1	1	1
S4	1	1	1	1	1	1
S5	1	1	1	1	2	2
S6	1	2	1	1	1	1

Spatio-temporal Extreme Events

AI60002

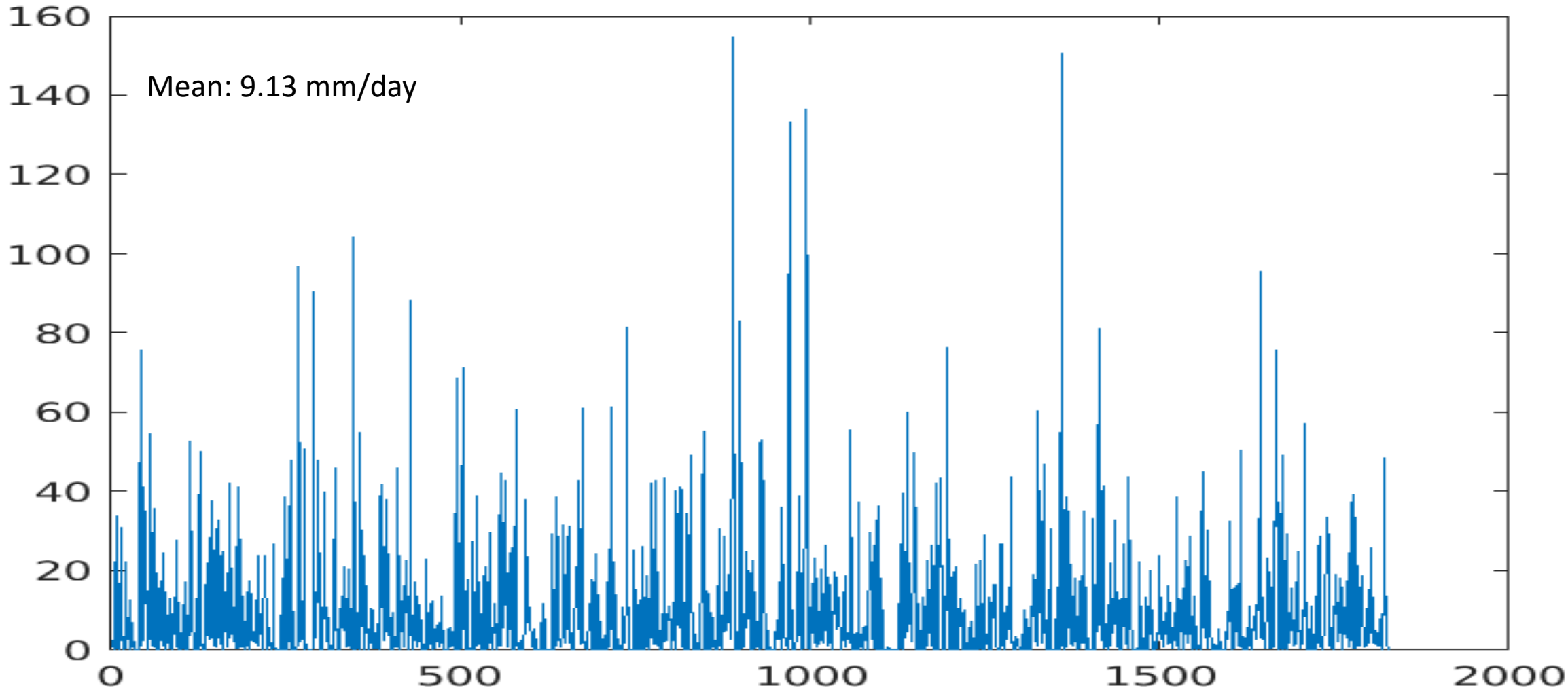
9th Feb 2021

Anomaly

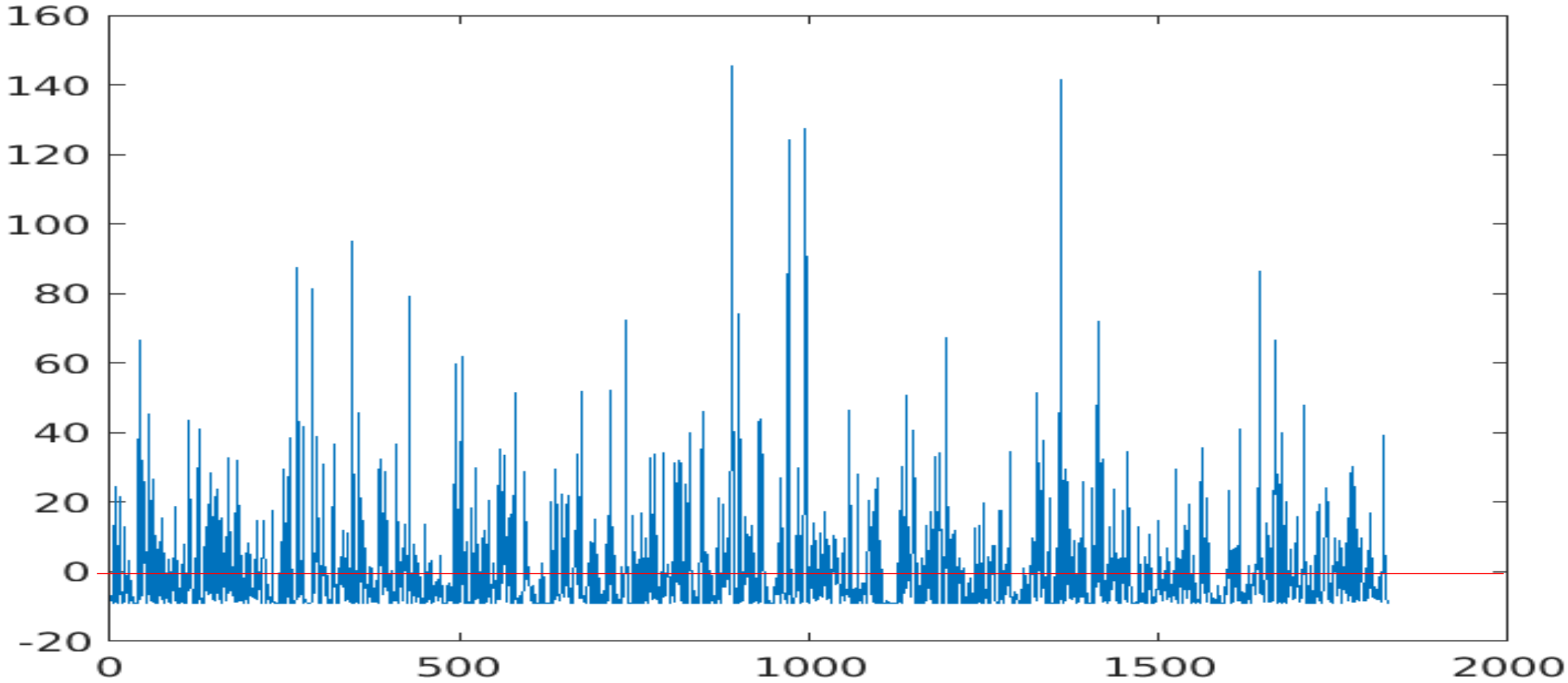
- Anomaly = Observation – expected value
- $Y(s,t) = X(s,t) - \mu(s,t)$
- $Y(s,t) > 0$: positive anomaly, $Y(s,t) < 0$: negative anomaly
- $Y(s,t) > \eta_U$: positive extreme event
- $Y(s,t) < -\eta_L$: negative extreme event

- η_U, η_L are usually double standard deviation of observations
- In some situations, one of the extreme events may not make sense

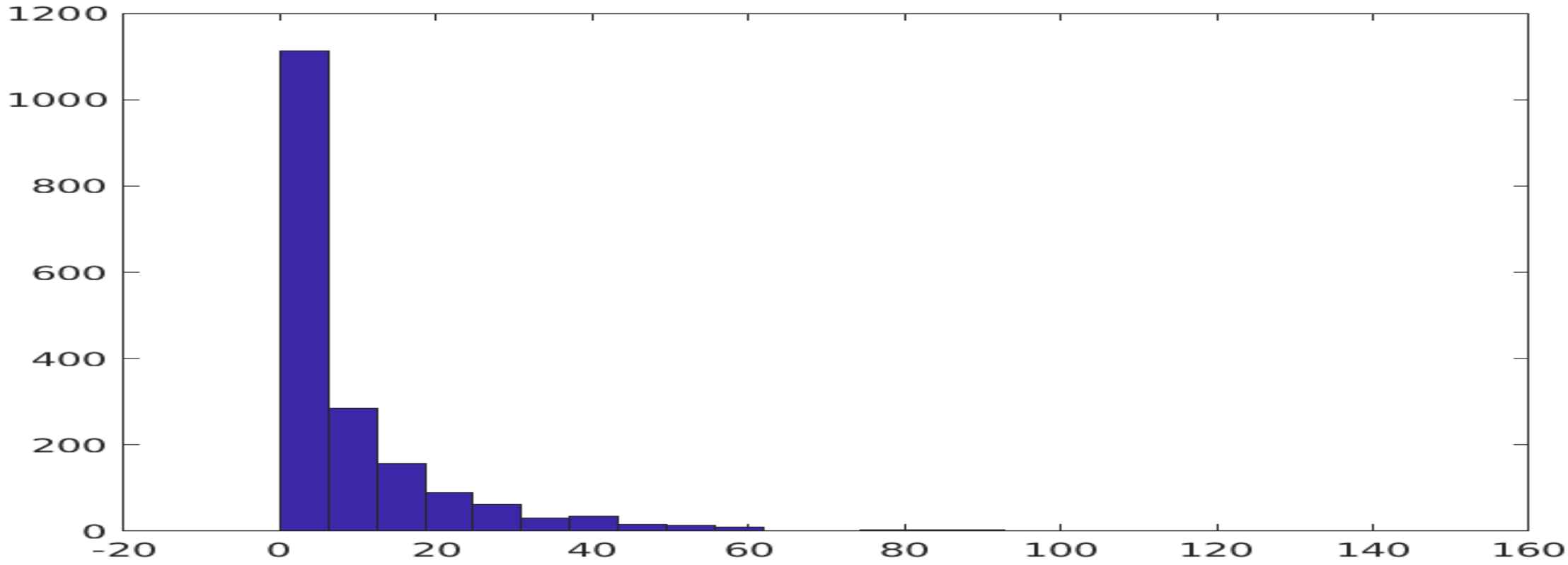
Daily Rainfall in Kharagpur (2000-2014, Jun-Sep)



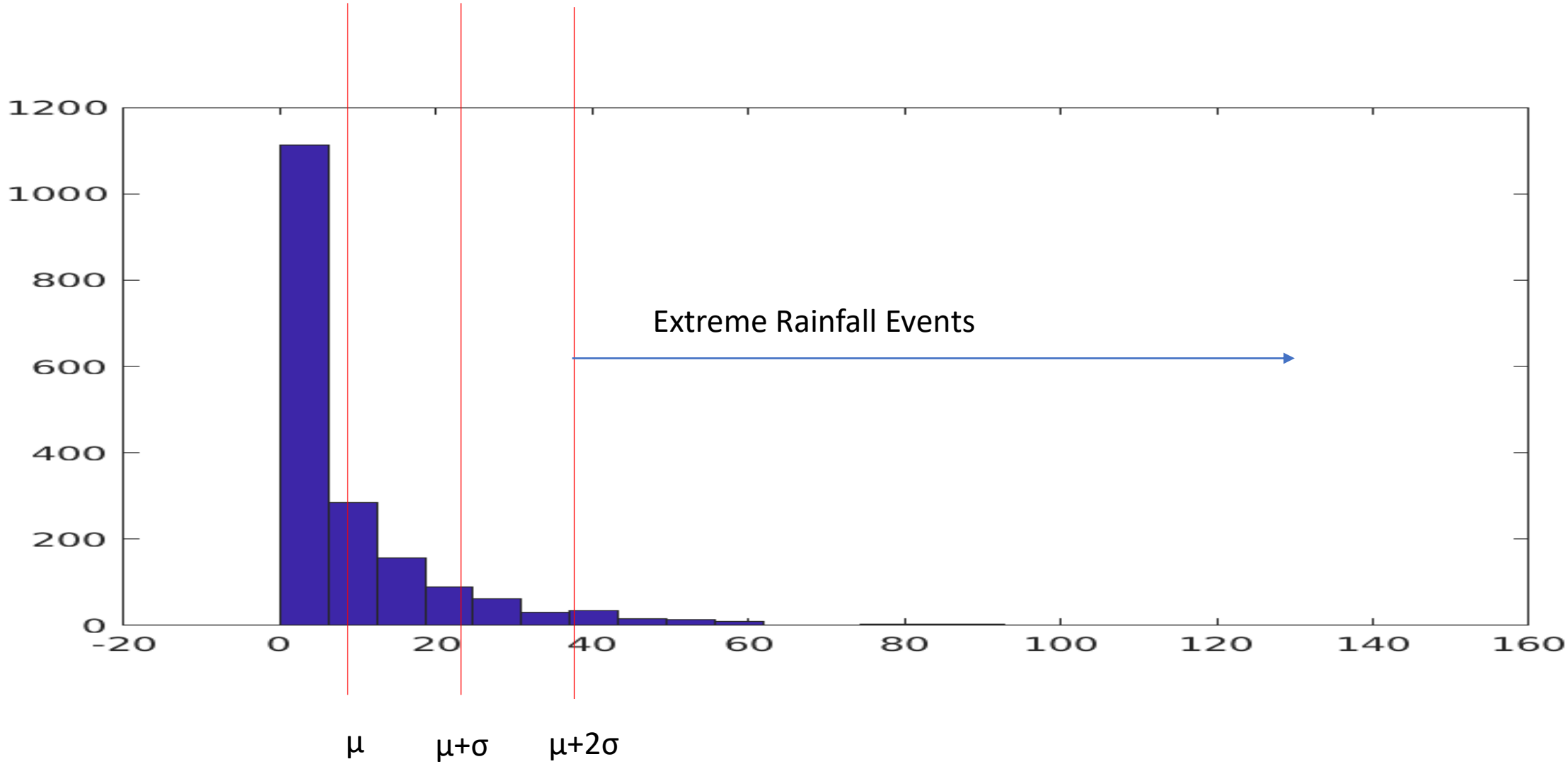
Daily Rainfall Anomaly in Kharagpur (2000-2014, Jun-Sep)



Histogram of Daily Rainfall in Kharagpur



Histogram of Daily Rainfall in Kharagpur



Percentiles

- Maximum rainfall in Kharagpur: 5th July 2007: 154 mm rainfall!
- It was caused by a deep depression in Bay of Bengal
- Quantile: cut-off points in probability distribution
- p-th Percentile = x: “p” percent of times, observation < x!

p	P-th Percentile in KGP	Frequency in 2000-2014
0.99	69 mm/day	18 days
0.9	25 mm/day	183 days
0.75	12 mm/day	457 days
0.5	3.4 mm/day	915 days

Skewed Distribution

- Mean daily rainfall at Kharagpur: 9.13 mm/day
- Median daily rainfall at Kharagpur: 3.4 mm/day!
- Median < Mean: On most of the days, KGP receives less rainfall than mean!
- 573/1830 days: more rainfall than the mean!
- Negative anomaly more frequent than Positive anomaly!
- Most days are “dry”, rainfall concentrated in a few “wet” days!
- Skewed distribution!

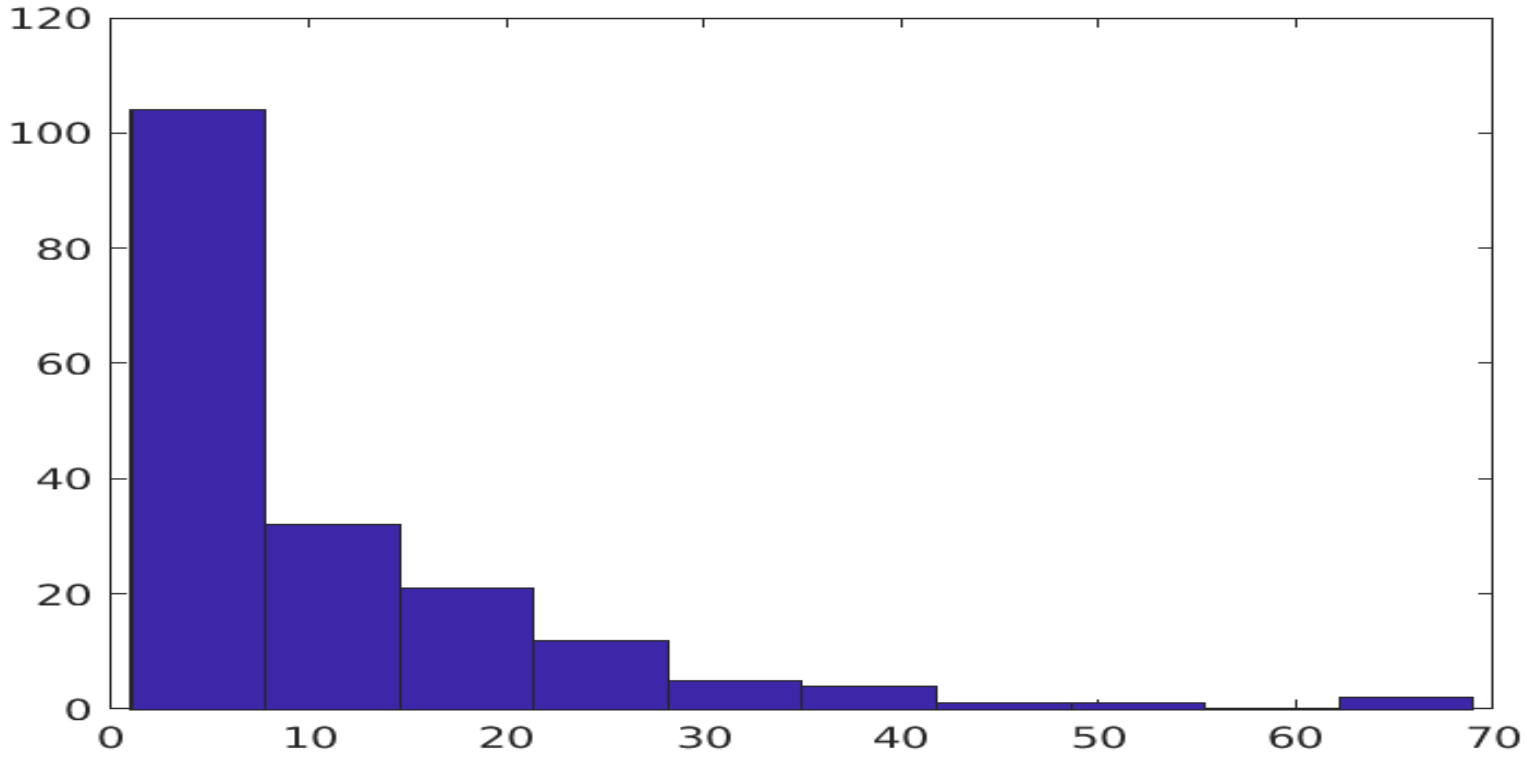
Return Periods

- Suppose any event happens today, when can you expect it to happen next?
- Common events may happen soon afterwards, rare events may happen much later!
- Return period = 1/p, where p is event probability
- Follows from Geometric Distribution
- 90% Percentile event: p=1-0.9 = 0.1, return period: 10 days
- i.e. expected difference between such events:10 days!

But actually …..

It seems that such events are “clustered” in time

Differences between two “90th-Percentile events” most likely to be <10 days!



Temporal Coherence of Anomaly

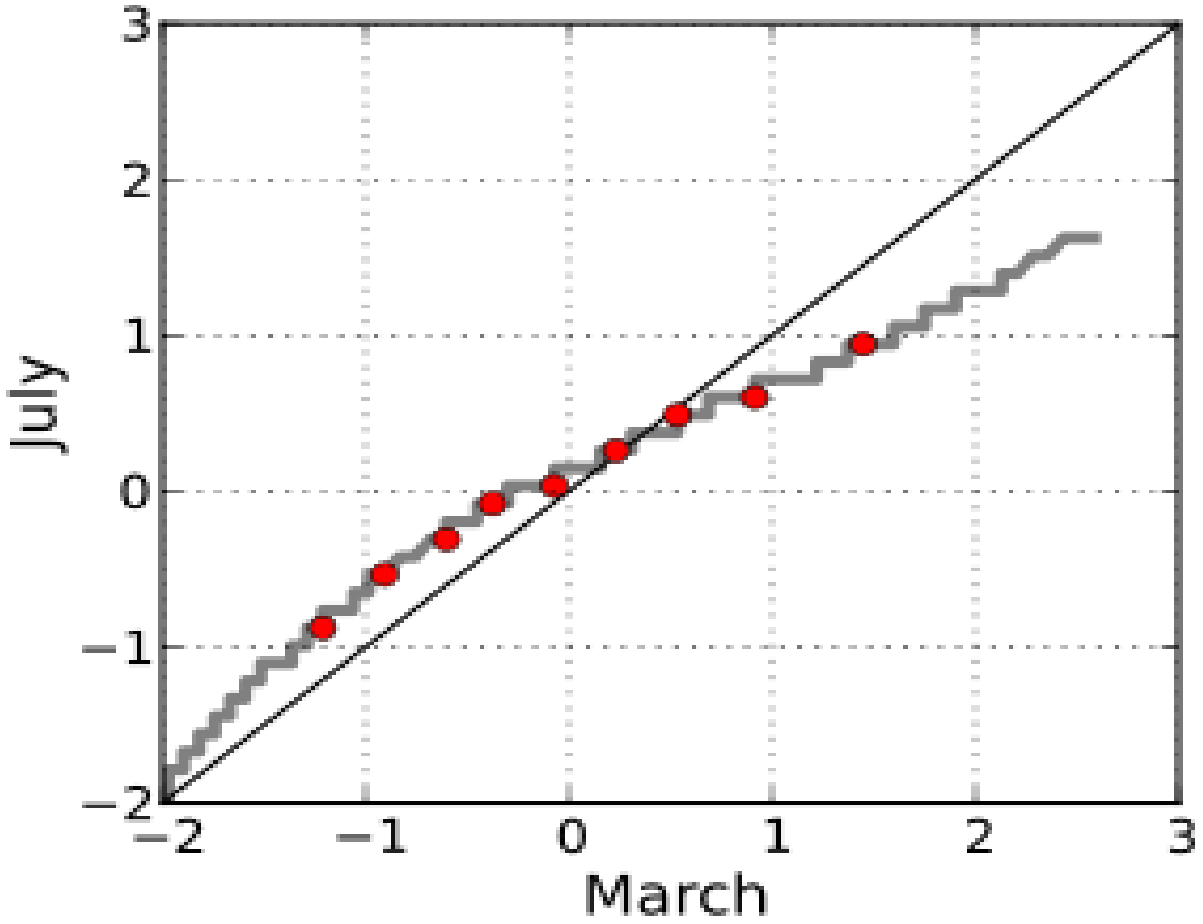
- Today positive anomaly => tomorrow positive anomaly (probably)
- KGP: prob(tomorrow positive anomaly) = 0.31
- KGP: prob(tomorrow positive anomaly | today positive anomaly) = 0.46!

- KGP: prob(tomorrow 90%-quantile event) = 0.1
- KGP: prob(tomorrow 90%-quantile event | today 90%-quantile event) = 0.25!

Spatial Coherence of Anomaly

- Anomalies are usually spatially coherent
- If one location has a positive anomaly, usually its surrounding locations also have it
- Whenever KGP has a positive rainfall anomaly during monsoon, on 60% occasions its surrounding locations also have positive rainfall anomaly!
- 90^{th} -Percentile rainfall in KGP => 90^{th}-Percentile in neighboring regions in 45% cases!
- 90^{th} -Percentile rainfall in KGP => 80^{th}-Percentile in neighboring regions in 60% cases!

Q-Q (Quantile-quantile plot)



A Q–Q plot comparing the distributions of standardized daily maximum temperatures at 25 stations in the US state of Ohio in March and in July. The curved pattern suggests that the central quantiles are more closely spaced in July than in March, and that the July distribution is skewed to the left compared to the March distribution. The data cover the period 1893–2001.

Wikipedia

Extreme Event Definitions

- PoT (Peak over Threshold) - An observation with high magnitude of anomaly
- Should also be a “peak” in the time-series (higher than neighbors)
- Threshold may be i) $\mu+2\sigma$

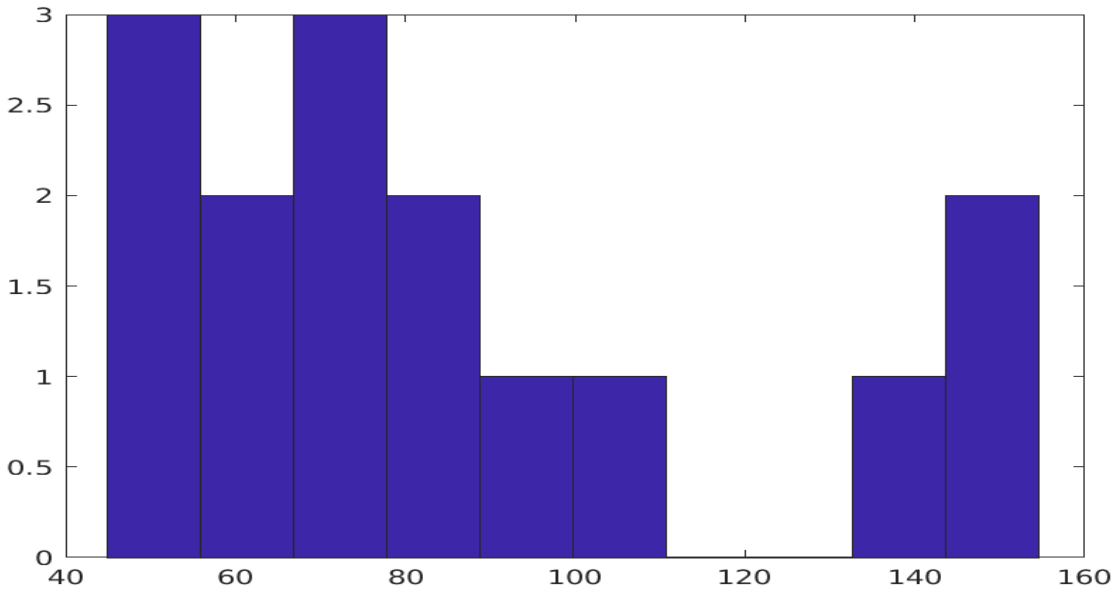
  ii) quantile (often 90^{th}, 95^{th}, 99^{th})
- Second definition: Block-maxima/minima
- Take a “set” of observations and calculate their maxima/minima
- The “set” can be spatial or temporal

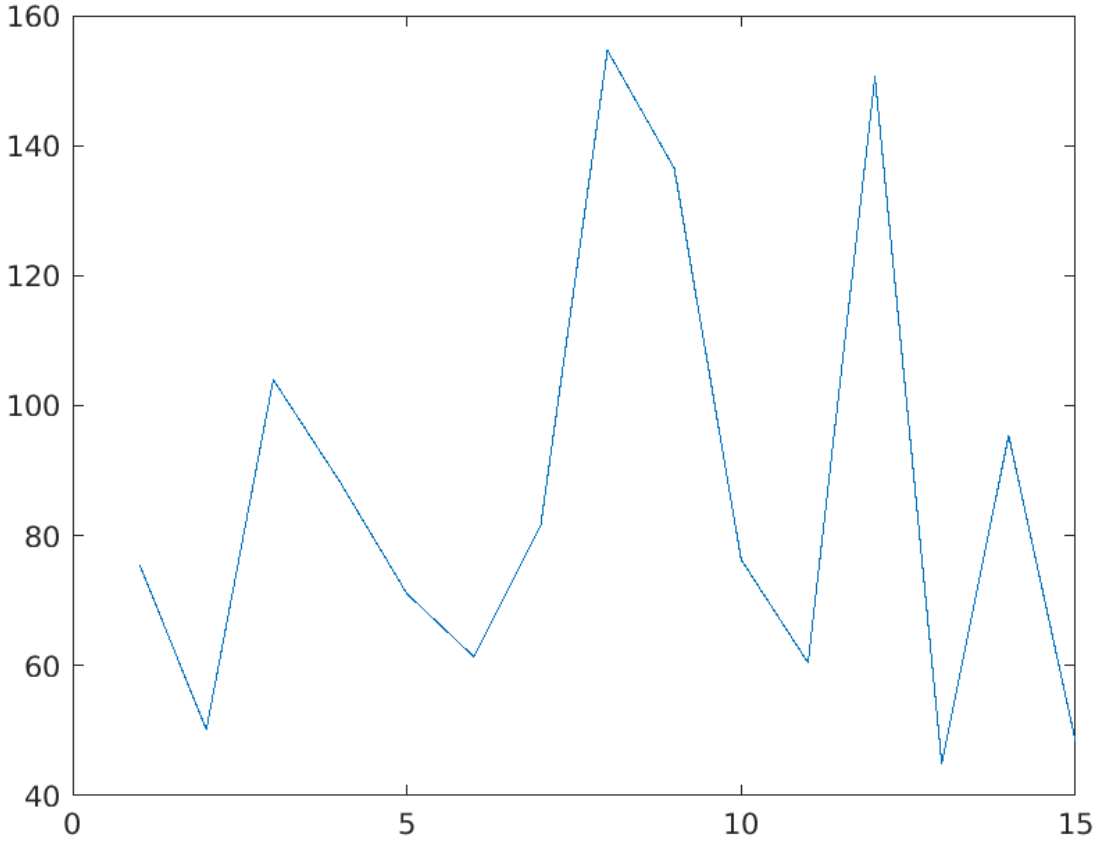
Block-wise Extremes

- Seasonal maximum precipitation over KGP
- Instead of dealing with all observations, we now deal with only the max. values

  Advantage: less variance

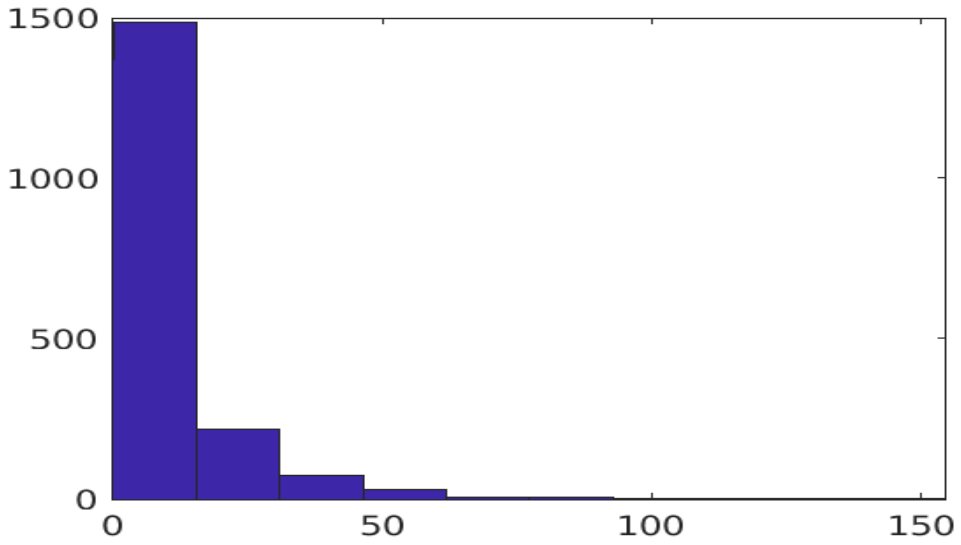
  Problem: less data

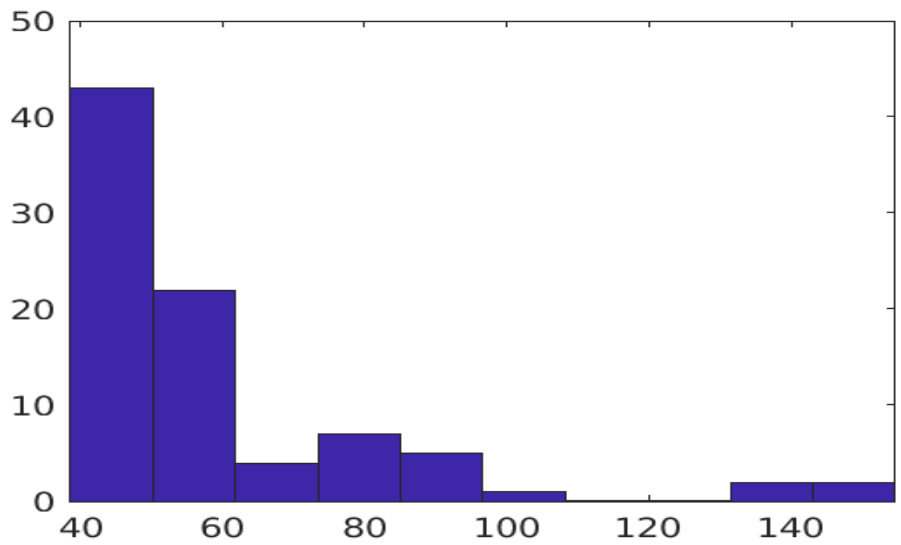




Extreme-Value Theory

- We focus on only the values at the “tail” of the distribution!
- Very different from the original distribution
- Original distribution more left-skewed than extreme distribution
- Needs a new distribution: Extreme-Value Distribution!





Parameter Estimation: Maximum-Likelihood and Bayesian

Adway Mitra
MLFA AI42001 Center for Artificial Intelligence
Indian Institute of Technology Kharagpur

October 10, 2019

Common Discrete Distributions

Distribution	Support	PMF	Parameters
Bernoulli	$\{0, 1\}$	$p^x(1-p)^{(1-x)}$	p
Binomial	$\mathcal{Z}$	$\binom{N}{x} p^x (1-p)^{N-x}$	N, p
Poisson	$\mathcal{Z}$	$\frac{e^{-\lambda}\lambda^x}{x!}$	λ
Geometric	$\mathcal{Z}^+$	$(1-p)^{x-1}p$	p
Categorical	$\{V_1, \ldots, V_K\}$	$\prod_{k=1}^{K} p_k^{I(x=k)}$	$(p_1, p_2, \ldots, p_k)$
Multinomial	$\mathcal{Z}^K$	$\frac{N!}{n_1!..n_K!} \prod_{k=1}^{K} p_k^{n_k}$	$(N_1, p_1, \ldots, N_K, p_K)$

Common Continuous Distributions

Distribution	Support	PDF	Parameters
Beta	$(0, 1)$	$\frac{1}{B(a,b)} x^{(a-1)}(1-x)^{(b-1)}$	(a, b)
Gamma	$\mathcal{R}+$	$\frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} exp(-\beta x)$	(α, β)
Gaussian	$\mathcal{R}$	$\frac{1}{\sqrt{(2\pi)}\sigma} exp(-\frac{1}{2\sigma^2}(x-\mu)^2)$	(μ, σ)
M.V. Gaussian	$\mathcal{R}^D$	$\frac{1}{2\pi^{\frac{D}{2}} \lvert\Sigma\rvert^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)$	(μ, Σ)

Parameter Estimation Problem

- Given: N observations $x_1, x_2, \ldots, x_N$
- Imagine these observations are observations of IID random variables
- Choose a suitable distribution for them
- Support, histogram important considerations to choose distribution
- Need parameters for the distribution!

Maximum Likelihood Estimation (MLE)

- Write down joint PMF/PDF of the observations
- $prob(X_1 = x1, X_2 = x_2, \ldots, X_N = x_N) = \prod_{n=1}^{N} prob(X_n = x_n)$
- This is also called **likelihood function** $\mathcal{L}(p)$ of parameters p of the distribution (prob)
- Choose parameters such that this likelihood is maximized!
- Differentiate w.r.t p, equate to 0, solve equations!

MLE of Bernoulli

- Input: results of N tosses, $X_i \in \{0, 1\}$
- Based on support, we choose Bernoulli Distribution
- Need to estimate parameter p

$$\begin{aligned} \mathcal{L}(p) = \prod_{n=1}^{N} prob(X_n = x_n) &= \prod_{n=1}^{N} p^{x_n}(1-p)^{1-x_n} \\ &= p^{N_1}(1-p)^{N_0} \end{aligned} \tag{1}$$

$$p_{MLE} = argmax_p \mathcal{L}(p) = \frac{N_1}{N_1 + N_0} \tag{2}$$

MLE for Poisson

- Input: N integer observations, $X_i \in \mathcal{Z}$
- Based on support and histogram, we may choose Poisson Distribution
- Need to estimate parameter λ

$$\begin{aligned} \mathcal{L}(\lambda) = \prod_{n=1}^{N} prob(X_n = x_n) &\propto \prod_{n=1}^{N} e^{-\lambda} \lambda^{x_n} \\ &= e^{-N\lambda} \lambda^{\sum_{n=1}^{N} x_n} \end{aligned} \tag{3}$$

$$\lambda_{MLE} = argmax_\lambda \mathcal{L}(\lambda) = \frac{\sum_{n=1}^{N} x_n}{N} \tag{4}$$

MLE for Gaussian

- Input: N real observations, $X_i \in \mathcal{R}$
- Based on support and histogram, we may choose Gaussian/Normal Distribution
- Need to estimate parameter (μ, σ)

$$\mathcal{L}(\mu,\sigma) = \prod_{n=1}^{N} prob(X_n = x_n) \propto \prod_{n=1}^{N} \frac{1}{\sigma} exp(-\frac{(x_n-\mu)^2}{\sigma^2})$$

$$= \frac{1}{\sigma^N} exp(-\sum_{n=1}^{N} \frac{(x_n-\mu)^2}{\sigma^2}) \quad (5)$$

$$\mu_{MLE}, \sigma_{MLE} = argmax_{\mu,\sigma} \mathcal{L}(\mu,\sigma)$$

$$\mu_{MLE} = \frac{\sum_{n=1}^{N} x_n}{N}, \sigma_{MLE} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{MLE})^2 \quad (6)$$

MLE for Multivariate Gaussian

- Input: N real vector observations, $X_i \in \mathcal{R}^D$
- Based on support and histogram, we may choose Gaussian/Normal Distribution
- Need to estimate parameter (μ, Σ)

$$\mathcal{L}(\mu, \Sigma) = \prod_{n=1}^{N} prob(X_n = x_n) \quad \propto \quad \prod_{n=1}^{N} \frac{1}{|\Sigma|^{\frac{1}{2}}} exp(-(x_n - \mu)^T \Sigma^{-1} (x_n - \mu)$$

$$= \quad \frac{1}{|\Sigma|^{\frac{N}{2}}} exp(-\sum_{n=1}^{N} (x_n - \mu)^T \Sigma^{-1} (x_n - \mu) \quad (7)$$

$$\mu_{MLE}, \Sigma_{MLE} = argmax_{\mu,\Sigma} \mathcal{L}(\mu, \Sigma)$$

$$\mu_{MLE} = \frac{\sum_{n=1}^{N} x_n}{N}, \Sigma_{MLE} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu_{MLE})(x_n - \mu_{MLE})^T \quad (8)$$

Bayesian Parameter Estimation

- Maximum likelihood estimate - entirely based on data
- But if data is not reliable?
- Bayesian approach: we may have some prior beliefs
- Bayesian approach: combine our prior beliefs with evidence, i.e. data
- Bayesian approach: keep updating our beliefs as more and more data comes in!

Bayesian Parameter Estimation

- Consider the parameters as random variables
- Put a prior distribution on the parameters
- $posterior(param|data) \propto prob(data|param) * prior(param)$
- $prob(data|param) = \mathcal{L}(param)$ (likelihood function)
- Difference from MLE - we get a distribution of the parameter instead of single value
- Maximum A-Posteriori (MAP) estimate:
  $param_{Bayes} = argmax_{param} posterior(param|data)$

Bayesian Parameter Estimation

- How to choose prior distribution?
  - Reflect our belief on parameter
  - Mathematical tractability (posterior should be a valid distribution)
- Some likelihood functions have conjugate prior
- Prior and Posterior on parameters should be same distribution with different parameters!
  - Easy to interpret

Bayesian estimate of Bernoulli Distribution

- Data: $\{x_1, \ldots, x_N\} \in \{0, 1\}$, Model: $X_i \sim Bernoulli(p)$
- $p \in (0, 1)$, so $prior(p) = Beta(a, b)$
  - (a,b) hyperparameters - parameters of prior
  - Assume $a = b$ if no information

$$\begin{aligned} posterior(p|X) &\propto \prod_{i=1}^{N} prob(X_i = x_i|p) * prior(p) \\ &= p^{N_1}(1-p)^{N_0} * p^{a-1}(1-p)^{b-1} \\ &= p^{N_1+a-1}(1-p)^{N_0+b-1} \end{aligned} \tag{9}$$

- $posterior(p) : Beta(N_1 + a, N_0 + b)$
- $p_{Bayes} = \frac{N_1+a}{N+a+b}$

Bayesian estimate of Bernoulli parameters

- $prior(p) = Beta(5, 7)$, $p_{MAP} = 5/12$
- $X_1 = TAIL$, $posterior(p) = Beta(5, 8)$, $p_{MAP} = 5/13$
- $X_2 = HEAD$, $posterior(p) = Beta(6, 8)$, $p_{MAP} = 6/14$
- $X_3 = HEAD$, $posterior(p) = Beta(7, 8)$, $p_{MAP} = 7/15$
- $X_4 = TAIL$, $posterior(p) = Beta(7, 9)$, $p_{MAP} = 7/16$
- $X_5 = HEAD$, $posterior(p) = Beta(8, 9)$, $p_{MAP} = 8/17$
- $X_6 = HEAD$, $posterior(p) = Beta(9, 9)$, $p_{MAP} = 9/18$

Bayesian estimate of Gaussian Distribution - variance known

- Data: $\{x_1, \ldots, x_N\} \in \mathcal{R}$, Model: $X_i \sim \mathcal{N}(\mu, \sigma)$
- $\mu \in \mathcal{R}$, so $prior(\mu) = \mathcal{N}(\mu_0, \sigma_0)$
- Assume σ is known for simplicity

$$
\begin{aligned}
posterior(\mu|X) &\propto \prod_{i=1}^{N} prob(X_i = x_i|\mu) * prior(\mu) \\
&= \frac{1}{\sigma^N} exp(-\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{2\sigma^2}) * \frac{1}{\sigma_0} exp(-\frac{(\mu - \mu_0)^2}{2\sigma_0^2}) \\
&= \mathcal{N}(\frac{\frac{N}{\sigma^2}\hat{X} + \frac{\mu_0}{\sigma_0^2}}{\frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}}, \frac{1}{\frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}}) \qquad (10)
\end{aligned}
$$

where $\hat{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$

Bayesian estimate of Gaussian Distribution - Variance Unknown

- Data: $\{x_1, \ldots, x_N\} \in \mathcal{R}$, Model: $X_i \sim \mathcal{N}(\mu, \tau)$ where $\tau = \frac{1}{\sigma^2}$
- Define $prior(\mu, \tau) = prior_1(\mu|\tau), prior_2(\tau)$
- $prior_1(\mu|\tau) = \mathcal{N}(\mu_0, \eta\tau)$, $prior_2(\tau) = Gamma(a, b)$
- $posterior(\mu, \tau) = \mathcal{L}(\mu, \tau, X) * prior_1(\mu|\tau) * prior_2(\tau)$
- $posterior(\mu) = \int posterior(\mu, \tau) d\tau : GaussianDistribution$
- $posterior(\tau) = \int posterior(\mu, \tau) d\mu : GammaDistribution$

A Template Problem in Spatio-temporal Modeling and Data Mining

Adway Mitra

4 January 2021

Notations

- Consider S locations in a region
- A geo-physical variable, say X may be measured at every location
- Readings are taken at regular time intervals, say hourly/daily
- Denote the readings by X^s_{dh} (s: location, d: day, h: hour)
- Or maybe, X^s_{ymd} (y: year, m: month, d: day)
- Observations are available at only a subset of the locations!!

Template Setting

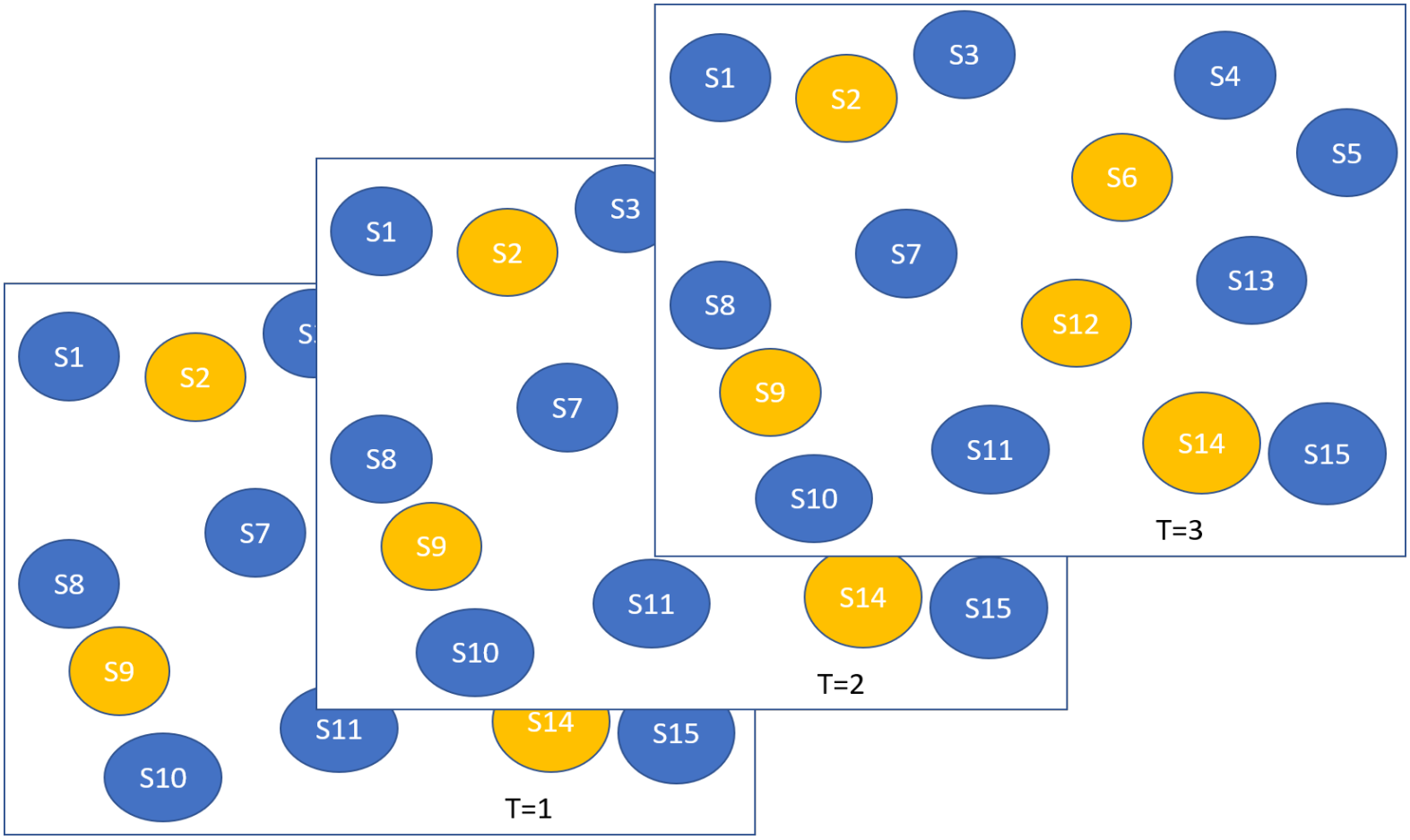


Figure: 15 locations: observations available in blue locations, not in orange locations

Template Problems

- Estimate the values of X at the locations which have no observations
- Predict future values at all locations
- Identify spatial relationships between locations
- Identify trends and periodic/seasonal behavior
- Identify "anomalies" or unusual events

Probabilistic Modeling

- Consider $\{X\}$ as random variables, whose values are sometimes known
- Each of them can be considered as separate R.V. (not useful)
- Each of them can be considered as a realization of the same R.V. (may not make sense physically)
- We can divide them into groups - all values in same group are realizations of one R.V.?
- How to define such groups?

Probabilistic Modeling

- Let us consider X_h^s as a R.V. (value at location s at hour h) *(Total* $24S$ *variables)*
- Its realizations are available for each day: $\{x_{dh}^s\}$
- We utilize the property of periodicity (values at same location, same hour likely to be similar on different days)
- Similarly, we can define X_m^s as a R.V. (value at location s for month m) *(Total* $12S$ *variables)*
- Its realizations are available for each year and each day: $\{x_{ymd}^s\}$

Probabilistic Modeling

- What sort of R.V. is X_h^s or X_m^s?
- Continuous or discrete?
  - *Decide based on the nature of the data*
- Follows which distribution?
  - *Decide based on the histogram of the data*
- Parameters of the distribution?
  - *Parameter Estimation techniques!*

Introduction to Spatio-temporal Statistics

Adway Mitra

AI60002: Machine Learning for Earth System Sciences

18 January 2021

Groups of Random Variables

- We are recording hourly temperature at a particular location, every day
- Denote by $x_{d,t}$, the temperature reading on day d, hour t
- We wish to consider x as a random variable
- Option 1: $X_{d,t} \sim f$
- Option 2: $X_{d,t} \sim f_{d,t}$
- Option 3: $X_{d,t} \sim f_d$
- Option 4: $X_{d,t} \sim f_t$

Groups of Random Variables

- We are recording hourly temperature at a particular location, every day
- Denote by $X_{d,t}$, the temperature reading on day d, hour t
- We wish to consider X as a random variable
- Option 1: $X_{d,t} \sim f$ all observations are IID
- Option 2: $X_{d,t} \sim f_{d,t}$ all observations are separate RVs
- Option 3: $X_{d,t} \sim f_d$ Separate distribution for each day
- Option 4: $X_{d,t} \sim f_t$ Separate distribution for each hour

Groups of Random Variables

- We are recording hourly temperature at a particular location, every day
- Denote by $X_{d,t}$, the temperature reading on day d, hour t
- We wish to consider X as a random variable
- Option 1: $X_{d,t} \sim f$ fails to capture variations
- Option 2: $X_{d,t} \sim f_{d,t}$ infeasible, not beneficial
- Option 3: $X_{d,t} \sim f_d$ fails to capture hourly variations
- Option 4: $X_{d,t} \sim f_t$ fails to capture seasonal variations

Temporal Auto-correlation

- Suppose we are focusing on one season only.
- We go for Option 4: $X_{d,t} \sim f_t$
- $\{x_{1,t}, x_{2,t}, \dots\}$ are realizations of X_t
- Missing out: relationship between hours!
- $Corr(X_{t_i}, X_{t_j})$: correlation coefficient between the Random variable for two different hours
- Example of **temporal autocorrelation**!
- Autocorrelation may be high or low, based on t_i and t_j

Temporal Auto-correlation

- Consider a set of temporal variables $\{X_{t1}, X_{t2}, \dots\}$
- **Mean stationarity:** $E(X_{ti}) = m$, i.e. constant
- **Covariance stationarity:** $Cov(X_{ti}, X_{tj}) = C_t(ti - tj)$,
- C_t is called **Temporal Covariance Function**
- Covariance stationarity implies *temporal autocorrelation* between X_{ti}, X_{tj} only a function of $(ti - tj)$
- **Weak stationarity**: Mean stationarity + Covariance stationarity + finite $E(|X_{ti}|^2)$

Temporal Auto-regression

- Can we express one temporal variable as a function of others?
- If $Corr(X_{ti}, X_{tj}) \neq 0$, can we have $X_{tj} = f(X_{ti})$?
- Simplest assumption: linear relation
- $X_{tj} = aX_{ti} + b$, where b is a random variable (eg. white noise)
- *Order-1 autoregressive process*: $X_{t,i+1} = aX_{t,i} + b$
- *Order-K autoregressive process*: $X_{t,i+1} = \sum_{k=0}^{K-1} a_k X_{t,i-k} + b$

Spatial Autocorrelation

- Consider the rainfall measured every day at locations S_1, S_2, ...
- x_{st} = rainfall at location s, day t
- $X_{st} \sim f_s$
- *Spatial autocorrelation*: $Corr(X_{si}, X_{sj})$
- *Mean Stationarity*: $E(X_s) = c$ (constant)
- *Covariance stationarity*: $Cov(X_{si}, X_{sj}) = C_S(||s_i - s_j||)$, $||.||$ denotes distance
- C_S is called **Spatial Covariance Function**
- Implies: spatial autocorrelation between any two points is a function of their distance!

Spatial Covariance Function

- **First Law of Geography**: *everything is related to everything else, but near things are more related than distant things*
- $Corr(X_{si}, X_{sj})$ should have high magnitude if $||si - sj||$ is low
- $Corr(X_{si}, X_{sj})$ should have low magnitude if $||si - sj||$ is high
- Possible covariance function:
  $Cov(X_{si}, X_{sj}) = k.exp(-\gamma||si - sj||^2)$
- k can be positive or negative, γ: scaling constant
- Temporal covariance function may be defined analogously

Variogram

- Defined as the variance of the difference between the variable at two different locations
- Measure of spatial smoothness of X
- $\gamma(si, sj) = \frac{1}{2}E((X_{si} - X_{sj})^2)$
- In case of *weakly stationery process*, this reduces to $\gamma(si, sj) = Var(X_{si}) + Var(X_{sj}) - 2Cov(X_{si}, X_{sj})$
- Further, $Var(X_{si}) = Var(X_{sj}) = C_S(0)$, and $Cov(X_{si}, X_{sj}) = C_S(||si - sj||)$
- So, for weakly stationery process, $\gamma(si, sj) = C_S(0) - C_S(||si - sj||)$

Spatial Autoregression

- Can we express one spatial variable as a function of others?
- If $Corr(X_{si}, X_{sj}) \neq 0$, can we have $X_{sj} = g(X_{si})$?
- Simplest assumption: linear relation
- $X_{sj} = aX_{si} + b$, where b is a random variable (eg. white noise)
- *Order-K autoregressive process*: $X_{sj} = \sum_{k=0}^{K-1} a_k X_{i_k} + b$

Autoregression Parameter Estimation

- How to estimate the coefficients like a?
- For each day t, we have $b_t = X_{sj,t} - aX_{si,t}$
- $b_t \sim \mathcal{N}(0, 1)$
- The likelihood $p(b_1, b_2, \dots) \propto \prod_t exp(\frac{1}{2} b_t^2)$
- Log-likelihood function $\mathcal{L}(a) \propto -\sum_t (X_{sj,t} - aX_{si,t})^2$
- Take derivative of log-likelihood, equate to 0, solve for a

Gaussian Process for Spatio-temporal Processes, Inverse Problem by Sampling

Adway Mitra

1 February 2021

Multivariate Gaussian Distribution

- Consider random variables $X_1, X_2, \ldots, X_D$
- $X = \{X_1, X_2, \ldots, X_D\}$
- $p(X) = \frac{1}{(2\pi)^{D/2} det(\Sigma)} exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$
- Σ is the $D \times D$ covariance matrix, $\Sigma(i,j) = Cov(X_i, X_j)$
- Covariance matrix size blows up with D!
- Possible solution: replace covariance matrix by covariance function!

Gaussian Process

- Consider a (finite or infinite) set of random variables X_1, X_2, ...
- Suppose each of them represents a location
- Consider any random finite subset $\{X_{i1}, X_{i2}, \ldots, X_{iN}\}$
- Then we have $(X_{i1}, X_{i2}, \ldots, X_{iN}) \sim \mathcal{N}(\mu, \Sigma)$
- $\mu(s)$: mean function (a function of s)
- $\Sigma(s, s') = K(||s - s'||)$, where K is the covariance function (a function of $||s - s'||$)

Gaussian Process

- $X \sim \mathcal{GP}(\mu, K)$, where $\mathcal{GP}$ represents Gaussian Process, with K as the covariance function
- X is now a continuous function, defined at every location
- Any finite set of locations follows multivariate Gaussian distribution
- Relationship between X at different locations represented through covariance function
- Given observations at each location, we can predict the values at other locations

Interpolation using Gaussian Processes

- Consider any location $s+1$ where we want to predict X_{s+1}
- Conditional PDF $p(X_{s+1}|X_1,\ldots,X_s) = \frac{p(X_1,X_2,\ldots,X_s,X_{s+1})}{p(X_1,X_2,\ldots,X_s)}$
- Both joint PDFs $p(X_1,X_2,\ldots,X_s,X_{s+1})$ and $p(X_1,X_2,\ldots,X_s)$ are Gaussian!
- For both, mean vector and covariance matrix will come from the GP mean and covariance functions!

Spatio-temporal Hierarchical Gaussian Process

- Data Model: $X \sim \mathcal{N}(\mu + \eta, \sigma I)$
- $\eta = AZ + BY$
- $\mu \sim \mathcal{GP}(\mu_0, K_0)$?
- $Z \sim \mathcal{GP}(\mu_Z, K_Z)$?
- These Gaussian Processes can be either spatial or temporal
- We can decompose μ and Z into spatial and temporal components

Spatio-temporal Hierarchical Gaussian Process

- Data Model: $X(s,t) \sim \mathcal{N}(\mu(s,t) + \eta(s,t), \sigma)$
- Parameter Model: $\mu(s,t) = \mu_S(s)\mu_T(t)$
- $\mu_S \sim \mathcal{GP}(\mu_{S0}, K_{S0})$, $\mu_T \sim \mathcal{GP}(\mu_{T0}, K_{T0})$
- Process Model: $\eta(s,t) = AZ(s,t) + BY(s,t)$ where $Z(s,t) = Z_S(s)Z_T(t)$
- $Z_S \sim \mathcal{GP}(\mu_{S0}, K_{S0})$, $\mu_T \sim \mathcal{GP}(\mu_{T0}, K_{T0})$

Forward Problem: Data Generation

- Forward problem: generate/simulate X using this model
- Identify a set of locations and time-points to generate data
- Generate $\mu(s,t)$ from the Gaussian Processes μ_S and μ_T
- For each of μ_S and μ_T, first generate data at one location/time-point
- Keep sampling as $p(\mu_s|\mu_1,\ldots,\mu_{s-1})$ and $p(\mu_t|\mu_1,\ldots,\mu_{t-1})$
- Similarly generate $Z(s,t)$ from the Gaussian Processes Z_S and Z_T
- Finally generate $X(s,t)$ using the data model

Inverse Problem

- We already have the observation X and covariates Y
- Can we estimate Z, μ, σ, A, B etc?
- Z, μ are latent random variables, σ, A, B are parameters
- Let us assume that parameters are fixed and known
- We need to estimate Z and μ by *Gibbs Sampling*
- Assign initial values to all Z and μ variables
- Optimal values to be found by repeated sampling of each variable

Inverse Problem

- Sample a new value of $Z(s,t)$ from $p(Z(s,t)|Z,X,\mu)$, conditional distribution of $Z(s,t)$ based on all other variables
- Repeat this process for all $s \in \{1,S\}$ and $t \in \{1,T\}$
- Sample a new value of $\mu(s,t)$ from $p(\mu(s,t)|\mu,X,Z)$, conditional distribution of $\mu(s,t)$ based on all other variables
- Repeat this process for all $s \in \{1,S\}$ and $t \in \{1,T\}$
- Store the sampled values of each variable
- Repeat the whole process for many iterations
- For each variable, select the mode of its samples

ASSIGNMENT (10%)

Generate spatio-temporal data over a 20×20 grid system for 100 time-steps

1. Generate μ using a Gaussian Process (μ_S, μ_T) using suitable covariance functions
2. Generate Z using a different Gaussian Process
3. Select the covariates Y in your own way
4. Choose A, B, σ and generate X
5. Calculate the variogram between different pairs of locations and plot histogram as a function of distance