

HyperLabelMe:

A web-platform for benchmarking remote sensing image classifiers

Jordi Muñoz-Marí, Luis Gómez-Chova, Ana B. Ruescas
Gonzalo Mateo-García,, Emma Izquierdo-Verdiguier
Manuel Campos Taberner, Gustau Camps-Valls

Image Processing Lab (IPL)
Universitat de València



Artificial intelligence, deep and machine learning

ARTIFICIAL INTELLIGENCE

Early artificial intelligence stirs excitement



MACHINE LEARNING

Machine learning begins to flourish



DEEP LEARNING

Deep learning breakthroughs drive AI boom



AI: Intelligence demonstrated by machines rather than humans or animals.

ML: Giving computers the skills to learn without explicit programming

DL: Is an ML subset, examining algorithms that learn and improve on their own.

1950's

1960's

1970's

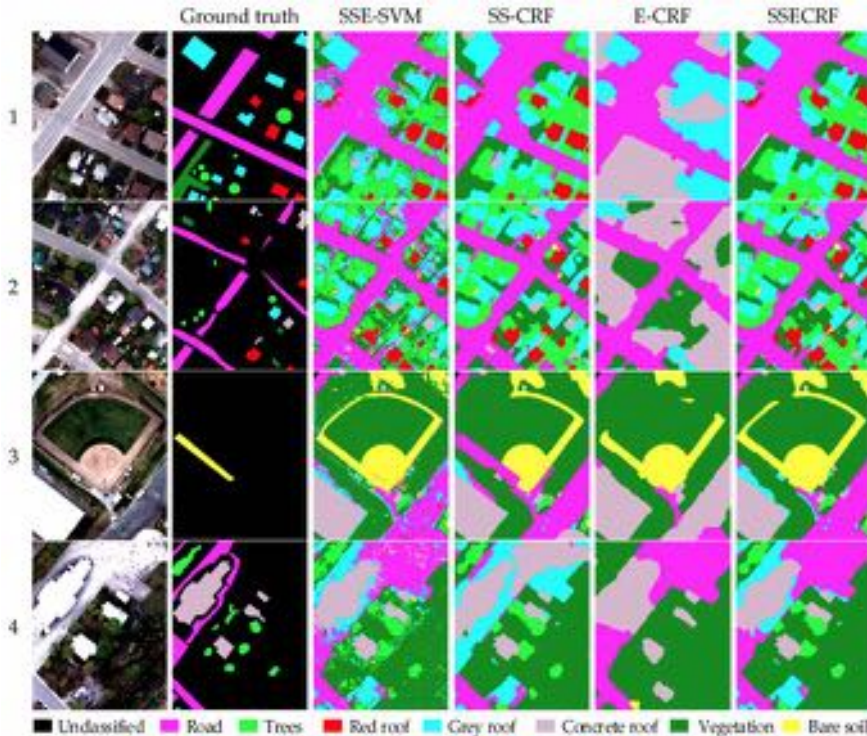
1980's

1990's

2000's

2010's

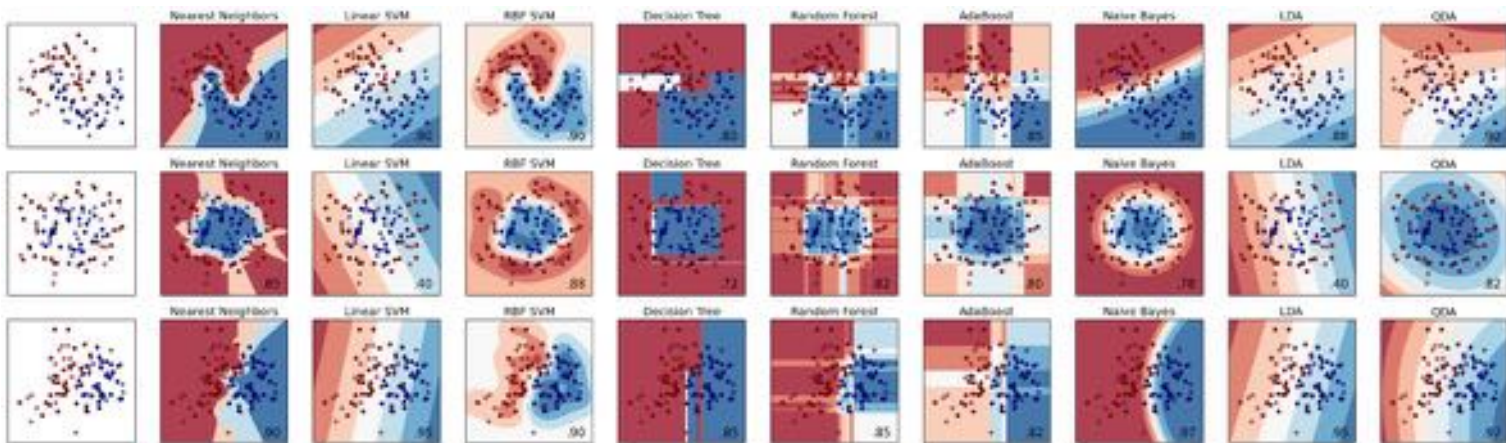
Remote sensing image classification



- **Goal:** “generate spatially explicit maps from remote sensing observations”
- **Machine learning**
 - Automates the process
 - Fast
 - Accurate
- **Many different classifiers**
 - SVM, RF, Boosting, Neural nets

Machine learning classifiers

- Computational cost?
- Accuracy?
- Easiness?
- Robustness to dimensionality?

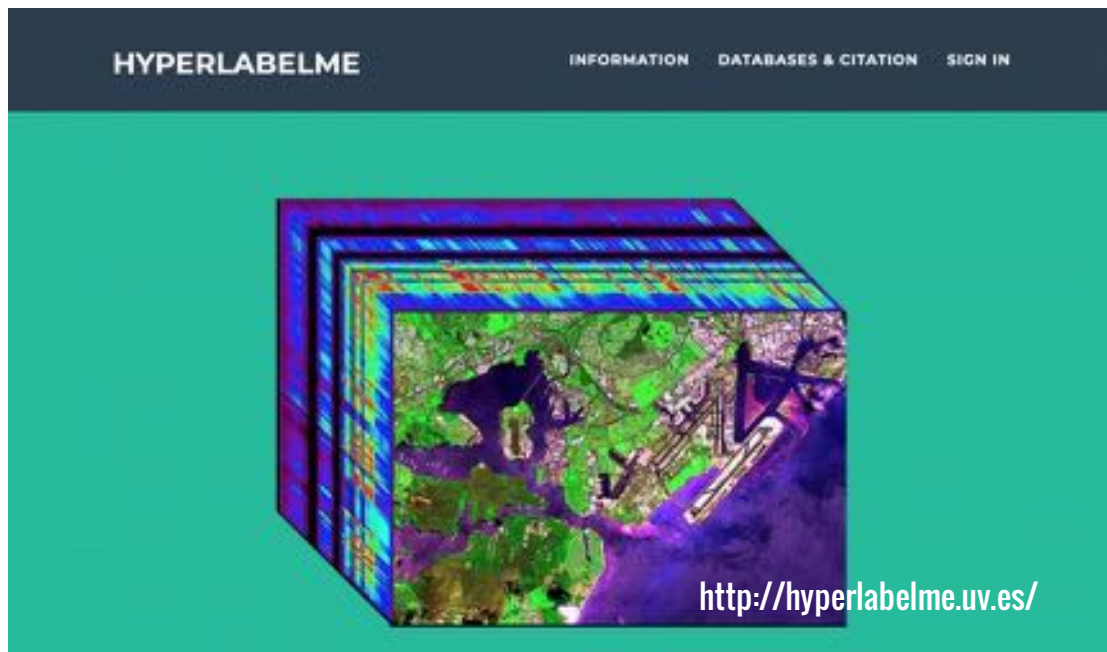


Hyperlabelme

<http://hyperlabelme.uv.es/>

Hyperlabelme:

- A platform to evaluate machine learning classifiers
- Harmonized large set of labeled multispectral and hyperspectral images
 - Number of classes
 - Dimensionality
 - Noise sources and levels
- Scalable and modular



Download image datasets

- HyperLabelMe runs a FAIR-use data policy
 - Data freely available to the public and scientific community
 - Download spectra and corresponding labels
 - Run your algorithms offline
 - Upload results
 - Get your scores

ANA RUESCAS DOWNLOAD TRAIN UPLOAD //HALL OF FAME LOGOUT

1. DOWNLOAD A DATASET

—★—

The datasets are in plain text. They start with a header containing some info (metadata), then the data with samples in rows and columns separated by commas. The first column is the class label, the rest the spectra. The first rows contain training samples, while the last rows, whose label is set to '-1' (meaning unknown) are the test samples your method will have to predict. We provide examples in MATLAB, Python and R of how to read the datasets.



Matlab

R

Python

Datasets are in plain text

DATABASES



Id	Acronym	Sensor	Rows	Columns	Bands	Classes	Download
1	Barrax99	Hymap	670	700	128	6	Im01
2	Botswana	EO-1	1476	256	145	14	Im02
3	KSC	AVIRIS	512	614	176	13	Im03
4	FlightLineC1	M7scanner	949	220	12	10	Im04
5	Indian Pines	AVIRIS	145	145	220	16	Im05
6	TipJut1	Landsat	169	169	7	5	Im06
7	Pavia	DAIS7915	400	400	40	9	Im07
8	Salinas	AVIRIS	217	512	224	16	Im08
9	Fahac	NaN	1094	2357	4	5	Im09

Datasets are in plain text

DATABASES



	A	B	C	D	E	F	G	H	I	J	K	L
1	name	img1										
2	name	img2										
3	name	img3										
4	name	img4										
5	name	img5										
6	2	1318	401	246	345	367	408	480	444	750	840	799
7	2	889	586	437	488	409	339	609	786	512	984	964
8	5	817	454	398	309	303	329	440	712	881	871	486
9	6	2247	1250	2432	1438	1558	3046	1833	2000	2212	2543	2827
10	1	947	179	627	642	690	769	858	997	1148	1281	1420
11	6	718	365	317	341	352	395	463	623	726	743	685
12	5	2447	425	313	172	283	290	400	604	795	754	612
13	6	942	200	367	145	104	176	200	286	340	314	312
14	4	717	150	186	150	109	168	156	168	340	367	345
15	6	1263	342	403	436	480	734	808	899	1026	1181	1312
16	1	802	868	867	864	812	1025	1099	1128	1177	1260	1294
17	4	841	208	105	113	108	171	208	272	308	343	306
18	2	1312	145	436	414	409	479	545	743	878	935	908
19	2	906	413	379	343	393	436	501	641	757	810	817
20	2	2214	150	463	311	371	361	481	600	1000	1242	1268
21	2	814	355	198	190	347	386	474	608	714	826	711
22	5	1025	341	282	170	274	298	402	609	807	809	642
23	1	1171	415	254	705	809	941	1042	1194	1476	1617	1680
24	4	810	305	232	220	280	280	280	378	449	462	410
25	5	1265	175	352	290	278	310	412	481	649	844	884
26	1	1191	800	806	838	858	942	1040	1178	1354	1526	1675
27	4	863	365	262	295	269	266	318	417	487	517	478
28	4	1032	216	124	111	120	137	172	249	300	336	383
29	4	728	306	190	143	169	187	243	343	420	441	405
30	4	814	479	308	122	129	265	400	575	840	873	610
31	1	790	765	777	794	806	824	1004	1108	1281	1417	1640
32	4	1187	454	407	388	412	432	524	670	770	808	776
33	8	1128	389	385	411	472	555	626	744	811	901	1005

Id	Acronym	Sensor	Rows	Columns	Bands	Classes	Download
1	Barrax99	Hymap	670	700	128	6	Im01
2	Botswana	EO-1	1476	256	145	14	Im02
3	KSC	AVIRIS	512	614	176	13	Im03
4	FlightLineC1	M7scanner	949	220	12	10	Im04
5	Indian Pines	AVIRIS	145	145	220	16	Im05
6	TipuJ1	Landsat	169	169	7	5	Im06
7	Pavia	DAIS7915	400	400	40	9	Im07
8	Salinas	AVIRIS	217	512	224	16	Im08
9	Fahac	NaN	1094	2357	4	5	Im09

Easy to read

DATABASES



Id	Acronym	Sensor	Rows	Columns	Bands	Classes	Download
1	Barrax99	Hymap	670	700	128	6	lm01
2	Botswana	EO-1	1476	256	145	14	lm02
3	KSC	AVIRIS	512	614	176	13	lm03
4	FlightLineC1	M7scanner	949	220	12	10	lm04
5	Indian Pines	AVIRIS	145	145	220	16	lm05
6	TipJul1	Landsat	169	169	7	5	lm06
7	Pavia	DAIS7915	400	400	40	9	lm07
8	Salinas	AVIRIS	217	512	224	16	lm08
9	Fahac	NaN	1094	2357	4	5	lm09

```
import numpy as np

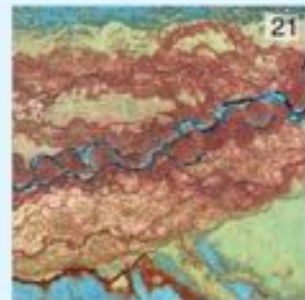
def dataread(filename):
    lasttag = 'description:'
    # Open file and locate lasttag
    f = open(filename, 'r')
    nl = 1
    for line in f:
        if line.startswith(lasttag): break
        nl += 1
    f.close()

    # Read data
    data = sp.loadtxt(filename, delimiter=',', skiprows=nl)
    Y = data[:, 0]
    X = data[:, 1:]
    # Separate train/test
    Xtest = X[Y < 0, :]
    X = X[Y >= 0, :]
    Y = Y[Y >= 0, None]

    return X, Y, Xtest

def datawrite(method, dataset, Yp):
    filename = '{0}_{1}_predictions.txt'.format(method, dataset)
    res = True
    try:
        with open(filename, mode='w') as f:
            f.write('{0} {1}'.format(method, dataset))
            for v in Yp:
                f.write(' {0}'.format(str(v)))
                f.write('\n')
    except Exception as e:
        print('Error', e)
        res = False
    return res
```

We provide examples in MATLAB, Python and R of how to read the datasets.



Train your models offline and upload your predictions

```
1 function [X, Y, Xtest] = dataRead(filename)
2
3 lasttag = 'description: ';
4
5 % Open file and locate lasttag
6 f = fopen(filename, 'r');
7 n1 = 1;
8 while ~feof(f)
9     line = fgetl(f);
10    if strcmp(line, lasttag, length(lasttag))
11        break
12    end
13    n1 = n1 + 1;
14 end
15 fclose(f);
16
17 % Read data
18 data = dlmread(filename, ',', n1, 0);
19 Y = data(:,1);
20 X = data(:,2:end);
21
22 Xtest = X(Y == 0, :);
23 Y = X(Y == 0, 1);
24 Y = Y(Y == 0, 1);
25
```

Read

```
1 clear, clc
2
3 % Read dataset
4 [X, Y, Xtest] = dataRead('Im01.txt');
5 % Use a simple linear classifier
6 Yp = classify(Xtest, X, Y);
7 % Write results
8 dataWrite('LOA_MATLAB', 'Im01', Yp);
9
```

Train

```
1 function dataWrite(method, dataset, Yp)
2
3 % Generate test file for validation on hyperlabelme server
4 % Inputs:
5 % method: name of the method
6 % dataset: name of the dataset
7 % Yp: predictions as a column vector
8
9 fname = sprintf('%s_%s_predictions.txt', method, dataset);
10
11 f = fopen(fname, 'w');
12 fprintf(f, '%s %s', method, dataset);
13 fprintf(f, ' %0', Yp);
14 fprintf(f, '\n');
15 fclose(f);
16
```

Save

Train your models offline and upload your predictions

CONFUSION MATRIX

Ground truth classes

	1	2	3	4	5	6	
1	816	3	1	2	5	7	834
2	0	804	10	2	5	3	824
3	3	18	785	5	6	19	836
4	0	0	9	817	0	6	832
5	1	0	5	0	805	7	818
6	10	5	20	4	9	788	836
Sum	830	830	830	830	830	830	

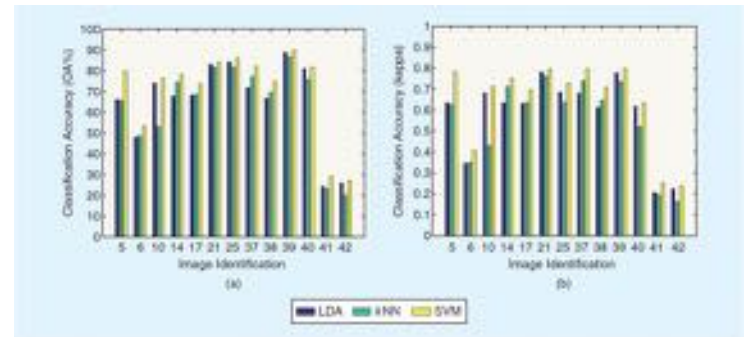
- Overall accuracy: 96.69 %
- Kappa coefficient: 0.96, CI: ± 0.005965 , z-score: 315.500146

Class	1	2	3	4	5	6
User's acc:	97.84	97.57	93.90	98.20	98.41	94.26
Prod's acc:	98.31	96.87	94.58	98.43	96.99	94.94

HALL OF FAME

Select dataset: 27-Barra (MELB)

Rank	User	Dataset	Method	Overall Accuracy (%)	Kappa Coefficient
1	igordian	in27	SVM	96.78	0.94
2	AnjanJang	in27	Rf_100	96.28	0.93
3	Luis	in27	KNN	94.56	0.89
4	Luis	in27	LDA	88.34	0.73

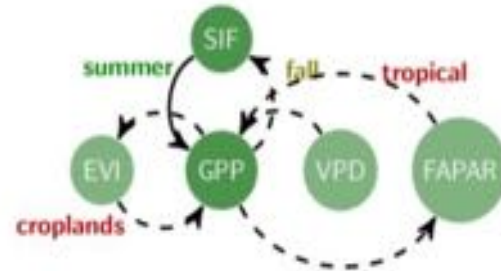
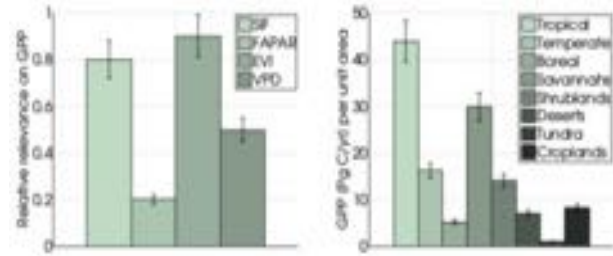
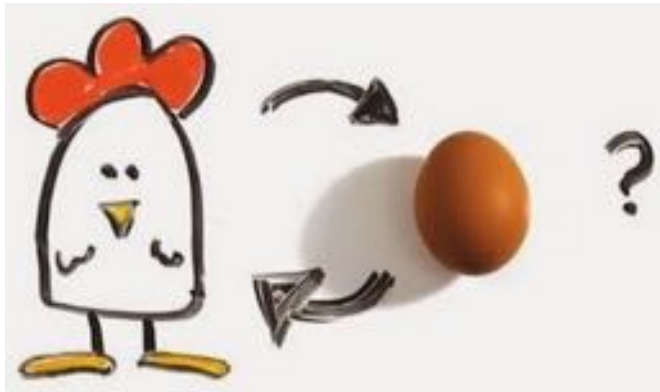


Causeme: understanding is harder than predicting!

<http://causeme.uv.es/>

Extension to causal discovery

- Learn what's the cause and the effect from time series of variables
- Important implications in Earth science and climate



"Inferring causation from time series with perspectives in Earth system sciences"

Runge, Bathiany, Bollt, Camps-Valls, et al. Nat Comm (submitted), 2018.

"Causal Inference in Geoscience and Remote Sensing from Observational Data,"

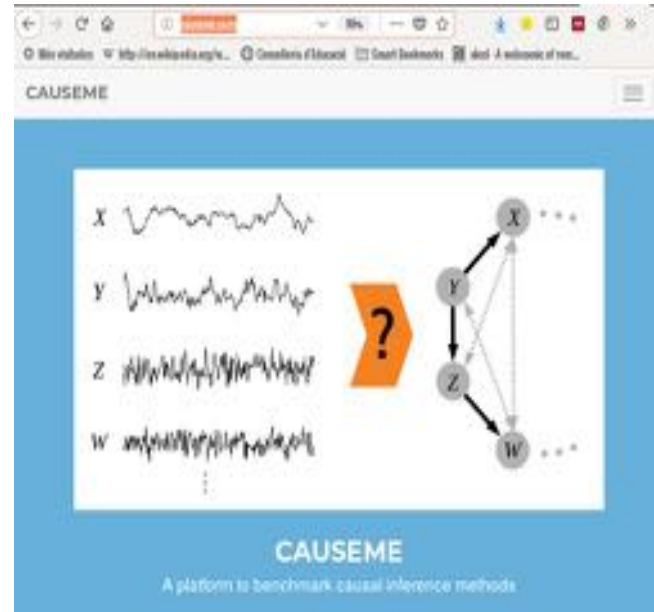
Pérez-Suay and Camps-Valls, IEEE Trans. Geosc. Rem. Sens, 2018

Extension to causal discovery

- **CauseMe: <http://causeme.uv.es>**
 - Download time series with ground truth
 - Run your causal discovery algorithm offline
 - Upload your causal graph
 - Get your results!

“Inferring causation from time series with perspectives in Earth system sciences”
Runge, Bathiany, Bollt, Camps-Valls, et al. Nat Comm (submitted), 2018.
“Causal Inference in Geoscience and Remote Sensing from Observational Data,”
Pérez-Suay and Camps-Valls, IEEE Trans. Geosc. Rem. Sens, 2018

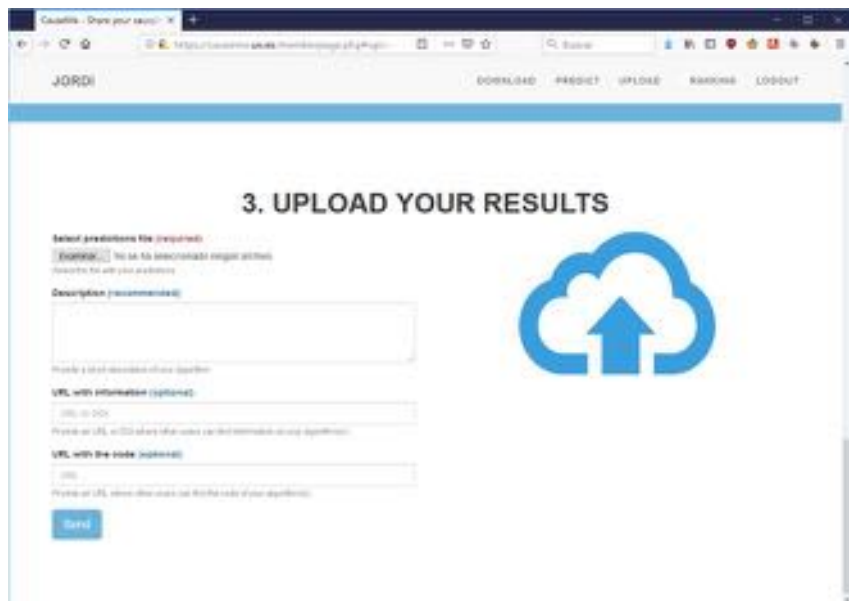
CauseMe: <http://causeme.uv.es>



Extension to causal discovery

- **CauseMe: <http://causeme.uv.es>**
 - Download time series with ground truth
 - Run your causal discovery algorithm offline
 - Upload your causal graph
 - Get your results!

“Inferring causation from time series with perspectives in Earth system sciences”
Runge, Bathiany, Bollt, Camps-Valls, et al. Nat Comm (submitted), 2018.
“Causal Inference in Geoscience and Remote Sensing from Observational Data,”
Pérez-Suay and Camps-Valls, IEEE Trans. Geosc. Rem. Sens, 2018



Extension to causal discovery

— — —

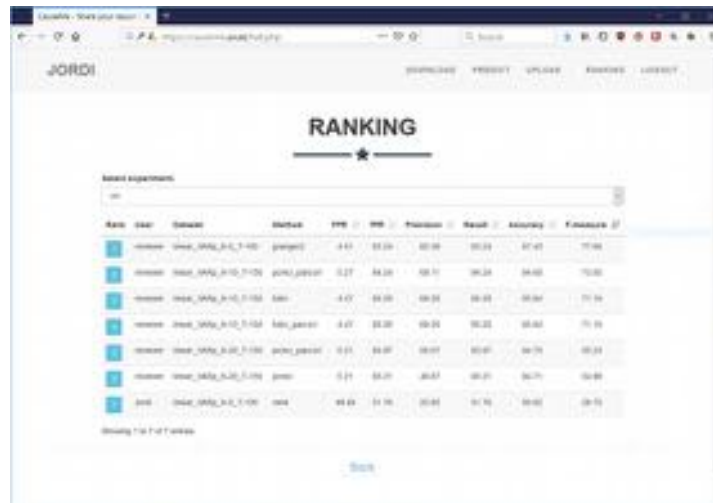
- **CauseMe: <http://causeme.uv.es>**
 - Download time series with ground truth
 - Run your causal discovery algorithm offline
 - Upload your causal graph
 - Get your results!



“Inferring causation from time series with perspectives in Earth system sciences”
Runge, Bathiany, Bollt, Camps-Valls, et al. Nat Comm (submitted), 2018.
“Causal Inference in Geoscience and Remote Sensing from Observational Data,”
Pérez-Suay and Camps-Valls, IEEE Trans. Geosc. Rem. Sens, 2018

Extension to causal discovery

- **CauseMe: <http://causeme.uv.es>**
 - Download time series with ground truth
 - Run your causal discovery algorithm offline
 - Upload your causal graph
 - Get your results!



The screenshot shows the 'RANKING' page of the CauseMe web application. The page features a search bar for experiments and a table with the following columns: Rank, ID, Dataset, Status, F1F, F1R, Precision, Recall, Accuracy, and Knowledge. The table contains six rows of data, each representing an experiment with its corresponding performance metrics.

Rank	ID	Dataset	Status	F1F	F1R	Precision	Recall	Accuracy	Knowledge
1	causal_0001_010_0100	germany	Completed	0.81	0.81	0.81	0.81	0.81	100%
2	causal_0002_010_0100	germany	Completed	0.77	0.77	0.77	0.77	0.77	100%
3	causal_0003_010_0100	germany	Completed	0.77	0.77	0.77	0.77	0.77	100%
4	causal_0004_010_0100	germany	Completed	0.77	0.77	0.77	0.77	0.77	100%
5	causal_0005_010_0100	germany	Completed	0.77	0.77	0.77	0.77	0.77	100%
6	causal_0006_010_0100	germany	Completed	0.77	0.77	0.77	0.77	0.77	100%

“Inferring causation from time series with perspectives in Earth system sciences”

Runge, Bathiany, Bollt, Camps-Valls, et al. Nat Comm (submitted), 2018.

“Causal Inference in Geoscience and Remote Sensing from Observational Data,”

Pérez-Suay and Camps-Valls, IEEE Trans. Geosc. Rem. Sens, 2018

Conclusions



Conclusions

— — —

- **A fact:** Remote sensing needs machine learning
- **A question:** What's the best classifier for my specific task?
- **An observation:** Often the simpler is the better
- **Our aim:** Hyperlabelme is a simple web platform to benchmark classifiers
- **Working towards extensions:**
 - Spatial context
 - Spatio-temporal classifiers
- **Some relatives:**
 - Causal discovery in time series
 - Regression (parameter retrieval) algorithms
- **Please contribute!**

<http://hyperlabelme.uv.es>

<http://causeme.uv.es>

<http://regressme.uv.es>