Scientific Big Data Analytics by HPC
-
Parallel and Scalable Machine Learning on JURECA

Dr.- Ing. Morris Riedel et al.

Head of Research Group, Juelich Supercomputing Centre

Adjunct Associated Professor, University of Iceland

JURECA Porting and Tuning Workshop

6th - 8th June 2016, Jülich Supercomputing Centre

HELMHOLTZ ASSOCIATION

Federated Systems and Data Division

Research Group

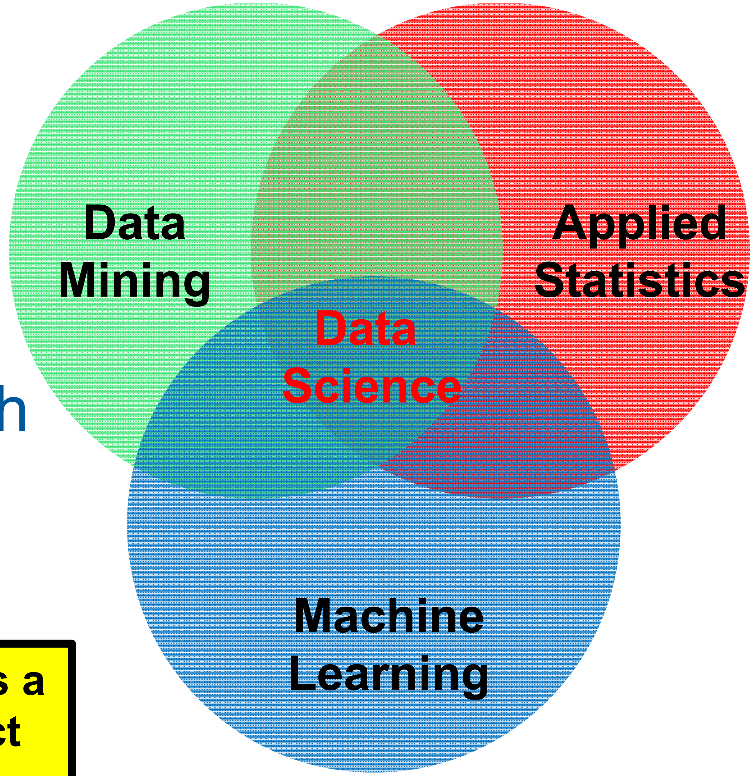
High Productivity Data Processing

JÜLICH FORSCHUNGSZENTRUM

UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

Learning from Data – Different to Simulation Science

1. Some pattern exists
2. No exact mathematical formula
3. Data exists



Idea 'Learning from Data' shared with a wide variety of other disciplines

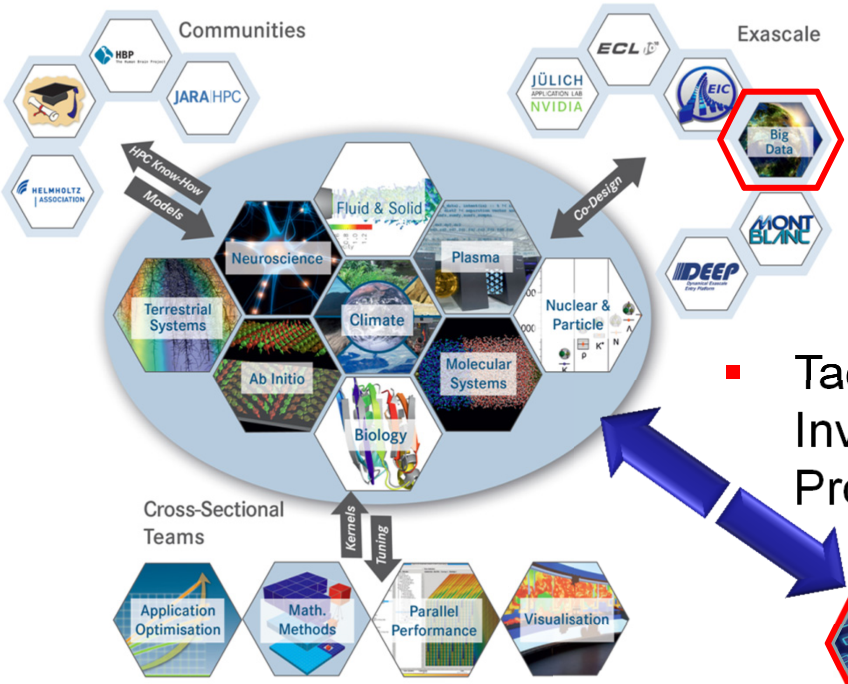
- E.g. signal processing, etc.

> - **Statistical data mining and machine learning is a very broad subject and goes from very abstract theory to extreme practice ('rules of thumb')**

Using HPC resources like JURECA useful

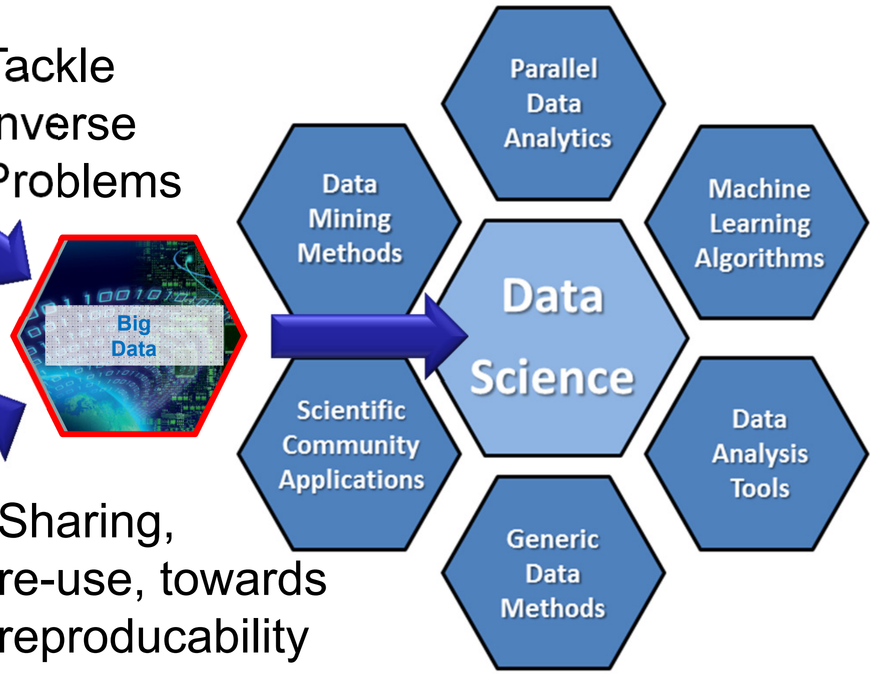
- Reasoning: parallel I/O, mature inter-process communication (MPI), OpenMP, GPGPUs, etc.

Context Juelich Supercomputing Centre



- Research data-intensive science and engineering applications
- Explore computing that is more intertwined with data analysis

- Tackle Inverse Problems



Federated Data Management, Preservation, Security & Access

- Sharing, re-use, towards reproducability

[8] Th. Lippert, D. Mallmann, M. Riedel, 'Scientific Big Data Analytics by HPC', Publication Series of the John von Neumann Institute for Computing (NIC) NIC Series 48, 417, ISBN 978-3-95806-109-5, pp. 1 - 10, 2016

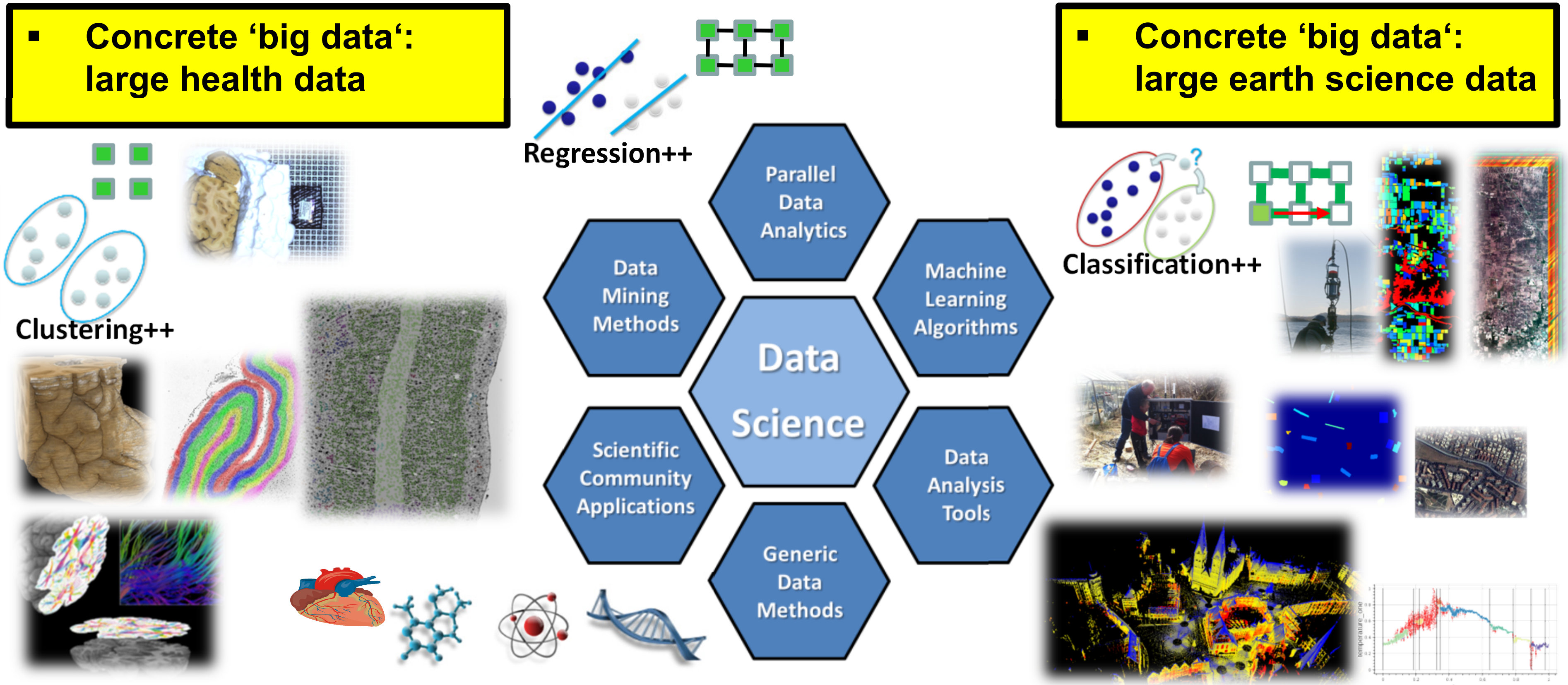
Parallelization Demand

Serial data analysis techniques/tools increasingly show limits

- Traditional methods still relevant, but need to scale for 'big data'
- Big Data: e.g. high number of dimensions/classes or 'data points'

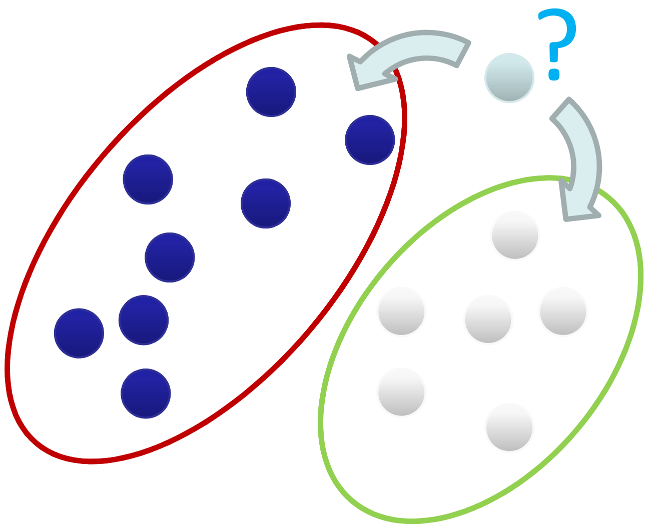
- **Concrete 'big data': large health data**

- **Concrete 'big data': large earth science data**



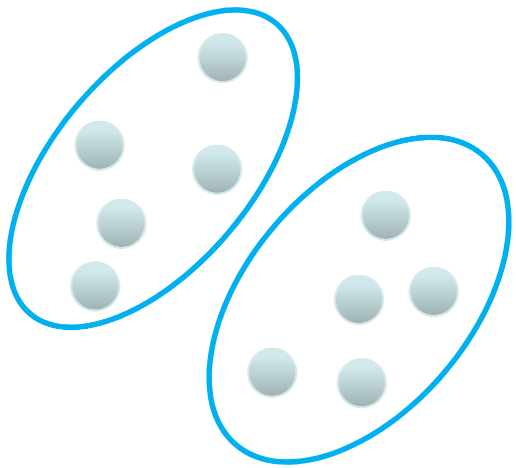
Clustering Technique

Classification



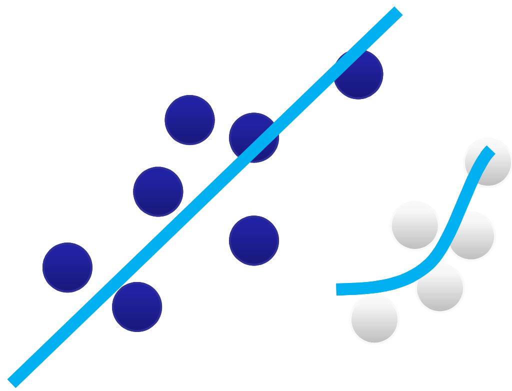
- Groups of data exist
- New data classified to existing groups

Clustering



- No groups of data exist
- Create groups from data close to each other

Regression



- Identify a line with a certain slope describing the data

Selected Clustering Methods

K-Means Clustering – Centroid based clustering

- Partitions a data set into K distinct clusters (centroids can be artificial)

K-Medoids Clustering – Centroid based clustering (variation)

- Partitions a data set into K distinct clusters (centroids are actual points)

Sequential Agglomerative hierarchic nonoverlapping (SAHN)

- Hiearchical Clustering (create tree-like data structure → 'dendrogram')

Clustering Using Representatives (CURE)

- Select representative points / cluster; as far from one another as possible

Density-based spatial clustering of applications + noise (DBSCAN)

Reasoning: density similiarity measure helpful in our driving applications

- Assumes clusters of similar density or areas of higher density in dataset

Technology Review of Available ‘Big Data ‘Tools

- **JSC courses 'parallel programming‘ useful: Introduction to parallel programming with MPI and OpenMP, Advanced parallel programming with MPI and OpenMP**

Technology	Platform Approach	Analysis
HPDBSCAN (authors implementation)	C; MPI; OpenMP	Parallel, hybrid, DBSCAN
Apache Mahout	Java; Hadoop	K-means variants, spectral, no DBSCAN
Apache Spark/MLlib	Java; Spark	Only k-means clustering, No DBSCAN
scikit-learn	Python	No parallelization strategy for DBSCAN
Northwestern University PDSDBSCAN-D	C++; MPI; OpenMP	Parallel DBSCAN

[1] M. Goetz, M. Riedel et al., **‘On Parallel and Scalable Classification and Clustering Techniques for Earth Science Datasets’,** *6th Workshop on Data Mining in Earth System Science, International Conference of Computational Science, 2015*

DBSCAN

DBSCAN Algorithm

[2] Ester et al.

Unclustered Data

- Introduced 1996 by Martin Ester et al.
- Groups number of similar points into clusters of data
- Similarity is defined by a distance measure (e.g. *euclidean distance*)

Distinct Algorithm Features

- Clusters a variable number of clusters
- Forms arbitrarily shaped clusters
- Identifies outliers/noise

Clustered Data

Understanding Parameters for MPI/OpenMP tool

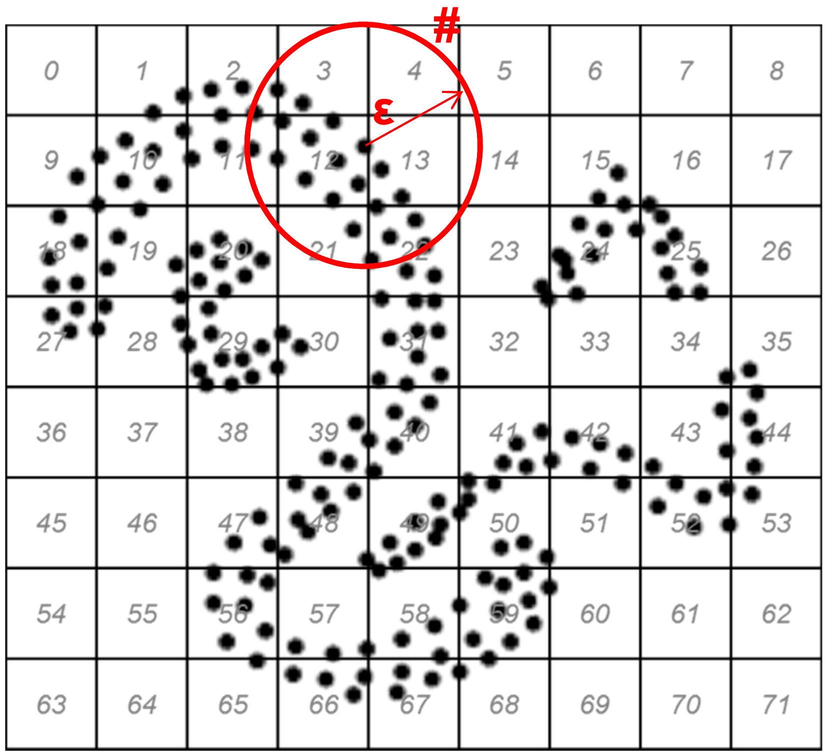
- Looks for a similar points within a given search radius
  → Parameter *epsilon*
- A cluster consist of a given minimum number of points
  → Parameter *minPoints*

[3] M.Goetz & C. Bodenstein, HPDBSCAN Tool

Parallel & Scalable HP-DBSCAN Tool on JURECA (1)

Parallelization Strategy

- Smart ‘Big Data‘ Preprocessing into Spatial Cells (‘indexed‘)
- OpenMP standalone
- MPI (+ optional OpenMP hybrid)



[4] M.Goetz, M. Riedel et al., ‘HPDBSCAN – Highly Parallel DBSCAN’, MLHPC Workshop at Supercomputing 2015

Preprocessing Step

- Spatial indexing and redistribution according to the point localities
- Data density based chunking of computations

Computational Optimizations

- Caching of point neighborhood searches
- Cluster merging based on comparisons instead of zone reclustering

Parallel & Scalable HP-DBSCAN Tool on JURECA (2)

Usage via jobscript

- Using job scheduler
- Important: module load hdf5/1.8.13
- Important: library gcc-4.9.2/lib64
- np = number of processors
- t = number of threads
- Uses parallel/IO

JURECA @ Juelich

DBSCAN
Parameters

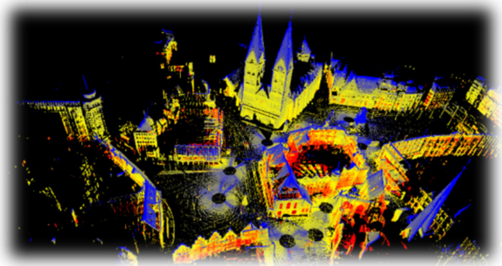
```
module load hdf5/1.8.13
export LD_LIBRARY_PATH=/homeb/zam/analytic/bigdata/hpdbscan/gcc-4.9.2/lib64:$LD_LIBRARY_PATH
DBSCAN=/homeb/zam/analytic/bigdata/hpdbscan/jsc_mpi/dbscan
SMALLBREMENDATA=/homeb/zam/analytic/bigdata/hpdbscan/jsc_mpi/mriruns/bremenSmall.h5

cd /homeb/zam/analytic/bigdata/hpdbscan/jsc_mpi/mriruns
mpiexec -np 1 $DBSCAN -e 300 -m 100 -t 12 $SMALLBREMENDATA
```

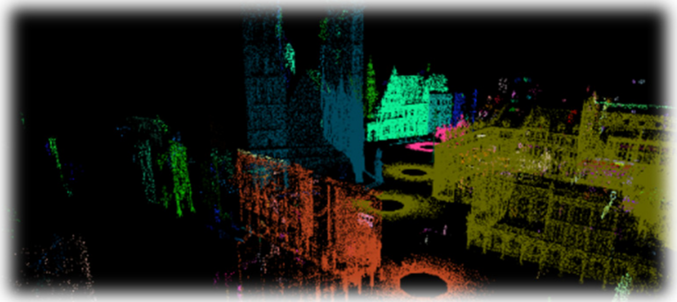
➢ **JSC courses ‘Parallel I/O‘ useful: Parallel I/O and portable data formats**

Clustering Applications – Large Point Clouds

‘Big Data‘: 3D/4D laser scans

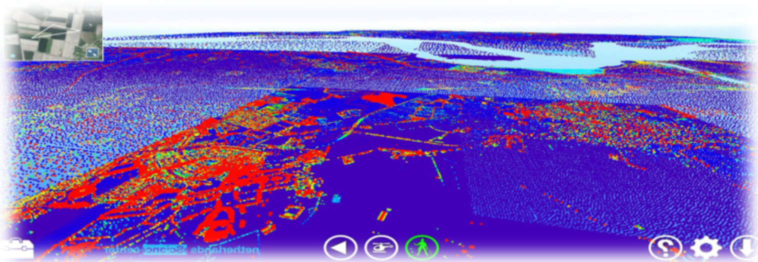
- Captured by robots or drones
- Millions to billion entries
- Inner cities (e.g. Bremen inner city)
- Whole countries (e.g. Netherlands)

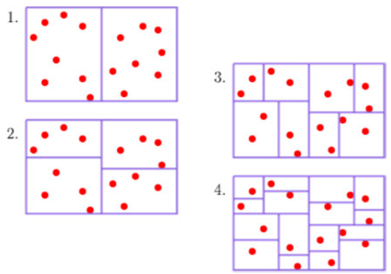


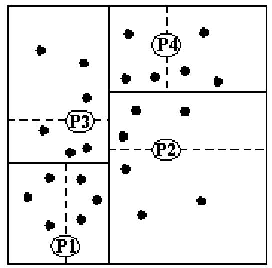


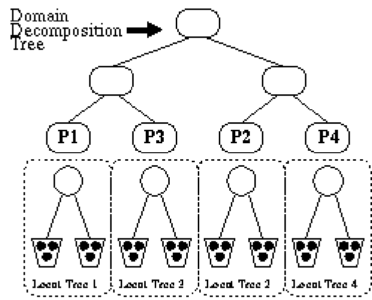
Selected Scientific Cases

- Filter noise to better represent real data
- Grouping of objects (e.g. buildings)
- Different level of details (e.g. trees)







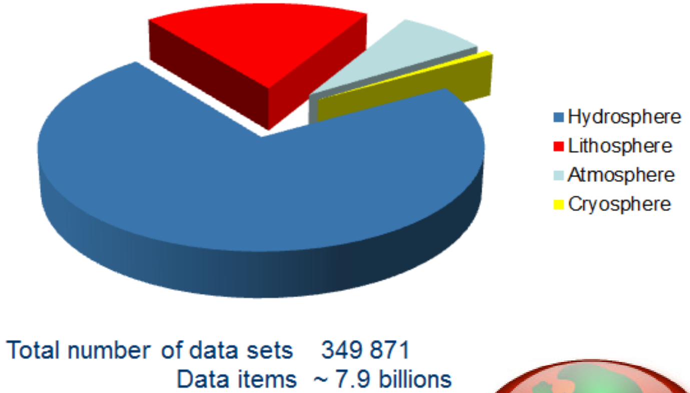


➢ **Research activities in collaboration with the Netherlands e-Science Centre & TU Delft**

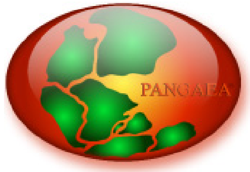
Clustering Applications – Many Time Series & Events

Earth Science Data Repository

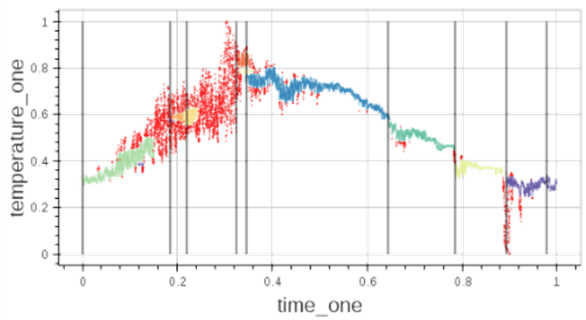
- Time series measurements (e.g. salinity)
- Millions to billions of data items/locations
- Less capacity of experts to analyse data



Selected Scientific Case

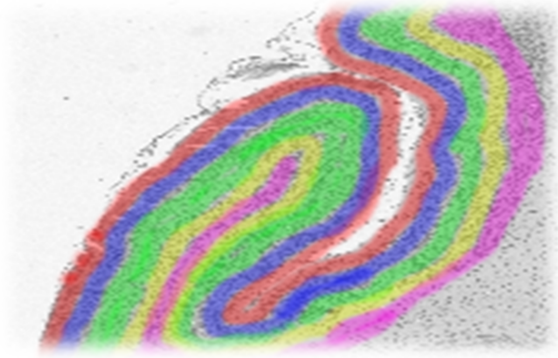


- Data from Koljöfjords in Sweden (Skagerrak)
- Each measurement small data, but whole sets are 'big data'
- Automated water mixing event detection & quality control (e.g. biofouling)
- Verification through domain experts



> ➢ **Research activities in collaboration with MARUM in Bremen and University of Gothenburg**

Clustering Applications – Neuro Science Image Analysis

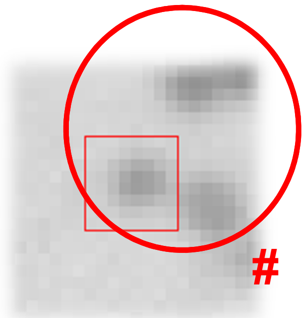


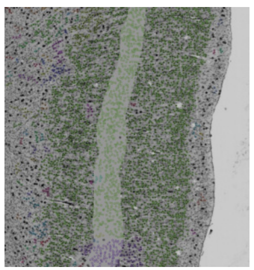
Large Brain Images

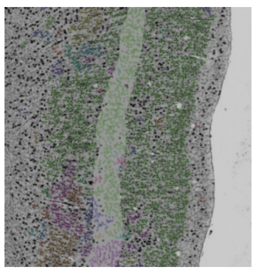
- High resolution scans of post mortem brains
- Rare ‘groundtruth available‘

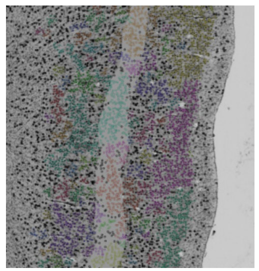
Selected Scientific Case

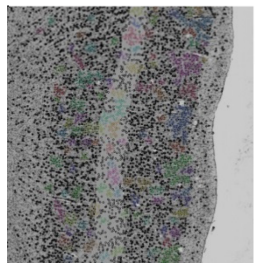
- Cell nuclei detection and tissue clustering
- Detect various layers (colored)
- Layers seem to have different density distribution of cells
- Extract cell nuclei into 2D/3D point cloud
- Cluster different brain areas by cell density







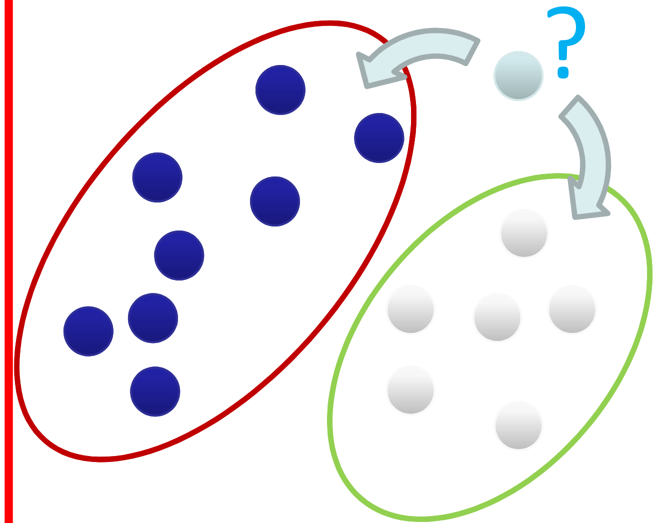




➢ **Research activities in collaboration with Institute of Medicine and Neuroscience (T. Dickscheid)**

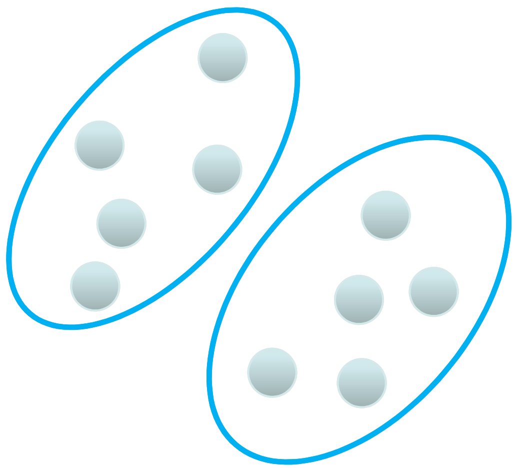
Classification Technique

Classification



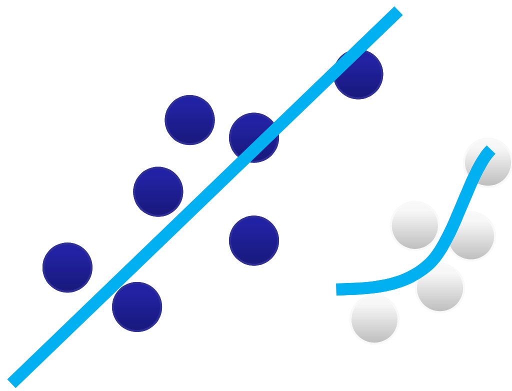
- Groups of data exist
- New data classified to existing groups

Clustering



- No groups of data exist
- Create groups from data close to each other

Regression



- Identify a line with a certain slope describing the data

Selected Classification Methods

Perceptron Learning Algorithm – simple linear classification

- Enables binary classification with 'a line' between classes of seperable data

Support Vector Machines (SVMs) – non-linear ('kernel') classification

- Enables non-linear classification with maximum margin (best 'out-of-the-box')

Reasoning: achieves often better results than other methods in tackled application domain

Decision Trees & Ensemble Methods – tree-based classification

- Grows trees for class decisions, ensemble methods average n trees

Artificial Neural Networks (ANNs) – brain-inspired classification

- Combine multiple linear perceptrons to a strong network for non-linear tasks

Naive Bayes Classifier – probabilistic classification

- Use of the Bayes theorem with strong/naive independence between features

Technology Review of Available ‘Big Data ‘Tools

- **JSC courses 'GPU programming‘ useful: Vectorisation and portable programming using OpenCL, GPU programming with OpenACC, GPU programming with CUDA**

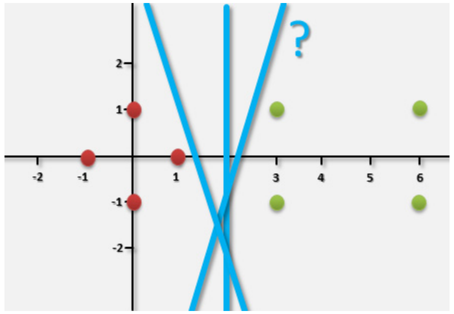
Technology	Platform Approach	Analysis
Apache Mahout	Java; Hadoop	No parallelization strategy for SVMs
Apache Spark/MLlib	Java; Spark	Parallel linear SVMs (no multi-class)
Twister/ParallelSVM	Java; Twister; Hadoop 1.0	Parallel SVMs, open source; developer version 0.9 beta
scikit-learn	Python	No parallelization strategy for SVMs
piSVM 1.2 & piSVM 1.3	C; MPI	Parallel SVMs; stable; not fully scalable
GPU LibSVM	CUDA	Parallel SVMs; hard to programs, early versions
pSVM	C; MPI	Parallel SVMs; unstable; beta version

[1] M. Goetz, M. Riedel et al., ‘*On Parallel and Scalable Classification and Clustering Techniques for Earth Science Datasets’, *6th Workshop on Data Mining in Earth System Science, International Conference of Computational Science

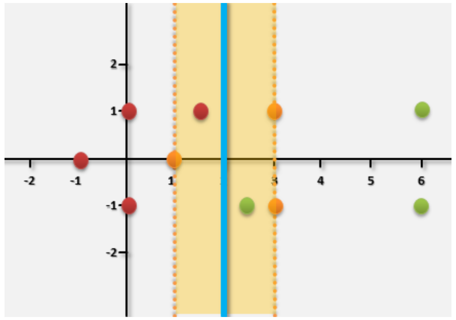
SVM Algorithm

[5] C. Cortes and V. Vapnik et al.

- Introduced 1995 by C.Cortes & V. Vapnik et al.
- Creates a ‘maximal margin classifier‘ to get future points (‘more often‘) right and take advantage of kernel methods
- Uses quadratic programming & Lagrangian method with N x N



(linear example)



(‘maximal margin clasifier‘ example)

(use of soft-margin approach for better generalization)

$$\min_{w,\xi_i,b}\left\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i\right\}$$

$$y_i(\mathbf{w}\cdot\mathbf{x_i} - b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

(maximizing hyperplane turned into optimization problem, minimization, allow some errors)

$$\mathcal{L}(\alpha) = \sum_{n=1}^{N}\alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \boxed{\mathbf{x}_n^T\mathbf{x}_m}$$

$$0 \leq \alpha_i \leq C$$

$$\boxed{k(\mathbf{x}_i,\mathbf{x}_j)} = \langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle$$

$$\begin{bmatrix} y_1y_1x_1^Tx_1 & y_1y_2x_1^Tx_2 & \ldots y_1y_Nx_1^Tx_N \\ \ldots & \ldots & \ddots \\ y_Ny_1x_N^Tx_1 & y_Ny_2x_N^Tx_2 & \ldots y_Ny_Nx_N^Tx_N \end{bmatrix}$$

(max. hyperplane → dual problem, using quadratic programming method, e.g. sequential minimal optimization)

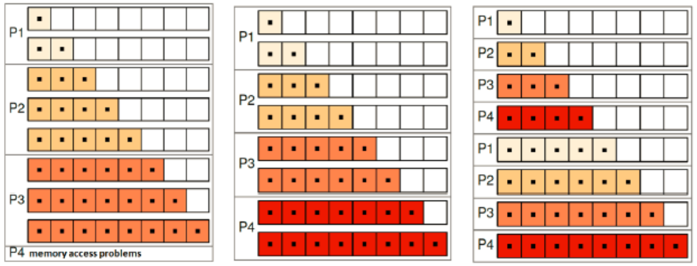
(kernel trick, quadratic coefficients – Computational Complexity & Big Data Impact)

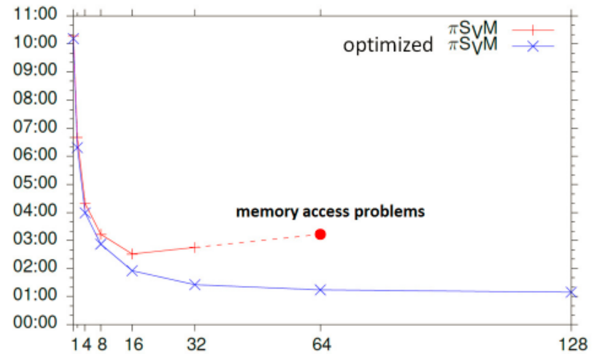
Parallel & Scalable piSVM Tool on JURECA (1)

Original parallel piSVM tool 1.2

- Open-source and based on libSVM library, C, 2011
- Message Passing Interface (MPI)
- New version appeared 2014-10 v. 1.3 (no major improvements)
- Lack of ‘big data‘ support (memory, layout, etc.)

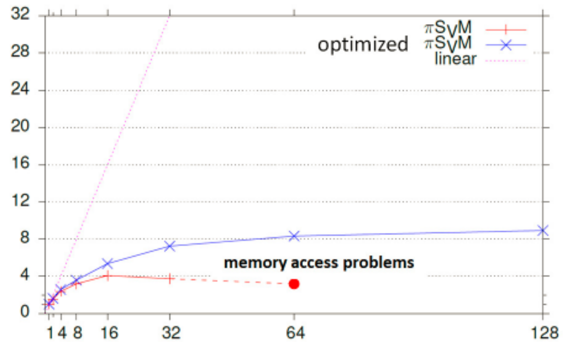
[6] piSVM Website, 2011/2014 code





Tuned scalable parallel piSVM tool 1.2.1

- Highly scalable version maintained by Juelich
- Based on original piSVM 1.2 tool
- Open-source (repository to be created)
- Optimizations: load balancing; MPI collectives



Parallel & Scalable piSVM Tool on JURECA (2)

Usage via jobscript

- Using job scheduler
- np = number of processors;
- o/q = problem partitioning
- c = cost (soft margin SVM)
- g = RBF kernel parameter
- T = type of SVM (here C-SVC)
- Example: train phase submit

JURECA @ Juelich

```
### location
PISVM=/homeb/zam/mriedel/pisvm-1.2/pisvm-1.2/pisvm-train

TRAINDATA=/homeb/zam/mriedel/bigdata/86-
romeok/sdap_area_all_training.el

### submit
mpiexec -np $NSLOTS $PISVM -o 1024 -q 512 -c 10000 -g 16 -t 2 -m
1024 -s 0 $TRAINDATA
```

SVM
Parameters

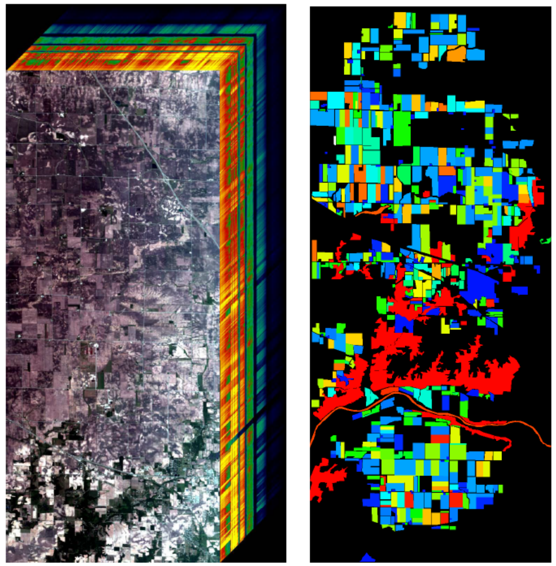
- **Submission of test phase similar but using labelled dataset + trained SVM model**

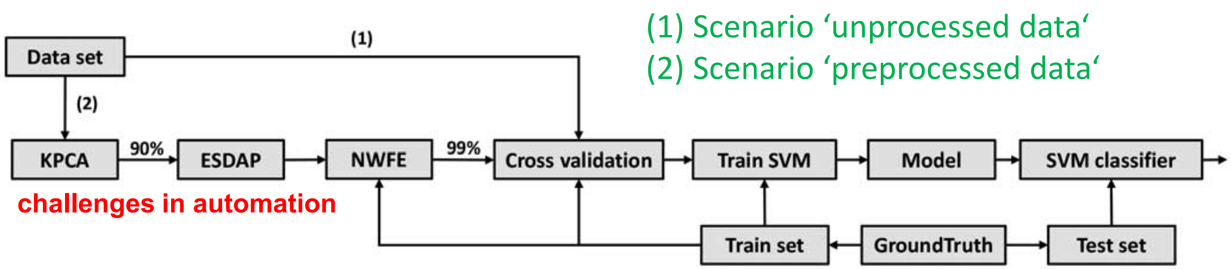
Classification Applications – Remote Sensing Images

Challenges: high number of classes, less samples, mixed pixels

Class		Number of samples		Class		Number of samples	
Number	Name	Training	Test	Number	Name	Training	Test
1	Buildings	1720	15 475	27	Pasture	1039	9347
2	Corn	1778	16 005	28	pond	10	92
3	Corn?	16	142	29	Soybeans	939	8452
4	Corn-EW	51	463	30	Soybeans?	89	805
5	Corn-NS	236	2120	31	Soybeans-NS	111	999
6	Corn-CleanTill	1240	11 164	32	Soybeans-CleanTill	507	4567
7	Corn-CleanTill-EW	2649	23 837	33	Soybeans-CleanTill?	273	2453
8	Corn-CleanTill-NS	3968	35 710	34	Soybeans-CleanTill-EW	1180	10 622
9	Corn-CleanTill-NS-Irrigated	80	720	35	Soybeans-CleanTill-NS	1039	9348
10	Corn-CleanTilled-NS?	173	1555	36	Soybeans-CleanTill-Drilled	224	2018
11	Corn-MinTill	105	944	37	Soybeans-CleanTill-Weedy	54	489
12	Corn-MinTill-EW	563	5066	38	Soybeans-Drilled	1512	13 606
13	Corn-MinTill-NS	886	7976	39	Soybeans-MinTill	267	2400
14	Corn-NoTill	438	3943	40	Soybeans-MinTill-EW	183	1649
15	Corn-NoTill-EW	121	1085	41	Soybeans-MinTill-Drilled	810	7288
16	Corn-NoTill-NS	569	5116	42	Soybeans-MinTill-NS	495	4458
17	Fescue	11	103	43	Soybeans-NoTill	216	1941
18	Grass	115	1032	44	Soybeans-NoTill-EW	253	2280
19	Grass/Trees	233	2098	45	Soybeans-NoTill-NS	93	836
20	Hay	113	1015	46	Soybeans-NoTill-Drilled	873	7858
21	Hay?	219	1966	47	Swampy Area	58	525
22	Hay-Alfalfa	226	2032	48	River	311	2799
23	Lake	22	202	49	Trees?	58	522
24	NotCropped	194	1746	50	Wheat	498	4481
25	Oats	174	1568	51	Woods	6356	57 206
26	Oats?	34	301	52	Woods?	14	130

remote sensing cube & ground reference

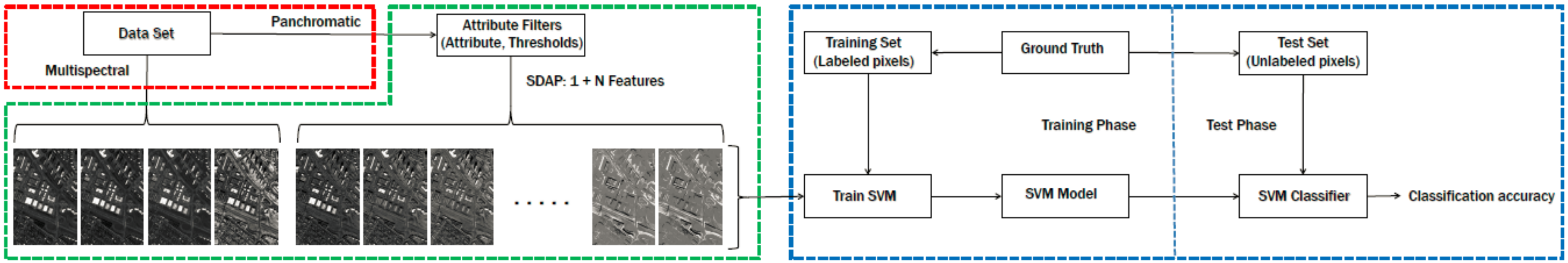




challenges in automation

[7] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

Classification Applications – SDAP Feature Extraction



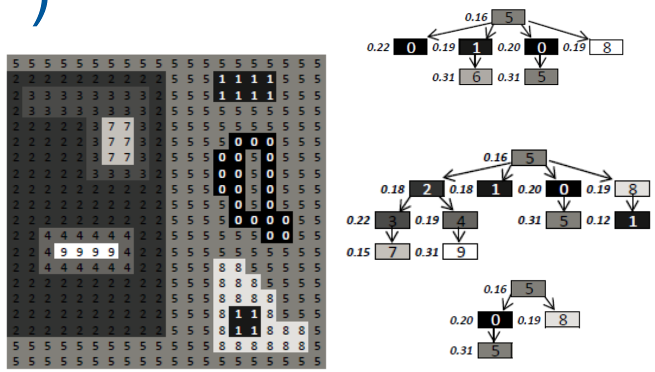
[7] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

Key importance

- Use feature extraction/enhancement
- Apply dimensionality reduction techniques (e.g. principle components)

Example: Self-Dual Attribute Profile (SDAP)

- Use different filtering strategies for morphological attributes as additional inputs
- Sequentially apply attribute filters on tree-based image representations

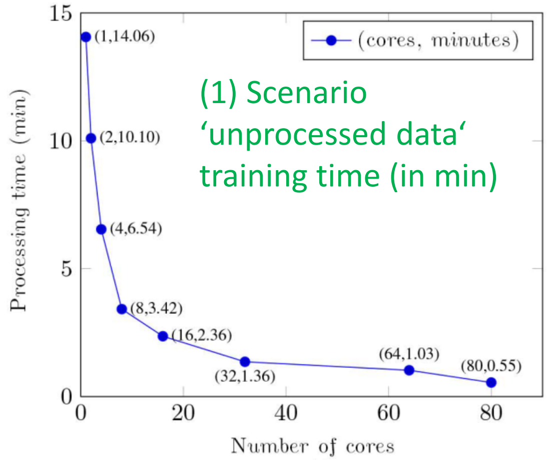


➢ **Research activities in collaboration with University of Iceland (G. Cavallaro, J.A. Benediktsson)**

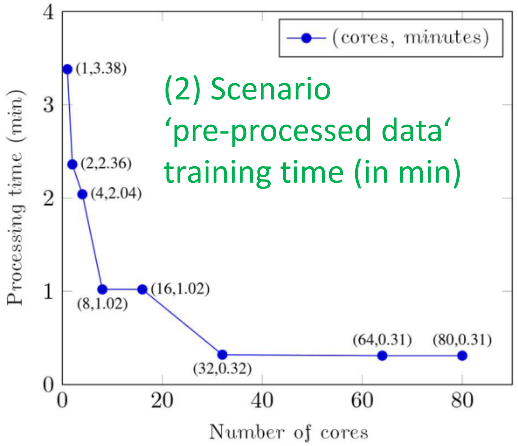
Classification Applications – Lower time to Solution

Example dataset: high number of classes & mixed pixels

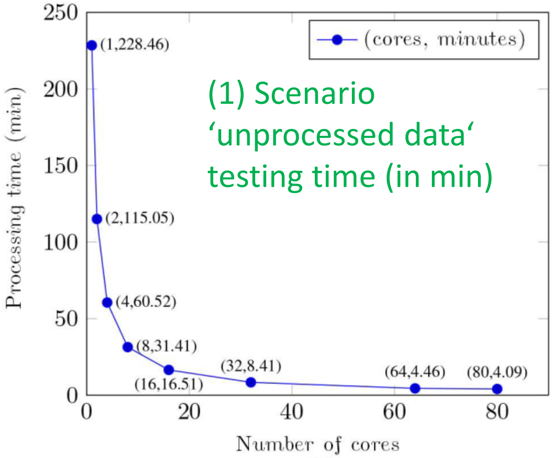
- Parallelization benefits: major speed-ups, ~interactive (<1 min) possible



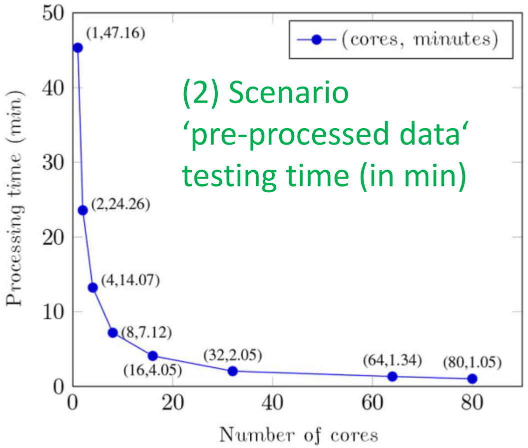
(a)



(a)



(b)



(b)

manual & serial activities (in min)

	kpca	esdap	nwfe	10x CSV	Training	Test	Total
(1) Scenario	0	0	0	4.47×10^3	10.45	71.08	4.55×10^3
(2) Scenario	5	15.38	1	529.55	1.37	23.25	575.55

'big data' is not always better data

	(1) Scenario	(2) Scenario
Number of features	200	30
Overall Accuracy (%)	40.68	77.96

[7] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

B2SHARE Store and Share Research Data

Classification Applications – Cross-Validation Benefits

2x benefits of parallelization (shown in n-fold cross validation)

- (1) Compute parallel; (2) Do all cross-validation runs in parallel (all cells)
- Evaluation between Matlab (aka ‘serial laptop‘) & parallel piSVM (80 cores)
- 10x cross-validation (RBF kernel parameter γ and C, aka ‘gridsearch‘)

(1) Scenario ‘unprocessed data‘, 10xCV **serial**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	27.30 (109.78)	34.59 (124.46)	39.05 (107.85)	37.38 (116.29)	37.20 (121.51)
4	29.24 (98.18)	37.75 (85.31)	38.91 (113.87)	38.36 (119.12)	38.36 (118.98)
8	31.31 (109.95)	**39.68** (118.28)	39.06 (112.99)	39.06 (190.72)	39.06 (872.27)
16	33.37 (126.14)	39.46 (171.11)	39.19 (206.66)	39.19 (181.82)	39.19 (146.98)
32	34.61 (179.04)	38.37 (202.30)	38.37 (231.10)	38.37 (240.36)	38.37 (278.02)

(1) Scenario ‘unprocessed data‘‘10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	27.26 (3.38)	34.49 (3.35)	39.16 (5.35)	37.56 (11.46)	37.57 (13.02)
4	29.12 (3.34)	37.58 (3.38)	38.91 (6.02)	38.43 (7.47)	38.43 (7.47)
8	31.24 (3.38)	**39.77 (4.09)**	39.14 (5.45)	39.14 (5.42)	39.14 (5.43)
16	33.36 (4.09)	39.61 (4.56)	39.25 (5.06)	39.25 (5.27)	39.25 (5.10)
32	34.61 (5.13)	38.37 (5.30)	38.36 (5.43)	38.36 (5.49)	38.36 (5.28)

First Result: best parameter set from 118.28 min to 4.09 min
Second Result: all parameter sets from ~3 days to ~2 hours

(2) Scenario ‘pre-processed data‘, 10xCV **serial**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	48.90 (18.81)	65.01 (19.57)	73.21 (20.11)	75.55 (22.53)	74.42 (21.21)
4	57.53 (16.82)	70.74 (13.94)	75.94 (13.53)	76.04 (14.04)	74.06 (15.55)
8	64.18 (18.30)	74.45 (15.04)	**77.00** (14.41)	75.78 (14.65)	74.58 (14.92)
16	68.37 (23.21)	76.20 (21.88)	76.51 (20.69)	75.32 (19.60)	74.72 (19.66)
32	70.17 (34.45)	75.48 (34.76)	74.88 (34.05)	74.08 (34.03)	73.84 (38.78)

(2) Scenario ‘pre-processed data‘, 10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	75.26 (1.02)	65.12 (1.03)	73.18 (1.33)	75.76 (2.35)	74.53 (4.40)
4	57.60 (1.03)	70.88 (1.02)	75.87 (1.03)	76.01 (1.33)	74.06 (2.35)
8	64.17 (1.02)	74.52 (1.03)	**77.02 (1.02)**	75.79 (1.04)	74.42 (1.34)
16	68.57 (1.33)	76.07 (1.33)	76.40 (1.34)	75.26 (1.05)	74.53 (1.34)
32	70.21 (1.33)	75.38 (1.34)	74.69 (1.34)	73.91 (1.47)	73.73 (1.33)

First Result: best parameter set from 14.41 min to 1.02 min
Second Result: all parameter sets from ~9 hours to ~35 min

[7] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

B2SHARE
Store and Share Research Data

Summary

Scientific Peer Review is essential to progress in the field

- Work in the field needs to be guided & steered by communities
- NIC Scientific Big Data Analytics (SBDA) first step (learn from HPC)
- Towards enabling reproducability by uploading runs and datasets

Selected SBDA by HPC benefit from parallelization

- Statistical data mining techniques able to reduce ‘big data‘ (e.g. PCA, etc.)
- Benefits in n-fold cross-validation & raw data, less on preprocessed data
- Two codes available to use and maintained @JSC: HPDBSCAN, piSVM
- HPDBSCAN and piSVM work on JURECA (less useful on JUQUEEN)

Number of ‘Data Analytics et al.‘ technologies incredible high

- (Less) open source & working versions available, often paper studies
- Evaluating approaches hard: HPC, map-reduce, Spark, SciDB, MaTex, …
- Collection of codes in Juelich Machine Learning Library (JUML) started…

References

[1] M. Goetz, M. Riedel et al.,' On Parallel and Scalable Classification and Clustering Techniques for Earth Science Datasets' 6th Workshop on Data Mining in Earth System Science, Proceedings of the International Conference of Computational Science (ICCS), Reykjavik, Online: http://www.proceedings.com/26605.html

[2] Ester, Martin, et al. "A density-based algorithm for discoveringclusters in large spatial databases with noise." Kdd. Vol. 96. 1996.

[3] M.Goetz & C. Bodenstein, Clustering Highly Parallelizable DBSCAN Algorithm, JSC, Online: http://www.fz-juelich.de/ias/jsc/EN/Research/DistributedComputing/DataAnalytics/Clustering/Clustering_node.html

[4] M.Goetz, M. Riedel et al.,'HPDBSCAN – Highly Parallel DBSCAN', MLHPC Workshop at Supercomputing 2015, Online: http://www.wikicfp.com/cfp/servlet/event.showcfp?eventid=46948

[5] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20(3), pp. 273–297, 1995

[6] Original piSVM tool, online: http://pisvm.sourceforge.net/

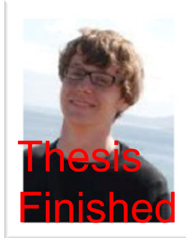
[7] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., 'On Understanding Big Data Impacts in Remotely Sensed Image Classification using Support Vector Machine Methods', IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015, DOI: 10.1109/JSTARS.2015.2458855

[8] Th. Lippert, D. Mallmann, M. Riedel, '*Scientific Big Data Analytics by HPC',* Publication Series of the John von Neumann Institute for Computing (NIC) NIC Series 48, 417, ISBN 978-3-95806-109-5, pp. 1 - 10, 2016

Acknowledgements

PhD Student Gabriele Cavallaro, University of Iceland

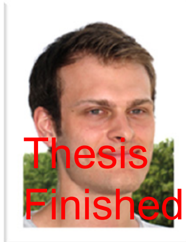
Tómas Philipp Runarsson, Kristján Jonasson,
Jón Atli Benediktsson, University of Iceland



Timo Dickscheid, Markus Axer, Stefan Köhnen, Tim Hütz,
Institute of Neuroscience & Medicine, Forschungszentrum Juelich

Selected Members of the Research Group on High Productivity Data Processing

Ahmed Shiraz Memon
Mohammad Shahbaz Memon
Markus Goetz
Christian Bodenstein
[Philipp Glock (moved to INM)]
Matthias Richerzhagen



Thanks

Talk soon available at: www.morrisriedel.de/talks