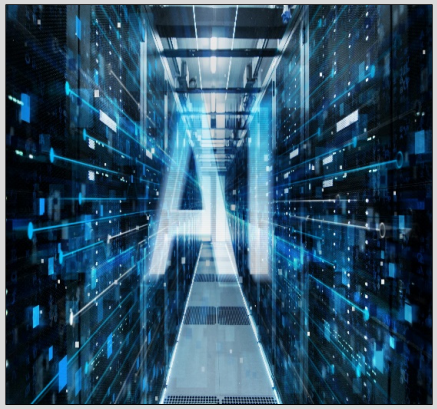


Artificial Intelligence Data Analysis (AIDA)

1st School for Heliophysicists

Prof. Dr. – Ing. Morris Riedel
Associated Professor
School of Engineering and Natural Sciences, University of Iceland, Reykjavik, Iceland
Research Group Leader, Juelich Supercomputing Centre, Forschungszentrum Juelich, Germany

@Morris Riedel @MorrisRiedel @MorrisRiedel

LECTURE 2

Unsupervised Learning – Clustering

January 20, 2020
CINECA, Bologna, Italy

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

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

DEEP Projects

HELMHOLTZ AI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT

Review of Lecture 1 – Introduction & Differences between AI, ML, NN & Big Data

Artificial Intelligence (AI)

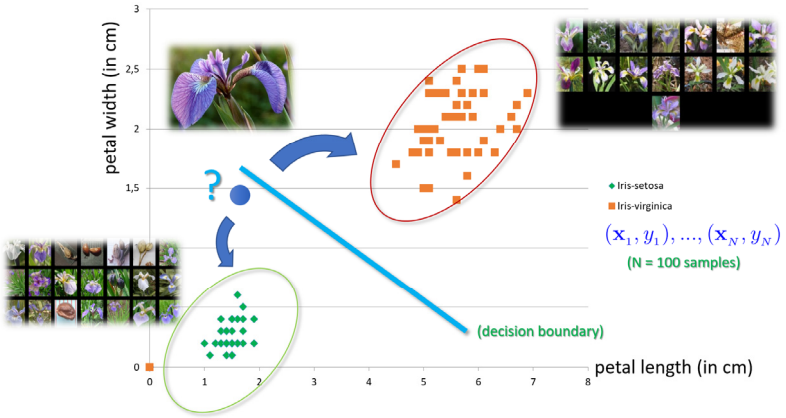
A wide area of techniques and tools that enable computers to mimic human behaviour (+ robotics)

Machine Learning (ML)

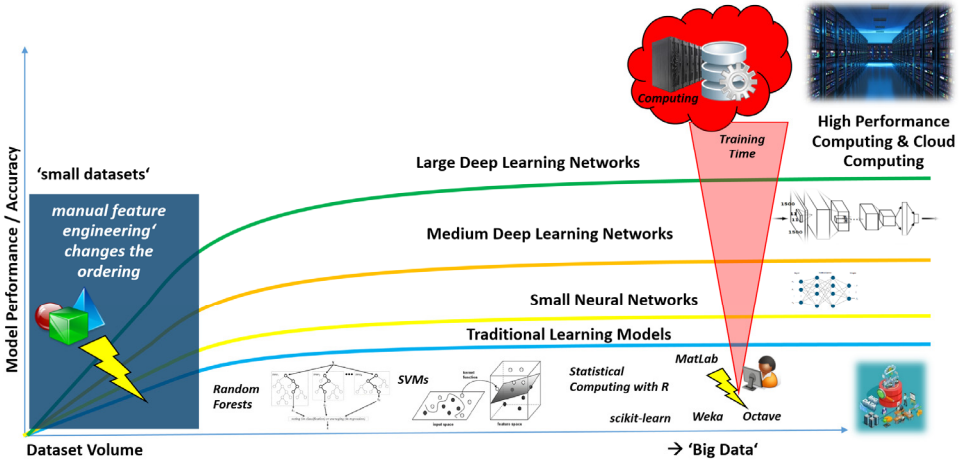
Learning from data without explicitly being programmed with common programming languages

Deep Learning (DL)

Systems with the ability to learn underlying features in data using large neural networks



[1] Image sources: Species Iris Group of North America Database, www.signa.org



Outline of the School

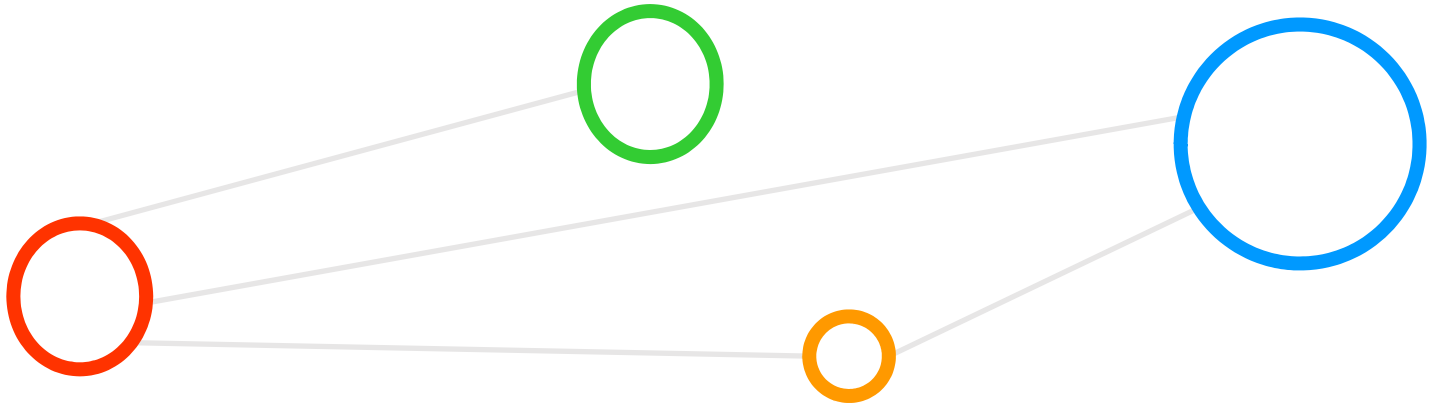
Time	**Day 1**	**Day 2**	**Day 3**
9 - 10	Welcome and intro to the school (Giovanni Lapenta, Jorge Amaya)	Space missions data acquisition (Hugo Breuillard)	Review of ML applied to heliophysics (Peter Wintoft)
10 - 11	Introduction and differences between AI, ML, NN and Big Data (Morris Riedel)	Data manipulation in python with pandas, xarray, and additional python tools (Geert Jan Bex)	Review of ML applied to heliophysics (Peter Wintoft)
	Coffee break	Coffee break	Coffee break
11:30 - 12:30	Unsupervised learning (Morris Riedel)	Feature engineering and data reduction (Geert Jan Bex)	Reinforcement learning (Morris Riedel)
	Lunch	Lunch	Lunch
14 - 15	Unsupervised learning (Morris Riedel)	Data reduction and visualization (Geert Jan Bex)	Physics informed ML (Romain Dupuis)
15 -16	Supervised learning (Morris Riedel)	CNN, DNN (Morris Riedel)	Explainable AI (Jorge Amaya)
	Coffee break	Coffee break	Coffee break
16:30 - 18:00	Supervised learning (Morris Riedel)	CNN, DNN (Morris Riedel)	Performance and tuning of ML (Morris Riedel)

Outline

- Unsupervised Learning with Clustering
  - Formalization of Unsupervised Learning
  - Clustering Methods and Approaches
  - K-Means & K-Median Clustering Algorithms
  - Simple Application Examples
  - DBSCAN Clustering Algorithm

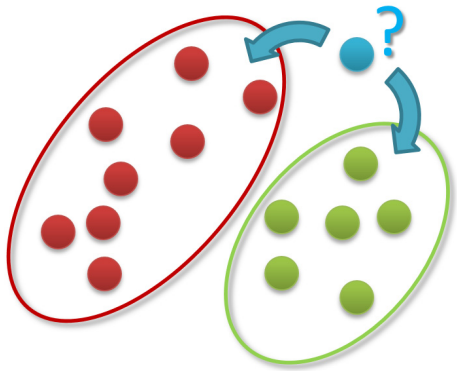
- Point Cloud Applications
  - Introduction to Application Domain
  - Bremen Datasets & Locations
  - Hierarchical Data Format (HDF) Basics
  - Parallel HPDBSCAN Algorithm & Point Cloud Library (PCL) format
  - Data Visualization with HDF Viewer & PCL Viewer

Unsupervised Clustering



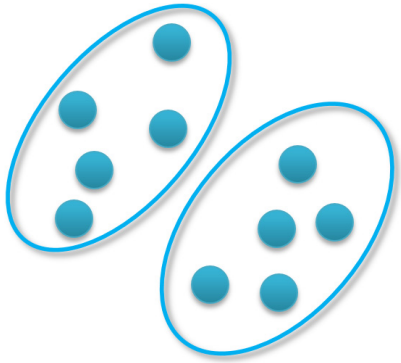
Machine Learning Models – Short Overview

Classification



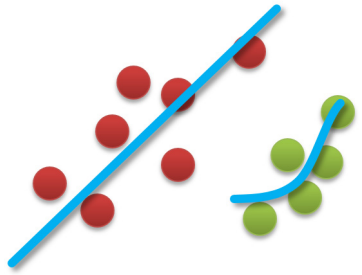
- Groups of data exist
- New data classified to existing groups

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

- **Machine learning methods can be roughly categorized in classification, clustering, or regression augmented with various techniques for data exploration, selection, or reduction – despite the momentum of deep learning, traditional machine learning algorithms are still widely relevant today**

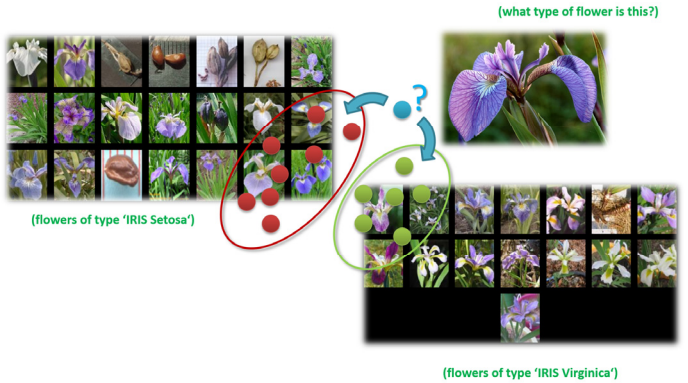
Learning Approaches – What means Learning from data?

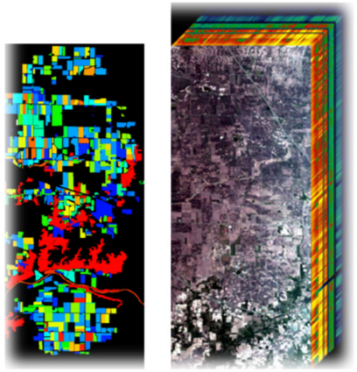
- **The basic meaning of learning is 'to use a set of observations to uncover an underlying process'**
- **The three different learning approaches are supervised, unsupervised, and reinforcement learning**

[14] Image sources: Species Iris Group of North America Database, www.signa.org

Supervised Learning

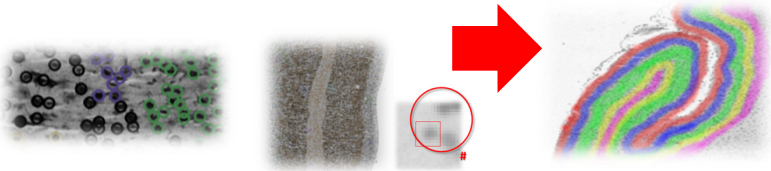
- Majority of methods follow this approach in this course
- Example: credit card approval based on previous customer applications



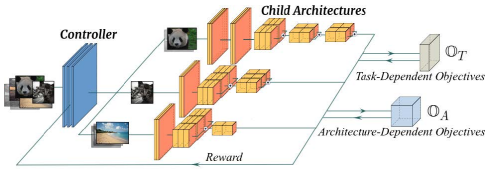


Unsupervised Learning

- Often applied before other learning → higher level data representation
- Example: Coin recognition in vending machine based on weight and size



[15] A.C. Cheng et al., 'InstaNAS: Instance-aware Neural Architecture Search', 2018



Reinforcement Learning

- Typical 'human way' of learning
- Example: Toddler tries to touch a hot cup of tea (again and again)

➢ **Day 1 offers details about unsupervised & supervised learning with examples & Day 3 offers an introduction to reinforcement learning**

Learning Approaches – Unsupervised Learning

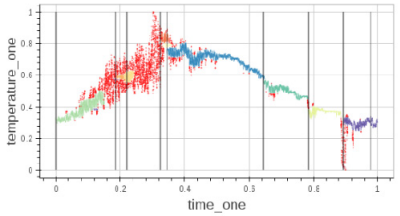
- Each observation of the predictor measurement(s) has no associated response measurement:
  - Input $\mathbf{x} = x_1, \ldots, x_d$
  - No output
  - Data $(\mathbf{x}_1), \ldots, (\mathbf{x}_N)$
- Goal: Seek to understand relationships between the observations
  - Clustering analysis: check whether the observations fall into distinct groups
- Challenges
  - No response/output that could supervise our data analysis
  - Clustering groups that overlap might be hardly recognized as distinct group

- **Unsupervised learning approaches seek to understand relationships between the observations**
- **Unsupervised learning approaches are used in clustering algorithms such as k-means, etc.**
- **Unupervised learning works with data = [input, ---]**

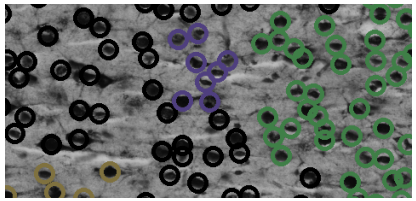
[2] An Introduction to Statistical Learning

Learning Approaches – Unsupervised Learning Use Cases

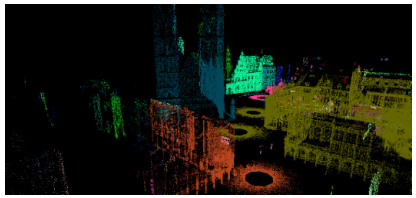
- Earth Science Data (e.g. PANGAEA repository)
  - Automatic quality control and event detection
  - Collaboration with the University of Gothenburg
  - Koljoefjords Sweden – Detect water mixing events



- Human Brain Data
  - Analyse human brain images as brain slices
  - Segment cell nuclei in brain slice images
  - Step in detecting layers of the cerebral cortex

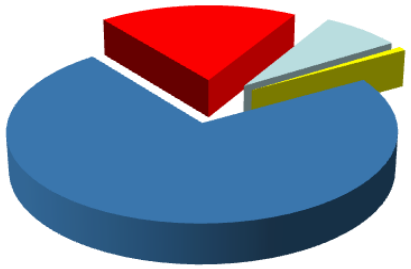


- Point Cloud Data
  - Analysis of point cloud datasets of various sizes
  - 3D/4D LIDAR scans of territories (cities, ruins, etc.)
  - Filter noise and reconstruct objects

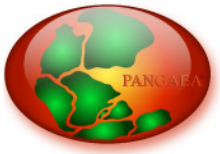


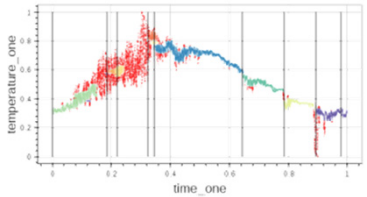
Unsupervised Learning – Earth Science Data Example

- 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



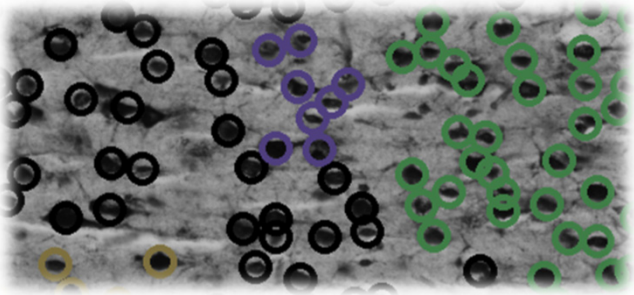
Total number of data sets 349 871
Data items ~ 7.9 billions

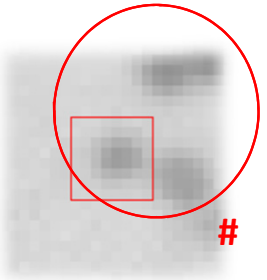


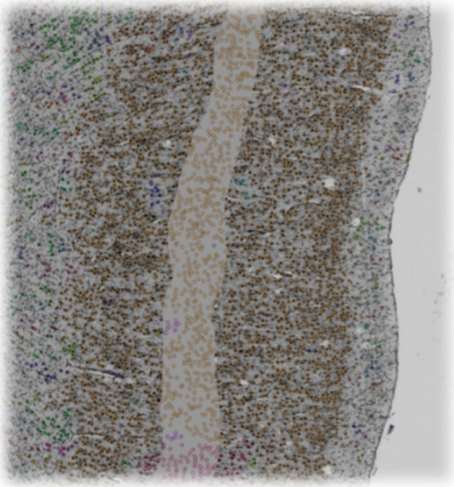


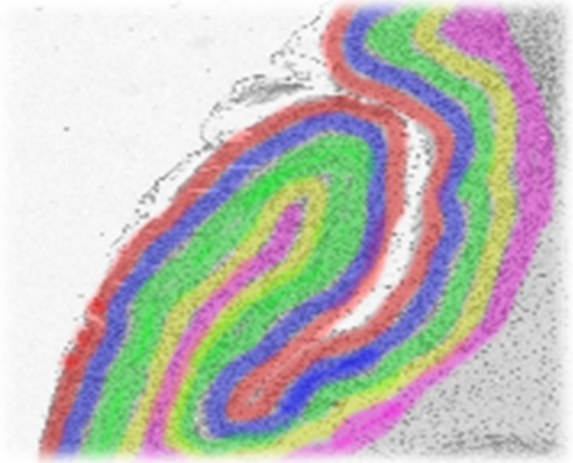
[3] PANGAEA data collection

Unsupervised Learning – Human Brain Data Example



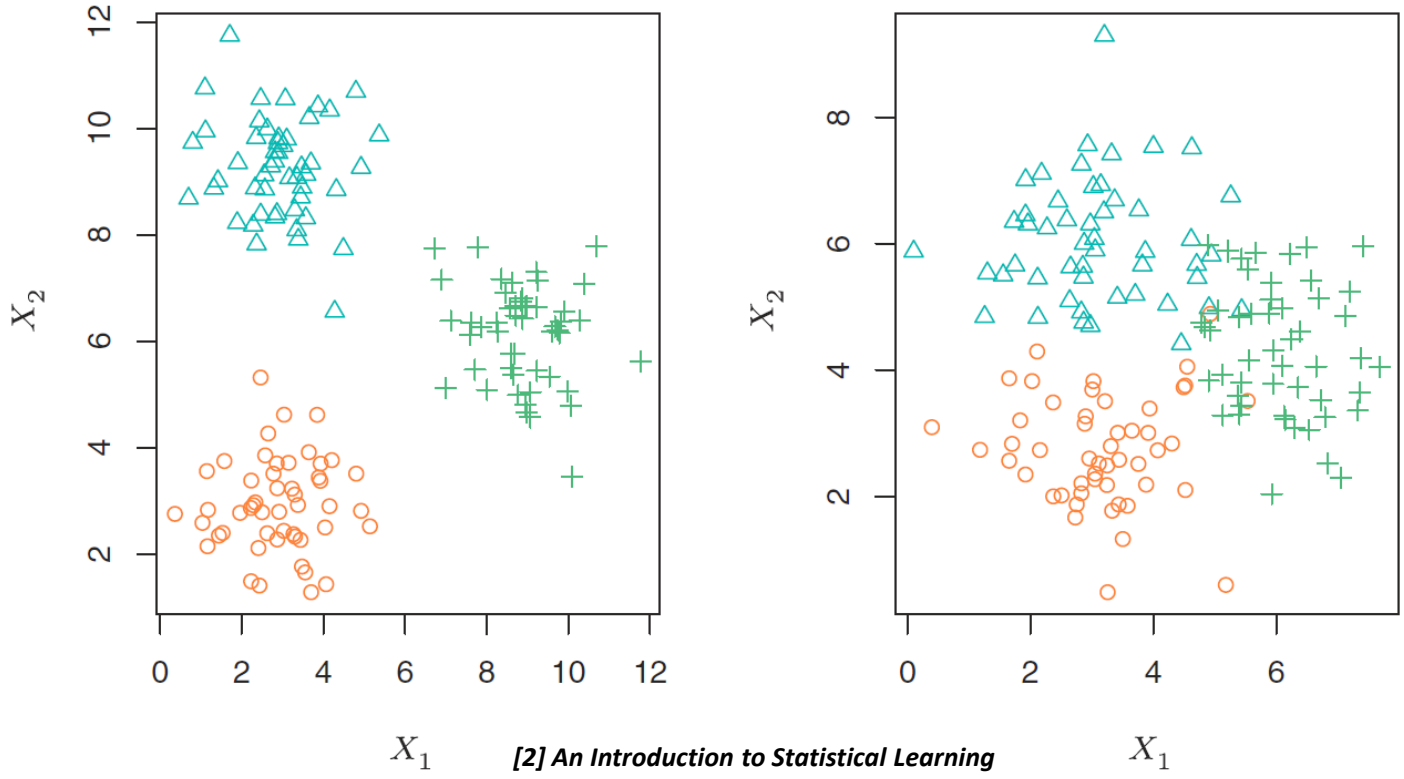






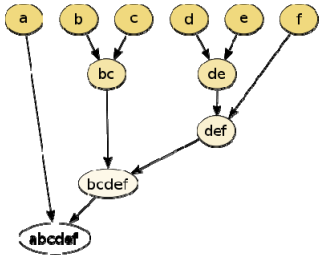
Learning Approaches – Unsupervised Learning Challenges

- Practice: The number of clusters can be ambiguities

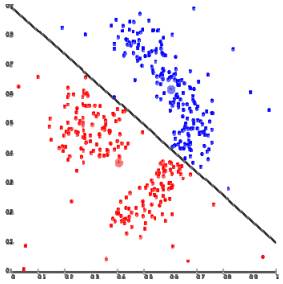


[2] An Introduction to Statistical Learning

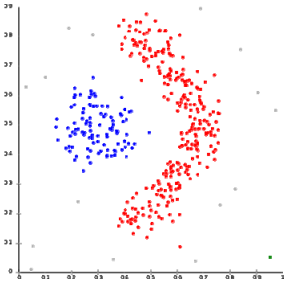
Unsupervised Learning – Different Clustering Approaches



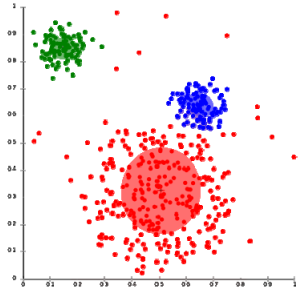
(hierarchical)



(centroid)



(density)



(distribution)

- **Clustering approaches can be categorized into four different approaches: (1) hierarchical, (2) centroid, (3) density, (4) distribution**

[2] An Introduction to Statistical Learning

Unsupervised Learning – Clustering Methods

- Characterization of clustering tasks
  - No prediction as there is no associated response Y to given inputs X
  - Discovering interesting facts & relationships about the inputs X
  - Partitioning of data in subgroups (i.e. ‘clusters‘) previously unknown
  - Being more subjective (and more challenging) than supervised learning

- Considered often as part of ‘exploratory data analysis‘
  - Assessing the results is hard, because no real validation mechanism exists
  - Simplifies data via a ‘small number of summaries’ good for interpretation

- **Clustering are a broad class of methods for discovering previously unknown subgroups in 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)
  - Assumes clusters of similar density or areas of higher density in dataset

Clustering Methods – Similiarity Measures

- How to partition data into distinct groups?
  - Data in same (homogenous) groups are somehow ‘similiar‘ to each other
  - Data not in same sub-groups are somehow ‘different‘ from each other
  - Concrete definitions of ‘similiarity‘ or ‘difference‘ often domain-specific
- Wide variety of similiarity measures exist, e.g. distance measures
  - Jaccard Distance, Cosine Distance, Edit Distance, Hamming Distance, …

> **A distance measure in some space is a function d(x,y) that takes two points in the space as arguments and produces a real number**

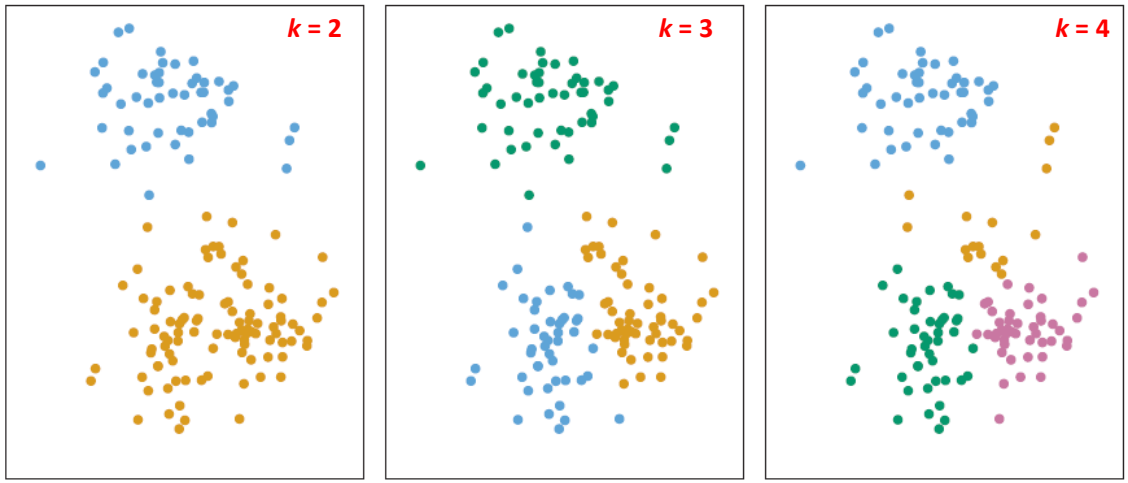
- Often used ‘similiarity measure‘ example
  - Distance-based: Euclidean distance
  - n-dimensional Euclidean space:
    A space where points are vectors of n real numbers

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

(ruler distance)

Clustering Methods – K-Means Approach

- Approach Overview
  - Partitions a data set into *K* distinct (i.e. non-overlapping) clusters
  - Requires the definition of the desired number of clusters *K* in advance
  - Assigns each observation / data element to exactly one of the *K* clusters
  - Example: 150 observations; 2 dimensions; 3 different values of *K*



[2] An Introduction to Statistical Learning

Clustering Methods – K-Means Algorithm

Set the desired number of clusters *K*

- Picking the right number k is not simple (→ later)

1. Randomly assign a number from 1 to K to each observation

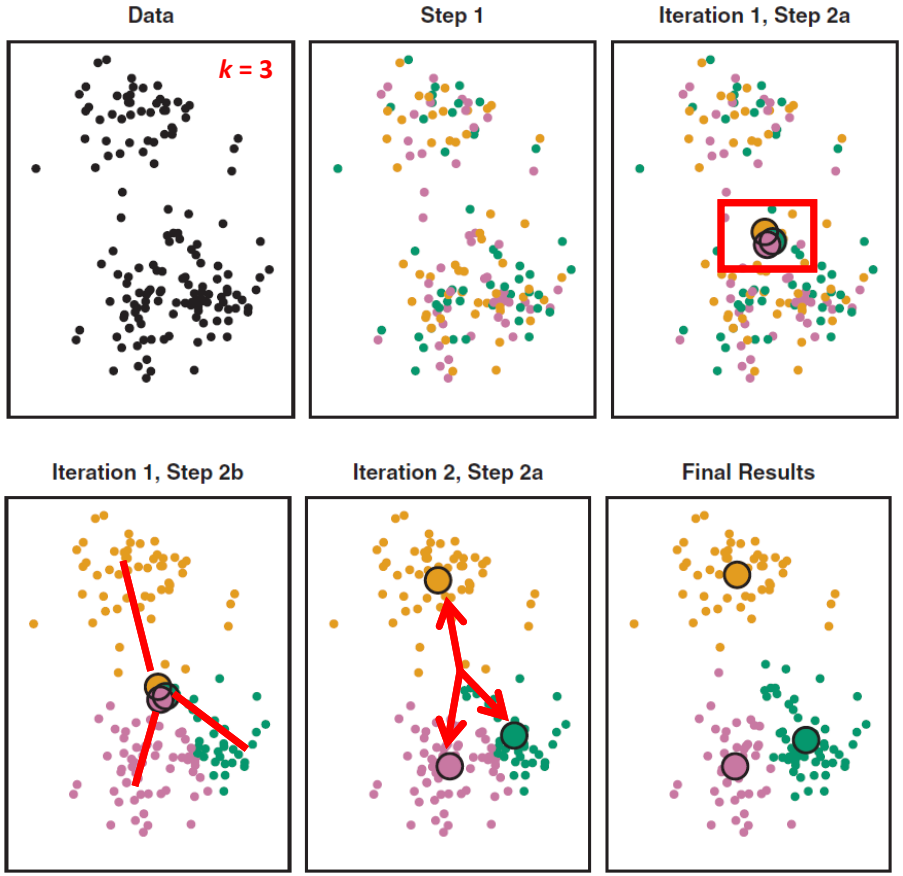
- Initializes cluster assignments for the observations
- Requires algorithm execution multiple times
  (results depend on random assignment, e.g. pick ‘best‘ after 6 runs)

2. Iterate until the cluster assignments stop changing

a. For each of the K clusters: compute the cluster centroid
   - The kth cluster centroid is the vector of the p feature means
     for all the observations in the kth cluster

b. Assign each observation to the cluster K whose centroid is closest
   - The definition of ‘closest‘ is the Euclidean distance

Clustering Methods – K-Means Algorithm Example

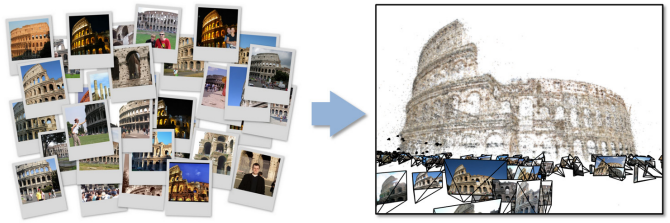


[2] An Introduction to Statistical Learning

1. Randomly assign a number from 1 to K to each observation
2. Iterate until the cluster assignments stop changing
   a. For each of the K clusters: compute the cluster centroid [centroids appear and move]
   b. Assign each observation to the cluster K whose centroid is closest [Euclidean distance]

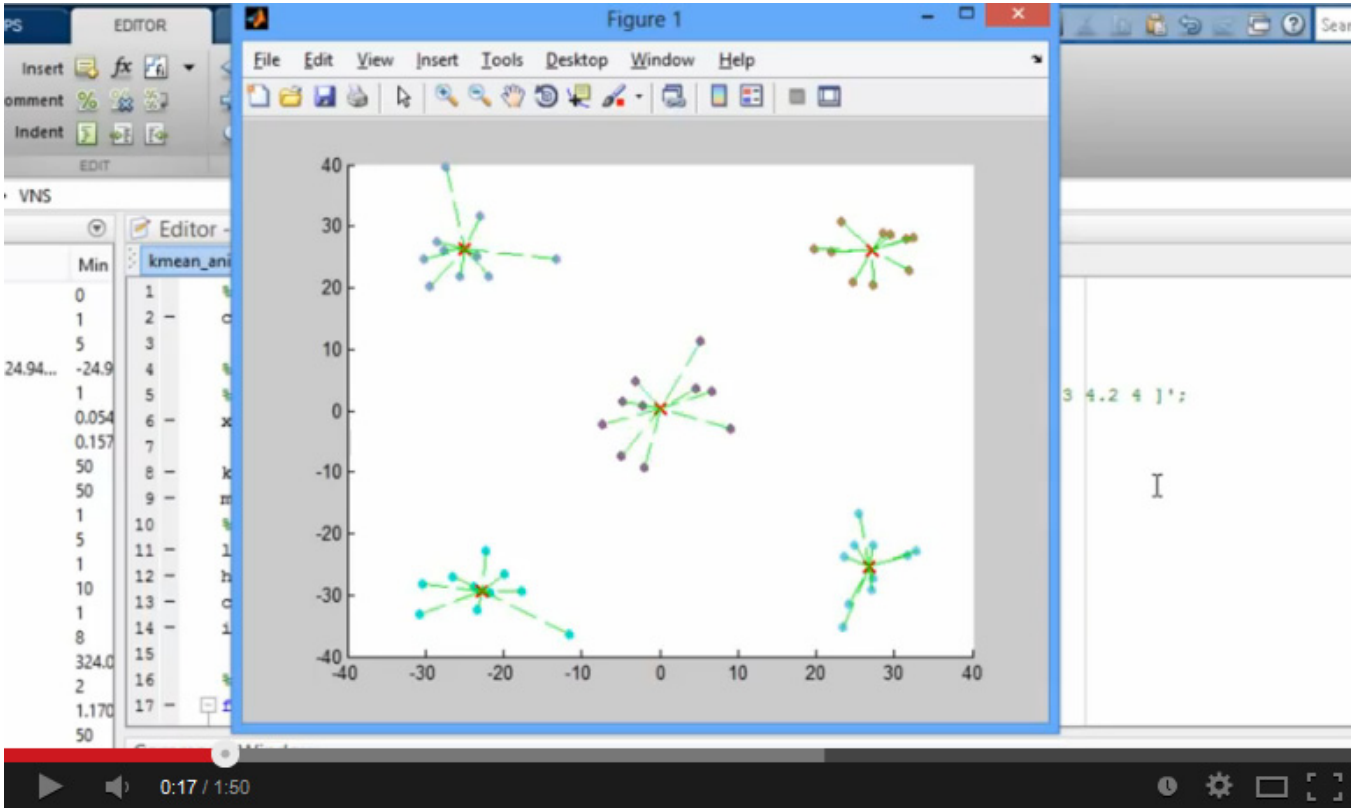
Clustering Methods – K-Means Usage

- Advantages
  - Handles large datasets (larger than hierarchical cluster approaches)
  - Move of observations / data elements between clusters (often improves the overall solution)
- Disadvantages
  - Use of ‘means‘ implies that all variables must be continous
  - Severaly affected by datasets with outliers (→ means)
  - Perform poorly in cases with non-convex (e.g. U-shaped) clusters
- ‘Big Data‘ Application Example
  - Image processing: 7 million images
  - 512 features/attributes per image;
  - 1 million clusters
  - 10000 Map tasks; 64GB broadcasting;
  - 20 TB intermediate data in shuffling;



[4] Judy Qiu, ‘Collective communication on Hadoop’, 2014

[Video] K-Means Clustering



[5] Animation of the k-means clustering algorithm, YouTube Video

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

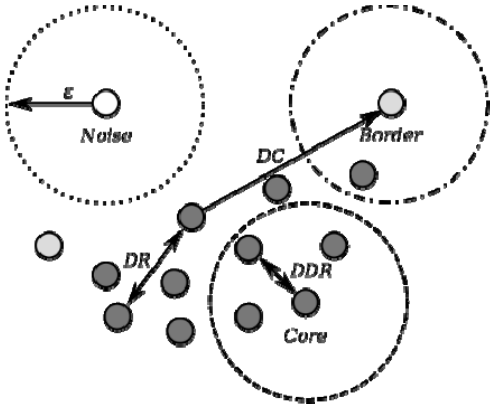
DBSCAN Algorithm

- DBSCAN Algorithm
  - Introduced 1996 and most cited clustering algorithm
  - 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 (except ‘bow ties‘)
  - Identifies inherently also outliers/noise

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

[6] Ester et al.



(MinPoints = 4)

(DR = Density Reachable)

(DDR = Directly Density Reachable)

(DC = Density Connected)

Exercise: DBSCAN Algorithm – Non-Trivial Example

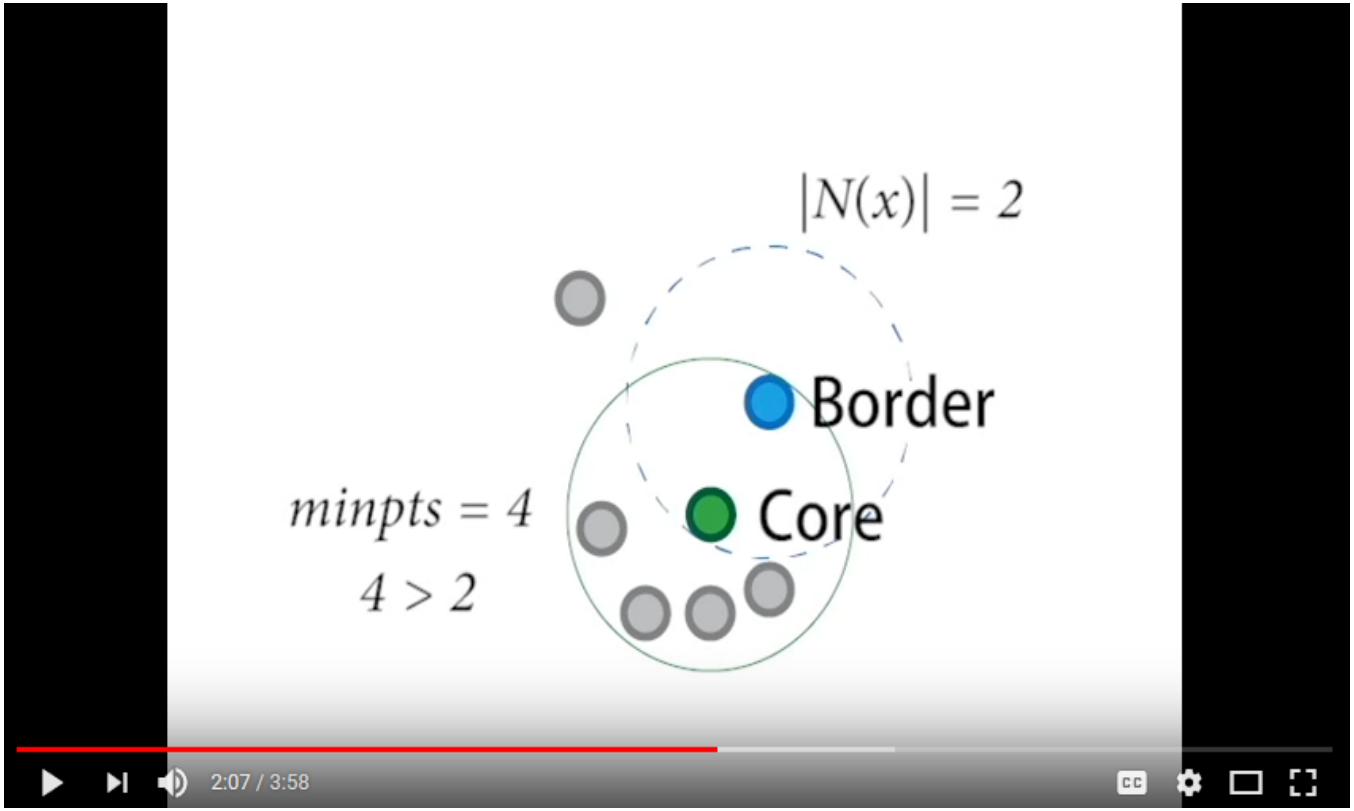
- Compare K-Means vs. DBSCAN – How would K-Means work?

Unclustered Data

Clustered Data

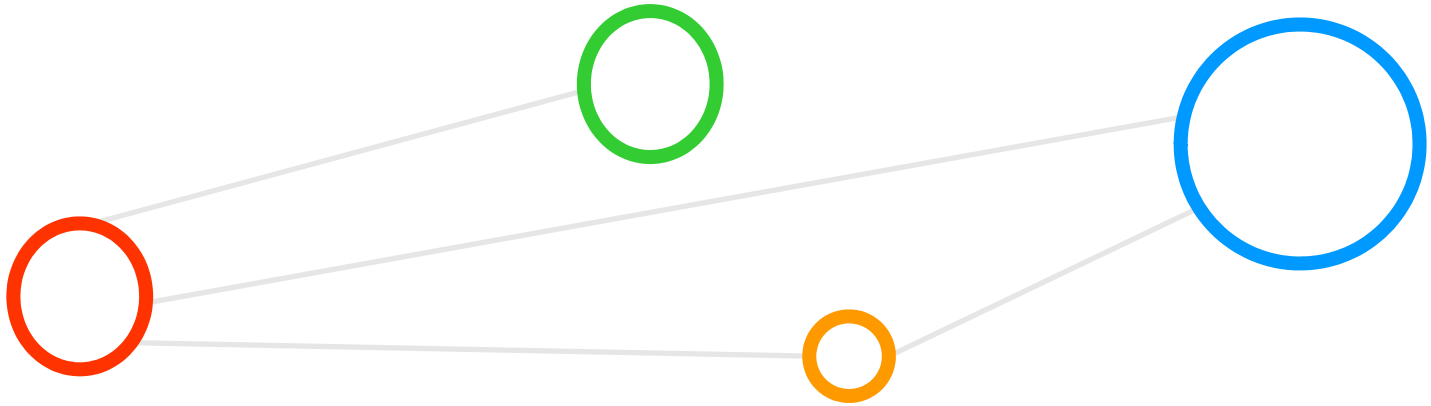
- **DBSCAN forms arbitrarily shaped clusters (except 'bow ties') where other clustering algorithms fail**

[Video] DBSCAN Clustering Algorithm



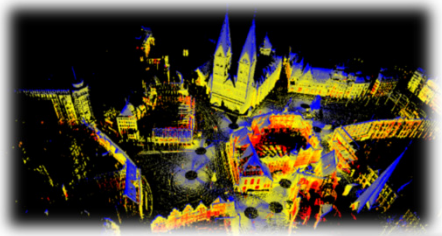
[7] YouTube Video, DBSCAN

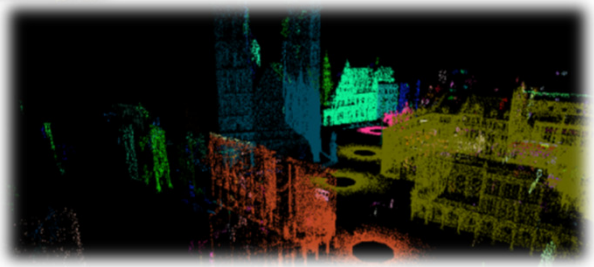
Point Cloud Applications



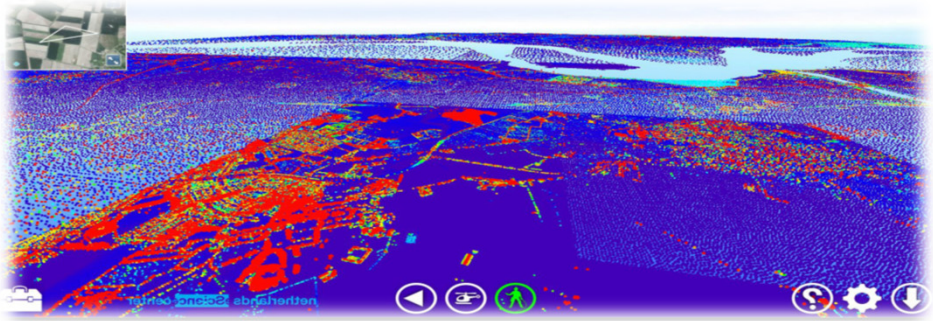
Point Cloud Applications

- ‘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)
  - Study level of continous details



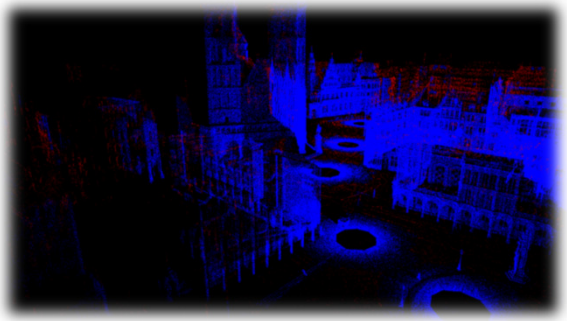
Point Cloud Application Example – Within Buildings

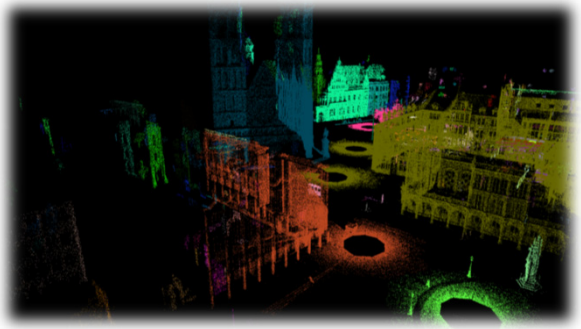
- Point based rendering example
  - Aachen Cathedral based on 3D laser scans and photos
  - Points are rendered as textured and blended splats
  - Visualisation can run in real-time on a desktop PC showing 6 million splats based of a 120 million point laser scan

[8] Aachen Cathedral Point Cloud Rendering, YouTube Video

Bremen Dataset Online in B2SHARE – Attention: Will be edited with ClusterID!

- Different clusterings of the inner city of Bremen
  - Using smart visualizations of the point cloud library (PCL)
  - Big Bremen (81 mio points) & sub sampled Small Bremen (3 mio points)





```
[train001@jrl07 bremen]$ pwd
/homea/hpclab/train001/data/bremen
[train001@jrl07 bremen]$ ls -al
total 1342208
drwxr-xr-x 2 train001 hpclab        512 Jan 14 09:58 .
drwxr-xr-x 4 train001 hpclab        512 Jan 14 08:38 ..
-rw-r--r-- 1 train001 hpclab 1302382632 Jan 14 09:56 bremen.h5
-rw-r--r-- 1 train001 hpclab   72002416 Jan 14 08:25 bremenSmall.h5
```

- **The Bremen Dataset is encoded in the HDF5 format (binary)**
- **You need your own copy of the file in your home directory to cluster!**

[9] Bremen Dataset

Exercises – Explore & Copy Bremen HDF5 Datasets (binary)

- Copy Bremen datasets to your own home directory (~)

```
[train001@jrl07 bremen]$ pwd
/homea/hpclab/train001/data/bremen
[train001@jrl07 bremen]$ cp * ~
```

- Check your home directory for the Bremen datasets

```
[train001@jrl07 bremen]$ cd ~
[train001@jrl07 ~]$ ls -al
total 1341824
drwxr-x---  13 train001 hpclab      32768 Jan 14 09:44 .
drwxr-xr-x 302 root     sys        32768 Mar 25  2013 ..
-rw-------   1 train001 hpclab       7547 Jan 14 08:28 .bash_history
-rw-r--r--   1 train001 hpclab         18 Jan  8 08:58 .bash_logout
-rw-r--r--   1 train001 hpclab        176 Jan  8 08:58 .bash_profile
-rw-r--r--   1 train001 hpclab        124 Jan  8 08:58 .bashrc
drwxr-xr-x   3 train001 hpclab        512 Jan 14 00:28 bin
-rw-r--r--   1 train001 hpclab 1302382632 Jan 14 09:59 bremen.h5
-rw-r--r--   1 train001 hpclab   72002416 Jan 14 09:59 bremenSmall.h5
```

- Notice binary content

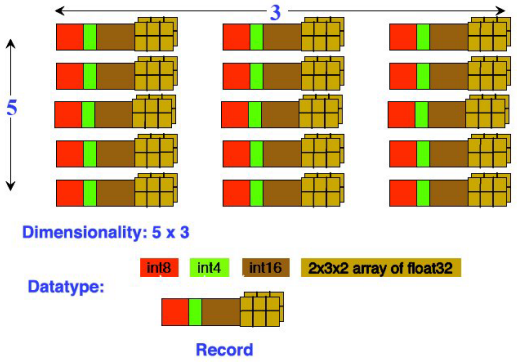
```
[train001@jrl07 ~]$ head bremen.h5
HDF
```

Hierarchical Data Format (HDF)

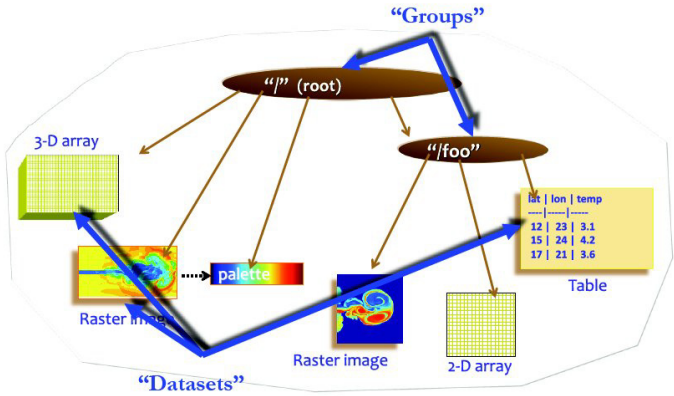
- **HDF is a technology suite that enables the work with extremely large and complex data collections**

[13] HDF@ I/O workshop

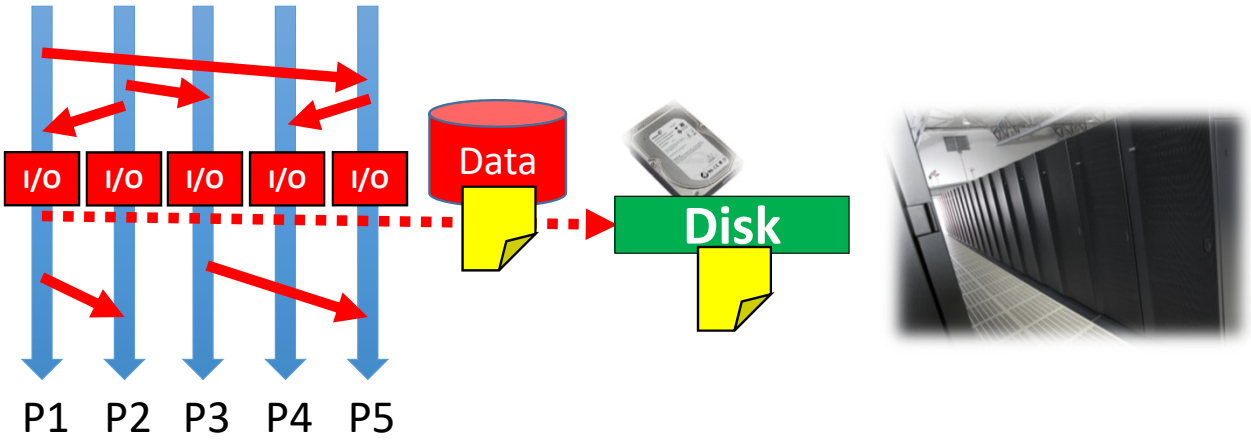
- Simple ‘compound type‘ example:
  - Array of data records with some descriptive information (5x3 dimension)
  - HDF5 data structure type with int(8); int(4); int(16); 2x3x2 array (float32)



‘HDF5 file is a container’ to organize data objects



HDF5 – Parallel I/O: Shared file



- Each process performs I/O to a single file
  - The file access is 'shared' across all processors involved
  - E.g. MPI/IO functions represent 'collective operations'
- Scalability and Performance
  - 'Data layout' within the shared file is crucial to the performance
  - High number of processors can still create 'contention' for file systems

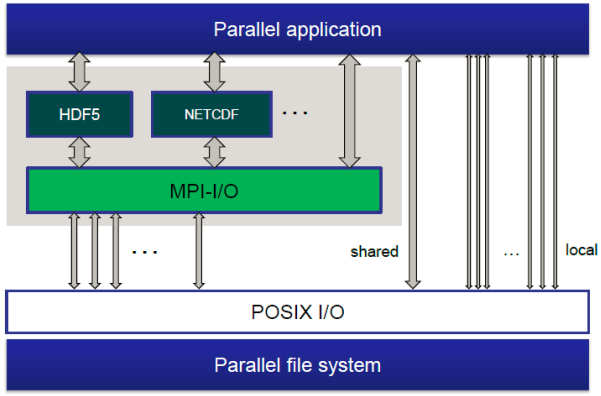
- **Parallel I/O: shared file means that processes can access their 'own portion' of a single file**
- **Parallel I/O with a shared file like MPI/IO is a scalable and even standardized solution**

HDF5 – Parallel I/O & File Systems

- **Hierarchical Data Format (HDF) is designed to store & organize large amounts of numerical data**
- **Parallel Network Common Data Form (NETCDF) is designed to store & organize array-oriented data**

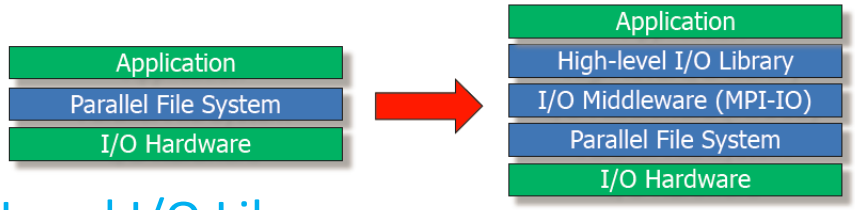
[14] HDF Group ***[15] Parallel NETCDF***

- Portable Operating System Interface for UNIX (POSIX) I/O
  - Family of standards to maintain OS compatibility, including I/O interfaces
  - E.g. read(), write(), open(), close(), …(very old interface, some say 'too old')
- 'Higher level I/O libraries' HDF5 & NETCDF
  - Integrated into a parallel application
  - Built on top of MPI I/O for portability
  - Offers machine-independent data access and data formats



I/O with Multiple Layers and Distinct Roles

- **Parallel I/O is supported by multiple software layers with distinct roles that are high-level I/O libraries, I/O middleware, and parallel file systems**



[15] R. Thakur, PRACE Training, Parallel I/O and MPI I/O

- High-Level I/O Library
  - Maps application abstractions to a structured portable file format
  - E.g. HDF-5, Parallel NetCDF
- I/O Middleware
  - E.g. MPI I/O
  - Deals with organizing access by many processes
- Parallel Filesystem
  - Maintains logical space and provides efficient access to data
  - E.g. GPFS, Lustre, PVFS

Review of Parallel DBSCAN Implementations

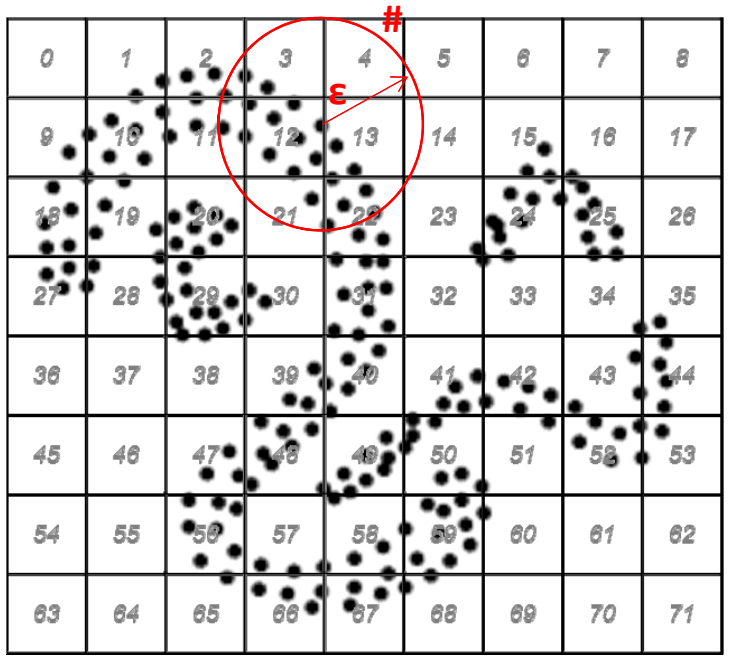
Technology	Platform Approach	Analysis
HPDBSCAN (authors implementation)	C; MPI; OpenMP	Parallel, hybrid, DBSCAN
Apache Mahout	Java; Hadoop	K-means variants, spectral, no DBSCAN
Apache Spark/MLlib	Java; Spark	Only k-means clustering, No DBSCAN
scikit-learn	Python	No parallelization strategy for DBSCAN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

[12] 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 (ICCS)***

HDBSCAN Algorithm Details

- Parallelization Strategy
  - Smart ‘Big Data‘ Preprocessing into Spatial Cells (‘indexed‘)
  - OpenMP and HDF5 parallel I/O
  - MPI (+ optional OpenMP hybrid)

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



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

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

HPC Environment – Modules

- Module environment tool
  - Avoids to manually setup environment information for every application
  - Simplifies shell initialization and lets users easily modify their environment
- Module avail
  - Lists all available modules on the HPC system (e.g. compilers, MPI, etc.)
- Module spider
  - Find modules in the installed set of modules and more information
- Module load → needed before HPDBSCAN run
  - Loads particular modules into the current work environment, E.g.:

```
[train001@jrl12 ~]$ module load GCC

Due to MODULEPATH changes, the following have been reloaded:
  1) binutils/.2.29

The following have been reloaded with a version change:
  1) GCCcore/.5.4.0 => GCCcore/.7.2.0

[train001@jrl12 ~]$ module load ParaStationMPI/5.2.0-1
[train001@jrl12 ~]$ module load HDF5/1.8.19
```

JURECA HPC System – HPDBSCAN Job Script Example

```
#!/bin/bash
#SBATCH --job-name=HPDBSCAN
#SBATCH -o HPDBSCAN-%j.out
#SBATCH -e HPDBSCAN-%j.err
#SBATCH --nodes=2
#SBATCH --ntasks=4
#SBATCH --ntasks-per-node=4
#SBATCH --time=00:20:00
#SBATCH --cpus-per-task=4
#SBATCH --reservation=ml-hpc-1

export OMP_NUM_THREADS=4

# location executable
HPDBSCAN=/homea/hpclab/train001/tools/hpdbscan/dbscan

# your own copy of bremen small
BREMENSMALLDATA=/homea/hpclab/train001/bremenSmall.h5

# your own copy of bremen big
BREMENBIGDATA=/homea/hpclab/train001/bremen.h5

srun $HPDBSCAN -m 100 -e 300 -t 12 $BREMENSMALLDATA
```

- **Job submit using command: sbatch <jobscript>**
- **Remember your <jobid> that is returned from the sbatch command**
- **Show status of the job then with: squeue –u <your-user-id>**

(parameters of DBSCAN and file to be clustered)

JURECA HPC System – HPDBSCAN Job Submit

- Load module environment (once after login)

```
[train001@jrl07 jsc_mpi]$ module load GCC

Due to MODULEPATH changes, the following have been reloaded:
  1) binutils/.2.29

The following have been reloaded with a version change:
  1) GCCcore/.5.4.0 => GCCcore/.7.2.0

[train001@jrl07 jsc_mpi]$ module load ParaStationMPI/5.2.0-1
[train001@jrl07 jsc_mpi]$ module load HDF5/1.8.19
```

- Submit job via jobscript

```
[train001@jrl07 jsc_mpi]$ sbatch submit-clustering-bremen.sh
Submitted batch job 4629728
```

- Check job status (and cancel if needed)

(scancel might take a second or two to take effect)

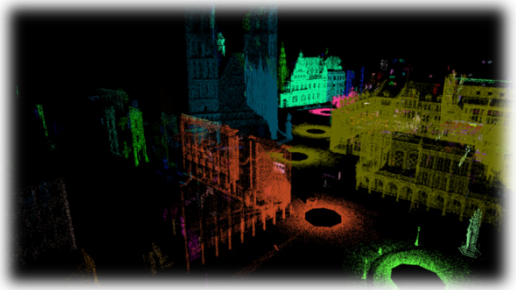
```
[train001@jrl07 hpdbscan]$ squeue -u train001
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4629867     batch HPDBSCAN train001  R       2:20      2 jrc[0672-0673]
[train001@jrl07 hpdbscan]$ scancel 4629867
[train001@jrl07 hpdbscan]$ squeue -u train001
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4629867     batch HPDBSCAN train001 CG       2:34      2 jrc[0672-0673]
[train001@jrl07 hpdbscan]$ squeue -u train001
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

JURECA HPC System – HPDBSCAN Check Outcome

```
[train001@jrl07 jsc_mpi]$ more HPDBSCAN-4629640.out
Calculating Cell Space...
        Computing Dimensions... [OK] in 0.001657
        Computing Cells...      [OK] in 0.029877
        Sorting Points...       [OK] in 0.174414
        Distributing Points...  [OK] in 0.113745
DBSCAN...
        Local Scan...           [OK] in 58.095238
        Merging Neighbors...    [OK] in 0.005433
        Adjust Labels ...       [OK] in 0.004473
        Rec. Init. Order ...    [OK] in 0.559311
        Writing File ...        [OK] in 0.008467
Result...
        65      Clusters
        2973821 Cluster Points
        26179   Noise Points
        2953129 Core Points
Took: 59.111594s
```

```
[train001@jrl07 ~]$ ls -al
total 1124800
drwxr-x---  13 train001 hpclab      32768 Jan 14 08:47 .
drwxr-xr-x 302 root     sys          32768 Mar 25  2013 ..
-rw-------   1 train001 hpclab       7547 Jan 14 08:28 .bash_history
-rw-r--r--   1 train001 hpclab         18 Jan  8 08:58 .bash_logout
-rw-r--r--   1 train001 hpclab        176 Jan  8 08:58 .bash_profile
-rw-r--r--   1 train001 hpclab        124 Jan  8 08:58 .bashrc
drwxr-xr-x   3 train001 hpclab        512 Jan 14 00:28 bin
-rw-r--r--   1 train001 hpclab 1079412312 Jan 14 08:39 bremen.h5.h5
-rw-r--r--   1 train001 hpclab   72002416 Jan 14 08:47 bremenSmall.h5.h5
```

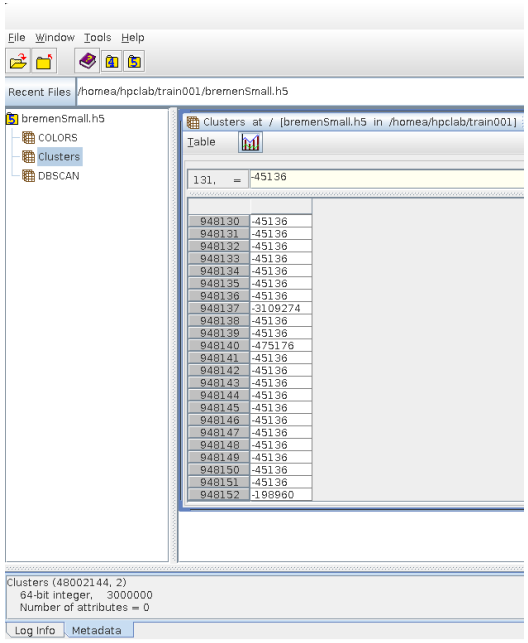
- **The outcome of the clustering process is written directly into the HDF5 file using cluster IDs and noise IDs**

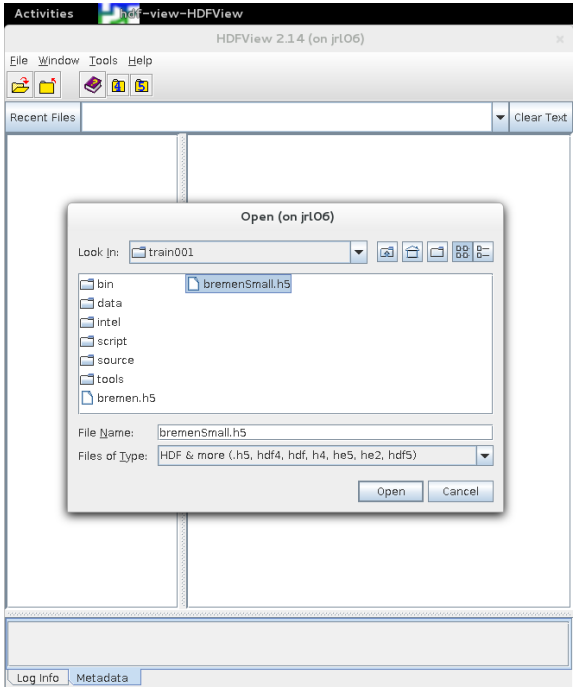


HDFView Example – Bremen Output

- HDFView is a visual tool for browsing and editing HDF files
  - Tools is using a GUI thus needs ssh –X when log into JURECA

```
[train001@jrl06 ~]$ module load HDFView/2.14-Java-1.8.0_144
[train001@jrl06 ~]$ hdfview.sh
```

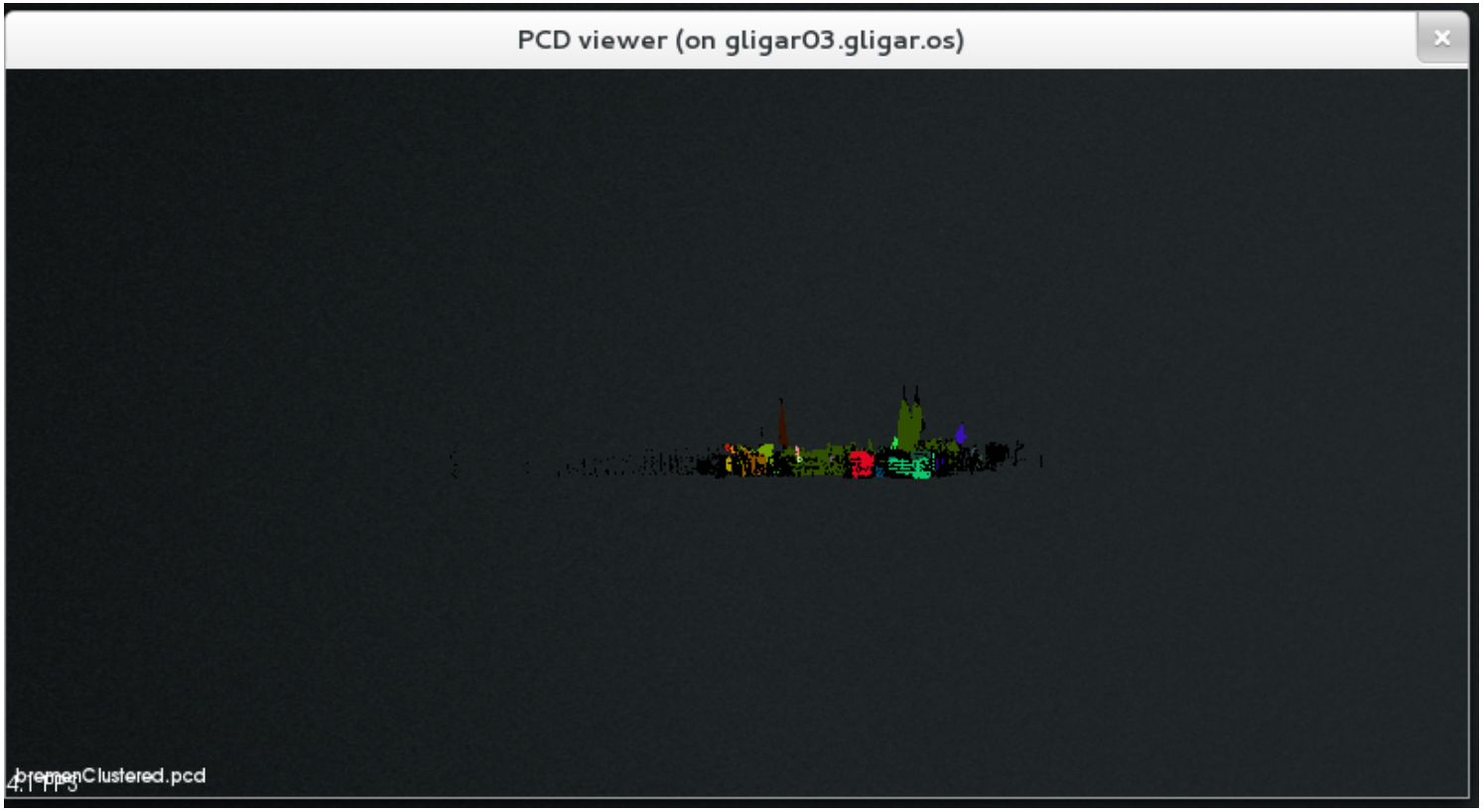




Point Cloud Viewer Example – Bremen Output & Point Cloud Viewer

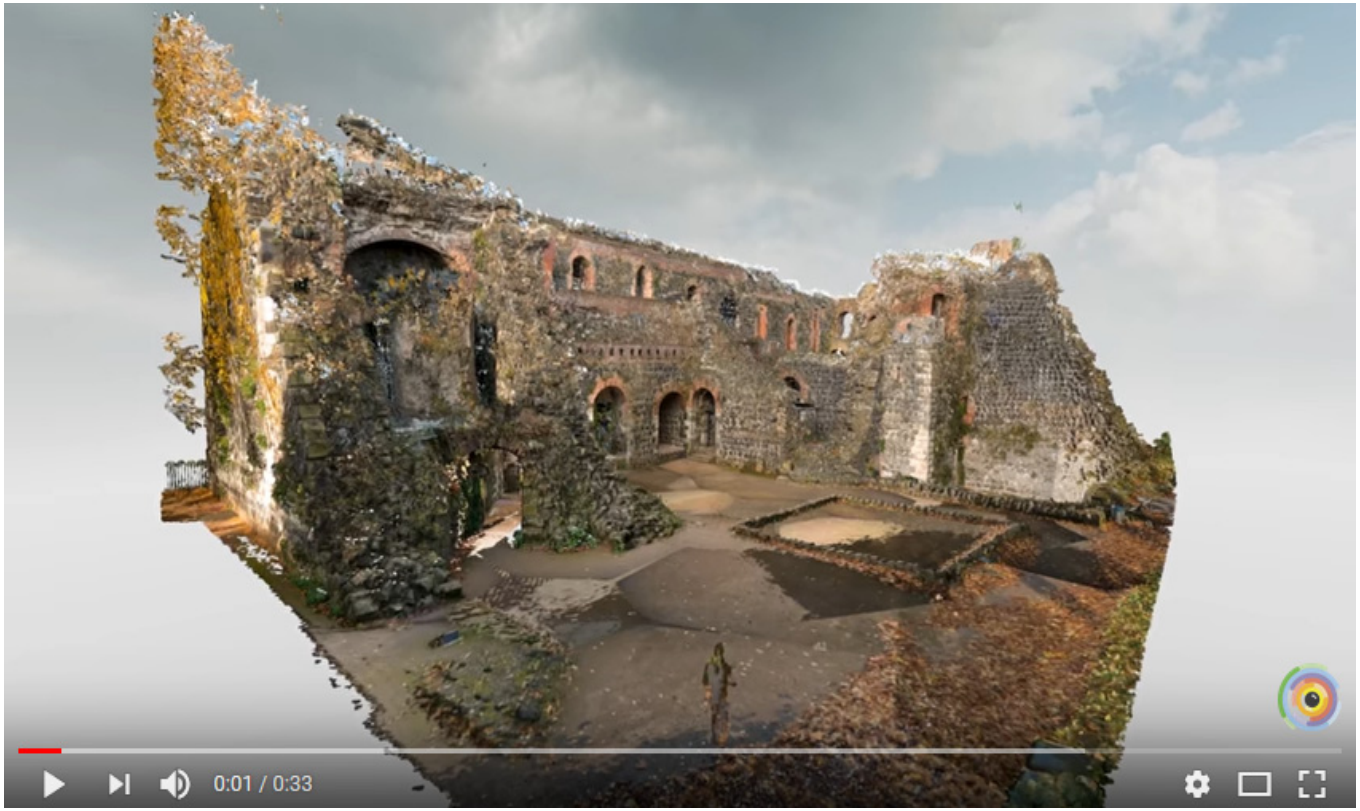
```
adminuser@linux-8djg:~> ssh -X vsc42544@login.hpc.ugent.be
Last login: Wed Nov 22 16:16:28 2017 from 91.177.4.215

STEVIN HPC-UGent infrastructure status on Thu, 23 Nov 2017 02:15:01

   cluster - full - free -  part - total - running - queued
             nodes  nodes   free   nodes    jobs      jobs
-----------------------------------------------------------------
  delcatty    157      0      0     159      N/A       N/A
    golett     96     45     53     196      N/A       N/A
    phanpy      9      0      7      16      N/A       N/A
    raichu     34      0     22      56      N/A       N/A
    swalot    110      0     18     128      N/A       N/A

For a full view of the current loads and queues see:
 http://hpc.ugent.be/clusterstate/
Updates on maintenance and unscheduled downtime can be found on
 https://www.vscentrum.be/en/user-portal/system-status

/usr/bin/xauth:  file /user/home/gent/vsc425/vsc42544/.Xauthority does not exist
[vsc42544@gligar03 Bremen]$ module load PCL/1.8.1-intel-2017b-Python-2.7.14
[vsc42544@gligar03 Bremen]$ pwd
/apps/gent/tutorials/machine_learning/clustering/Bremen
[vsc42544@gligar03 Bremen]$ ls -al
total 3431616
drwxr-xr-x 2 vsc40003 vsc40003       4096 Nov 22 22:39 .
drwxr-xr-x 5 vsc40003 vsc40003       4096 Nov 22 15:44 ..
-rw-r--r-- 1 vsc40003 vsc40003  382559971 Nov 22 22:39 bremenClustered.pcd
-rw-r--r-- 1 vsc40003 vsc40003 1302382632 Nov 22 14:07 bremen.h5.h5
-rw-r--r-- 1 vsc40003 vsc40003   72002416 Jan 13  2017 bremenSmall.h5.h5
[vsc42544@gligar03 Bremen]$ pcl_viewer bremenClustered.pcd
```

Point Cloud Viewer Example – Bremen Output Visualization



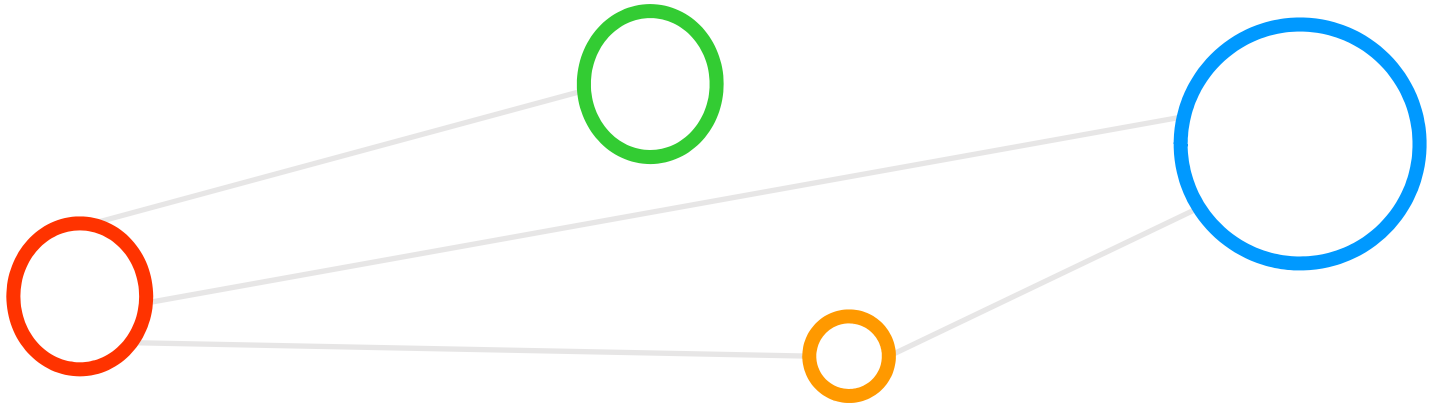
- **Use Strg and Mouse Wheel to Zoom and use numbers of keyboard for different visualizations**

[Video] Point Clouds



[10] Point Based Rendering of the Kaiserpfalz in Kaiserswerth, YouTube Video

Lecture Bibliography



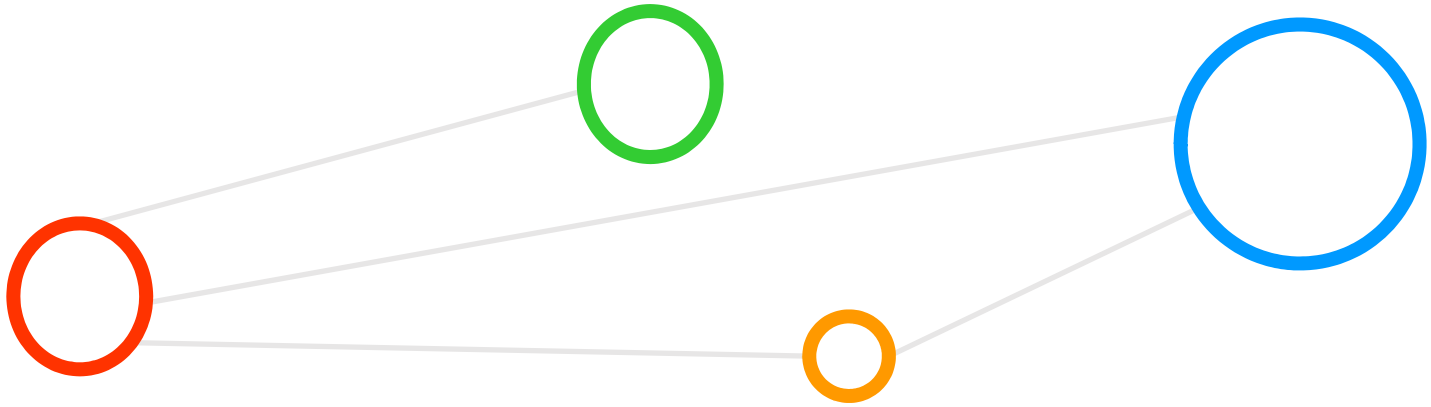
Lecture Bibliography (1)

- [1] Species Iris Group of North America Database, Online: http://www.signa.org
- [2] An Introduction to Statistical Learning with Applications in R, Online: http://www-bcf.usc.edu/~gareth/ISL/index.html
- [3] PANGAEA Data Collection, Data Publisher for Earth & Environmental Science, Online: http://www.pangaea.de/
- [4] Judy Qiu,'Harp: Collective Communication on Hadoop', 2014
- [5] Animation of the k-means algorithm using Matlab 2013, YouTube Video, Online: http://www.youtube.com/watch?v=5FmnJVv73fU
- [6] Ester, Martin, et al. "A density-based algorithm for discoveringclusters in large spatial databases with noise." Kdd. Vol. 96. 1996.
- [7] YouTube Video, 'CSCE 420 Communication Project – DBSCAN', Online: https://www.youtube.com/watch?v=5E097ZLE9Sg
- [8] YouTube Video ,'Point Based Rendering of the Aachen Cathedral', Online: https://www.youtube.com/watch?v=X_wyoroo4co
- [9] B2SHARE, 'HPDBSCAN Benchmark test files', Online: http://hdl.handle.net/11304/6eacaa76-c275-11e4-ac7e-860aa0063d1f
- [10] YouTube Video, 'Point Based Rendering of the Kaiserpfalz in Kaiserswerth', Online: https://www.youtube.com/watch?v=KvDb58YvlvQ
- [11] M.Goetz, M. Riedel et al.,'HPDBSCAN – Highly Parallel DBSCAN', Proceedings of MLHPC Workshop at Supercomputing 2015, Online: https://www.researchgate.net/publication/301463871_HPDBSCAN_highly_parallel_DBSCAN

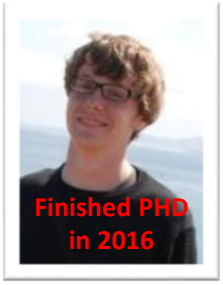
Lecture Bibliography (2)

- [12] 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
- [13] Michael Stephan,'Portable Parallel IO - 'Handling large datasets in heterogeneous parallel environments', Online: http://www.fz-juelich.de/SharedDocs/Downloads/IAS/JSC/EN/slides/parallelio-2014/parallel-io-hdf5.pdf?__blob=publicationFile
- [14] HDF Group, Online: http://www.hdfgroup.org/
- [15] Parallel NETCDF, Online: http://trac.mcs.anl.gov/projects/parallel-netcdf

Acknowledgements



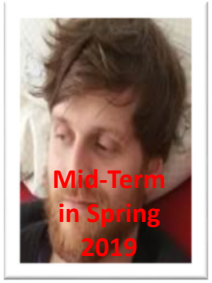
Acknowledgements – High Productivity Data Processing Research Group



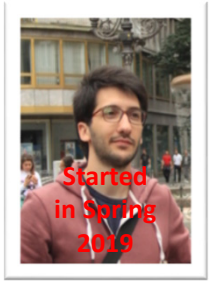
PD Dr.
G. Cavallaro

Senior PhD
Student A.S. Memon

Senior PhD
Student M.S. Memon



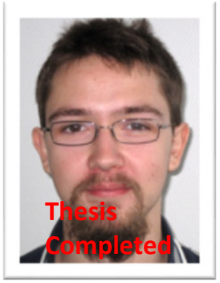
PhD Student
E. Erlingsson



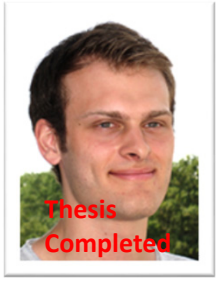
PhD Student
S. Bakarat

PhD Student
R. Sedona

Dr. M. Goetz
(now KIT)



MSc M.
Richerzhagen
(now other division)



MSc
P. Glock
(now INM-1)

MSc
C. Bodenstein
(now
Soccerwatch.tv)

MSc Student
G.S. Guðmundsson
(Landsverkjun)

This research group has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 763558 (DEEP-EST EU Project)

Data
Science
Big
Research
science
methods
simulations
Computational
E.g
Centre
big
Computing
processing
technologies
computing
analysis
computational
HPC
Juelich
Scientific
European
TB
Supercomputing
images
performance
datasets
data-intensive
Climate
modelling
structures
access
Hardware
Technologies
Earth
Health
Energy
Simulation
systems
day
NASA
Germany
Understanding
DLCLs
Provide
System
JSC
research
e.g
acquisition
DLCL
Project
databases
Big Data
Enable
Modelling
Environment
Infrastructure
storage
ENES
EUDAT
IPCC
Common
Methods
Using
Work
Service
Concrete
Matters
Landscape
Services
network
management
distributed
large-scale
cores
current
Structure
General
project
directory
Processing
Resources
stored
Summary
Large
Level
use
codes
formats
brain
increasing
via
physical
MapReduce
create
GB
analyze
external
levels
often
system
hundreds
Nutshell
compute
AR5
Test
Cross-Disciplinary
like
year
forms
cross-disciplinary
resources
Policy-based
measurement
services
new
funding
fusion
within
concepts
device
Analysis
Computer
expertise
Approaches
construction
Fusion
disciplines
results
nodes
centers
PB
High
community
Key
scientific
important
different
files
basis
DKRZ Use
etc
EU
SMAQ
international
engineering
References
model