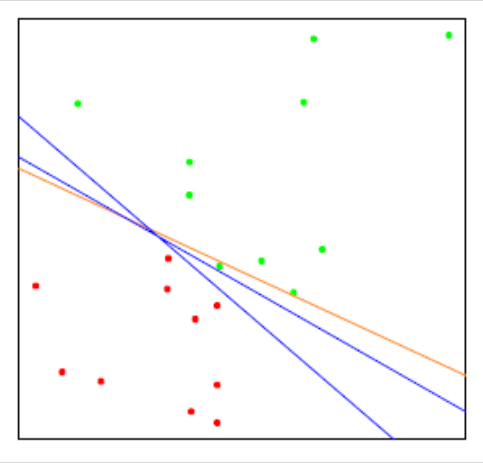


Parallel & Scalable Data Analysis

Introduction to Machine Learning Algorithms

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

LECTURE 2

Unsupervised Clustering & Applications

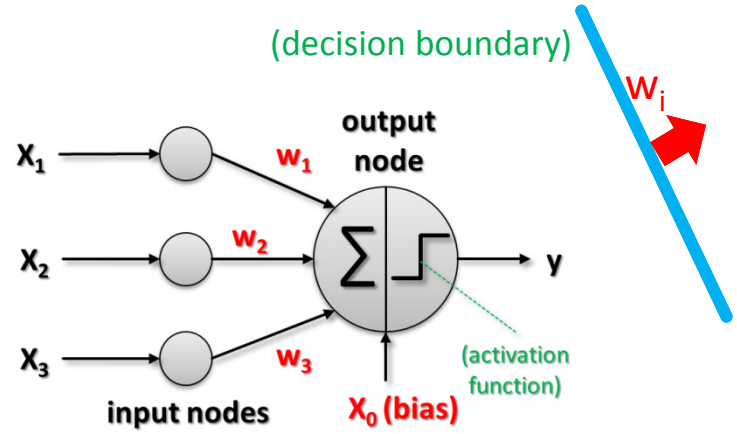
November 23th, 2017
Ghent, Belgium

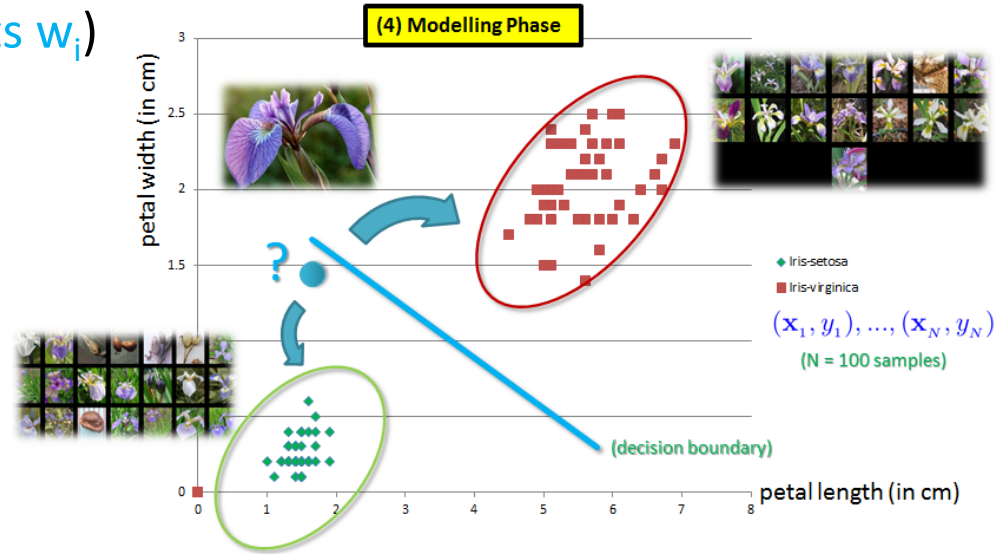
UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

JÜLICH
FORSCHUNGSZENTRUM

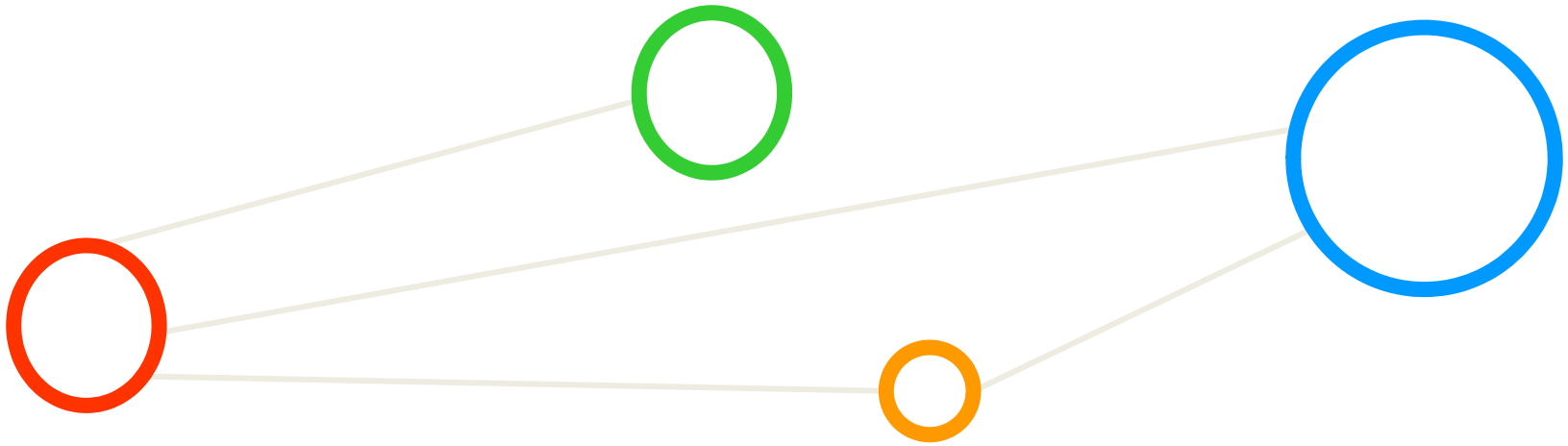
Review of Lecture 1

- Machine Learning Prerequisites
  1. Some pattern exists
  2. No exact mathematical formula
  3. Data exists
- Linearly seperable dataset Iris
  - Perceptron Model as hypothesis (simplest linear learning model, linearity in learned weights w_i)
  - Understood Perceptron Learning Algorithm (PLA)
- Learning Approaches
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning





Outline



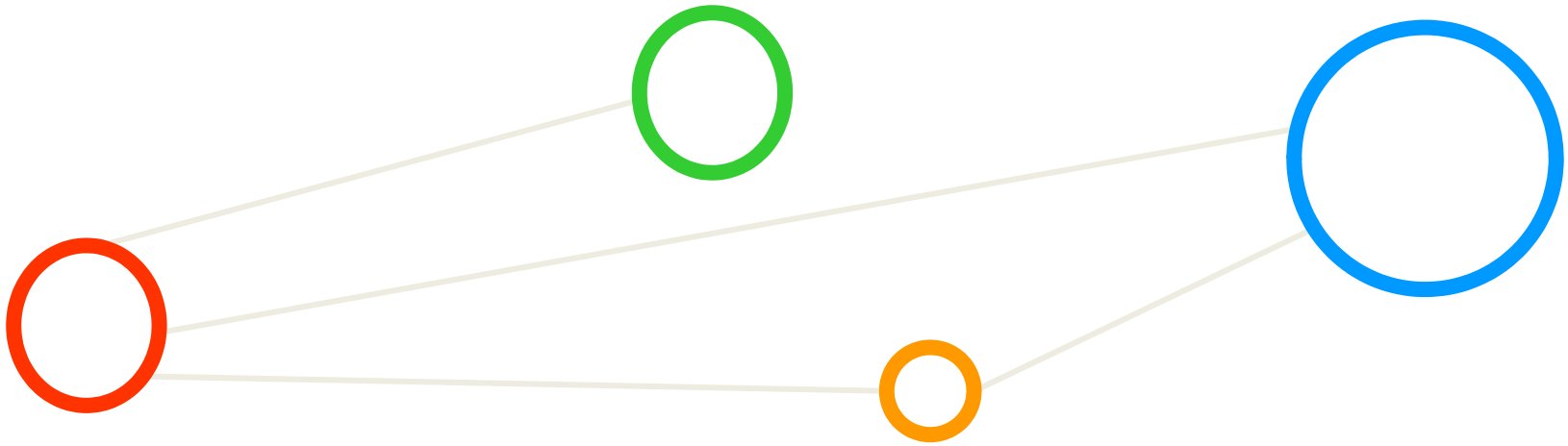
Outline of the Course

1. Machine Learning Fundamentals
2. Unsupervised Clustering and Applications
3. Supervised Classification and Applications
4. Classification Challenges and Solutions
5. Regularization and Support Vector Machines
6. Validation and Parallelization Benefits

Outline

- Unsupervised Clustering
  - Clustering Methods and Approaches
  - K-Means & K-Median Clustering Algorithms
  - DBSCAN Clustering Algorithm
- Point Cloud Applications
  - Introduction to Application Domain
  - Dataset Examples
  - Bremen Datasets & Locations
- Parallel Computing & Tools
  - High Performance Computing (HPC)
  - GOLETT Supercomputer for Tutorial
  - Parallel & Scalable HPDBSCAN on GOLETT
  - Batch System Usage on GOLETT
  - Apply HPDBSCAN to Point Cloud Data

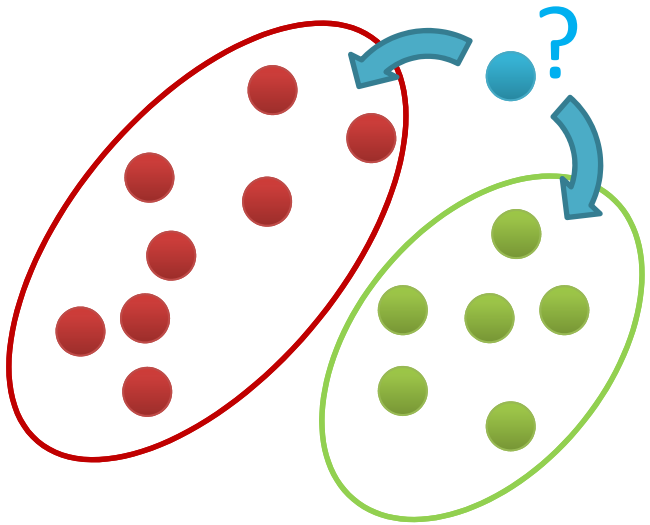
Unsupervised Clustering



Methods Overview

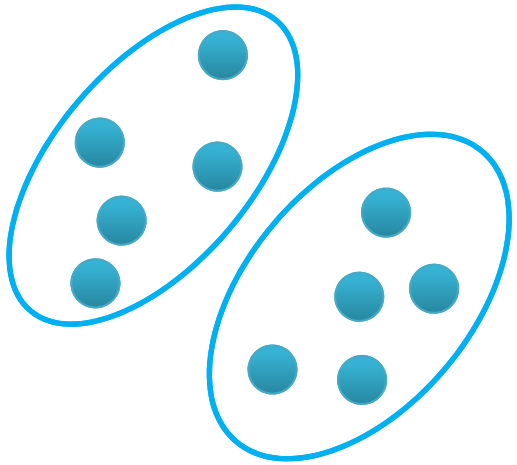
- **Statistical data mining methods can be roughly categorized in classification, clustering, or regression augmented with various techniques for data exploration, selection, or reduction**

Classification



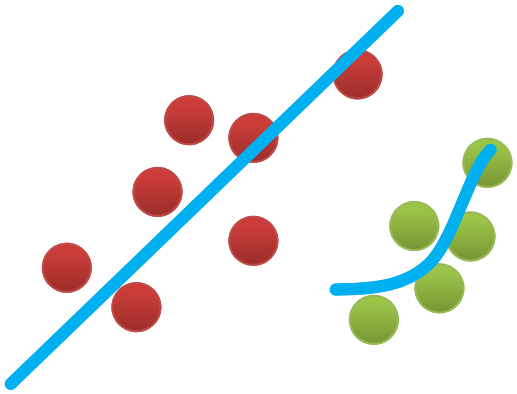
- Groups of data exist
- New data classified to existing groups

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

What means Learning – Revisited

- **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**

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

Learning Approaches – Unsupervised Learning – Revisited

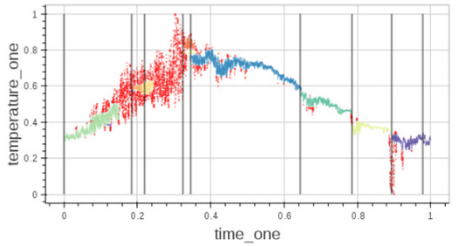
- Each observation of the predictor measurement(s) has no associated response measurement:
  - Input $\mathbf{x} = x_1, \ldots, x_d$
  - No output
  - Data $(\mathbf{x}_1), \ldots, (\mathbf{x}_N)$
- Goal: Seek to understand relationships between the observations
  - Clustering analysis: check whether 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, ---]**

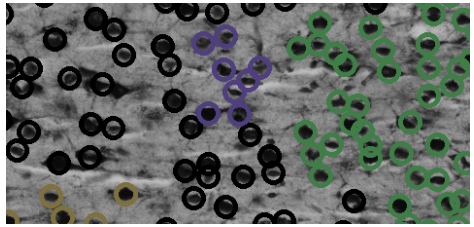
[1] An Introduction to Statistical Learning

Learning Approaches – Unsupervised Learning Use Cases

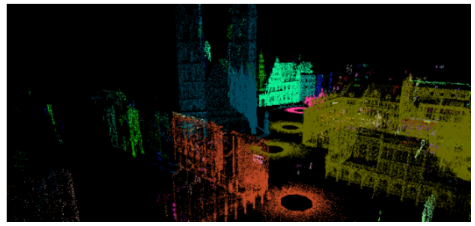
- Earth Science Data (PANGAEA, cf. Lecture 1)
  - Automatic quality control and event detection
  - Collaboration with the University of Gothenburg
  - Koljoefjords Sweden – Detect water mixing events



- 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



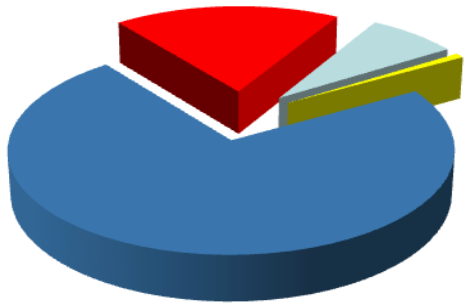
- Point Cloud Data
  - Analysis of point cloud datasets of various sizes
  - 3D/4D LIDAR scans of territories (cities, ruins, etc.)
  - Filter noise and reconstruct objects



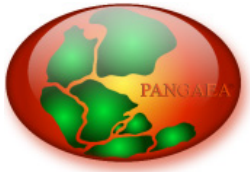
➢ **This clustering lecture uses a point cloud dataset of the city of Bremen as one concrete example**

Unsupervised Learning – Earth Science Data Example

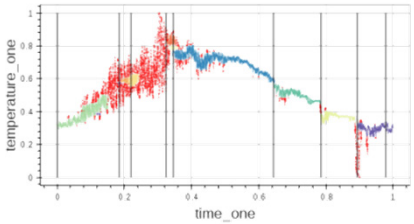
- Earth Science Data Repository
  - Time series measurements (e.g. salinity)
  - Millions to billions of data items/locations
  - Less capacity of experts to analyse data



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

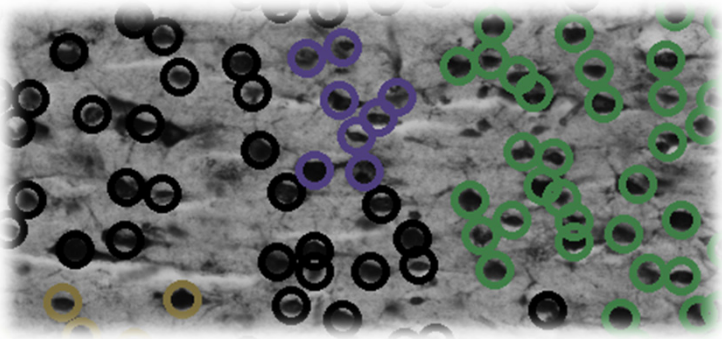


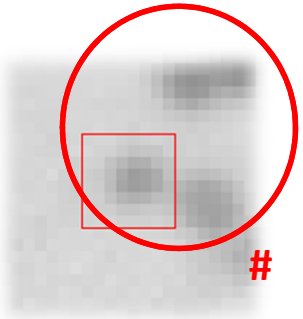
- Selected Scientific Case
  - Data from Koljöfjords in Sweden (Skagerrak)
  - Each measurement small data, but whole sets are ‘big data‘
  - Automated water mixing event detection & quality control (e.g. biofouling)
  - Verification through domain experts

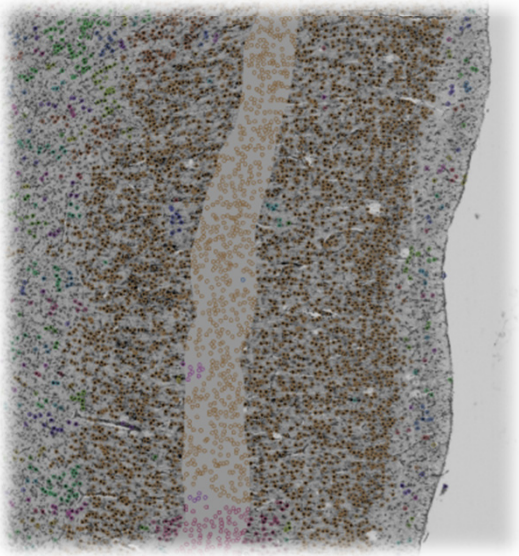


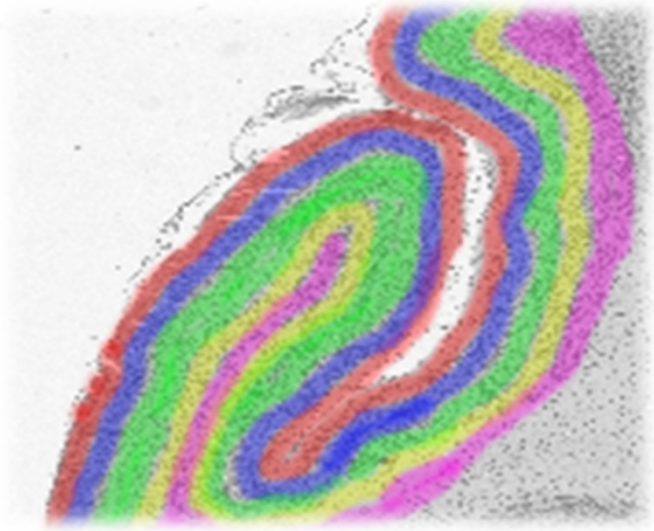
[2] PANGAEA data collection

Unsupervised Learning – Human Brain Data Example





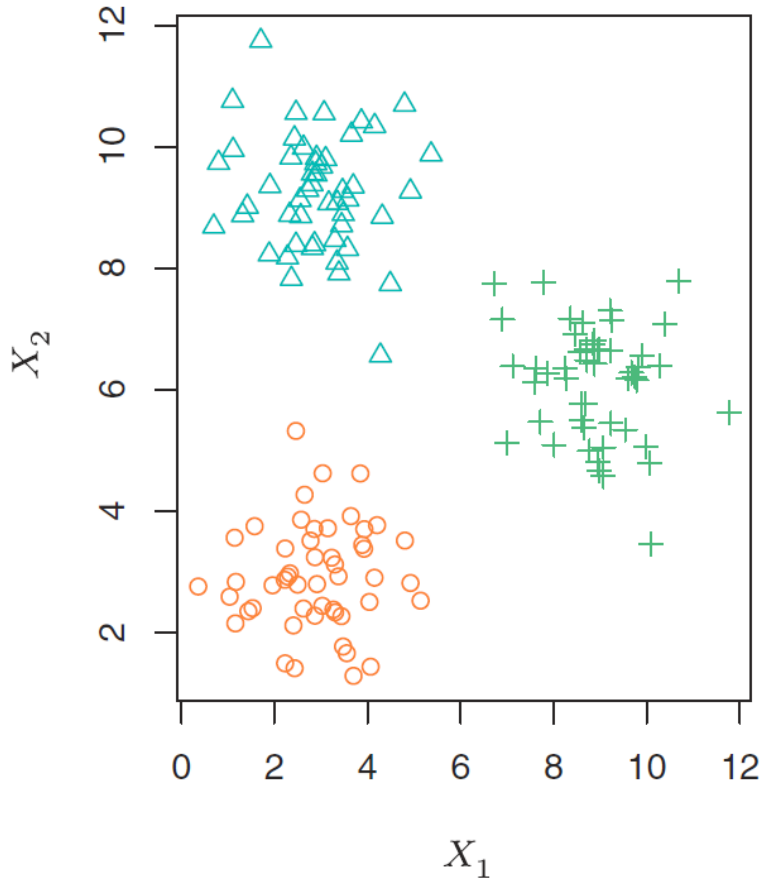


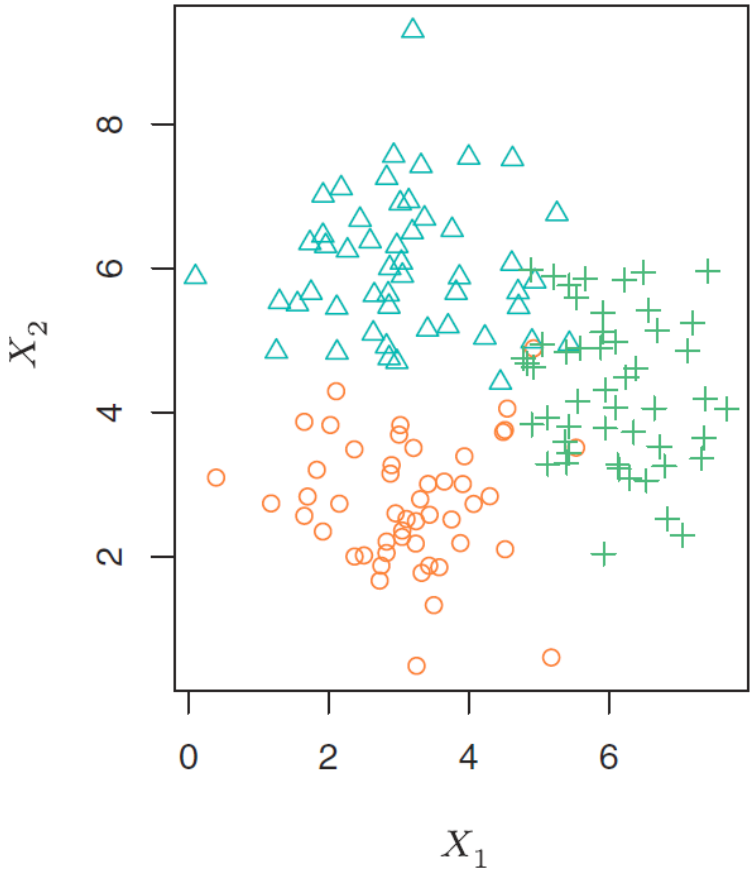


➢ **Research activities jointly with T. Dickscheid et al. (Juelich Institute of Neuroscience & Medicine)**

Learning Approaches – Unsupervised Learning Challenges

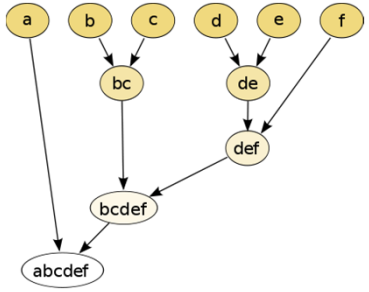
- Practice: The number of clusters can be ambiguities



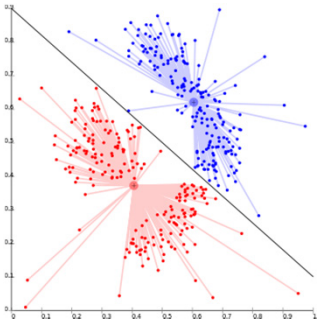


[1] An Introduction to Statistical Learning

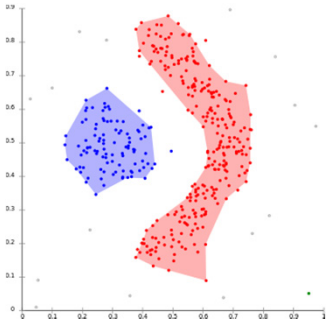
Unsupervised Learning – Different Clustering Approaches



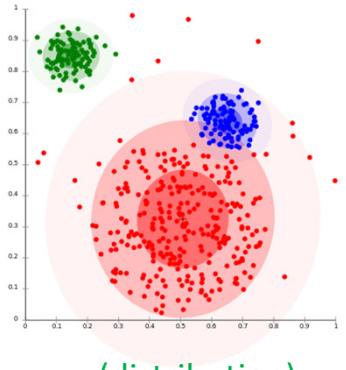
(hierarchical)



(centroid)



(density)



(distribution)

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

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

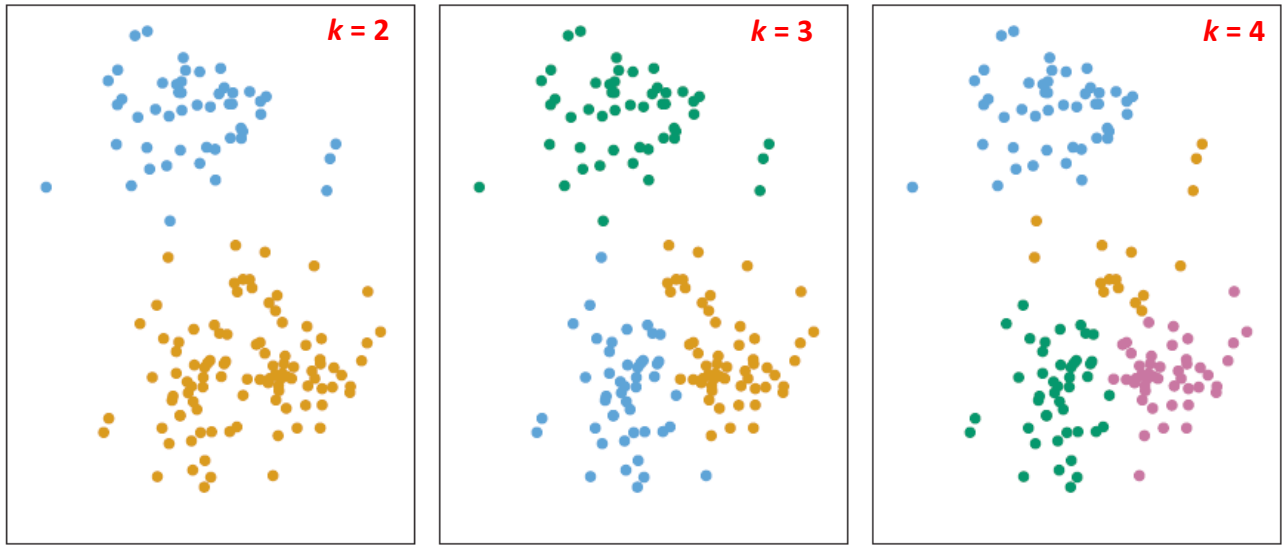
$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

(ruler distance)

  - n-dimensional Euclidean space:
    A space where points are vectors of n real numbers

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*



[1] An Introduction to Statistical Learning

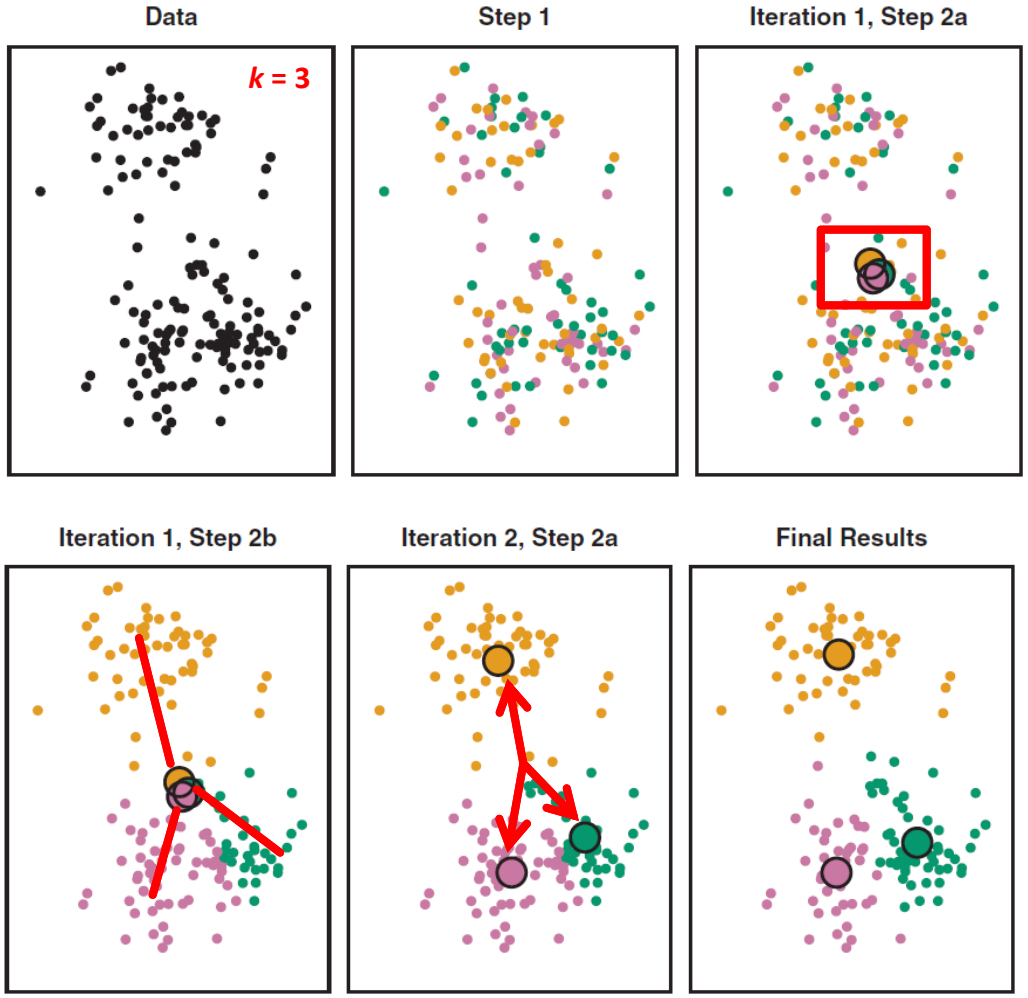
Clustering Methods – K-Means Algorithm

0. 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

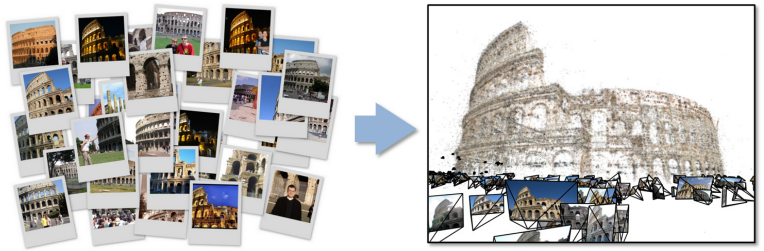


[1] An Introduction to Statistical Learning

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]

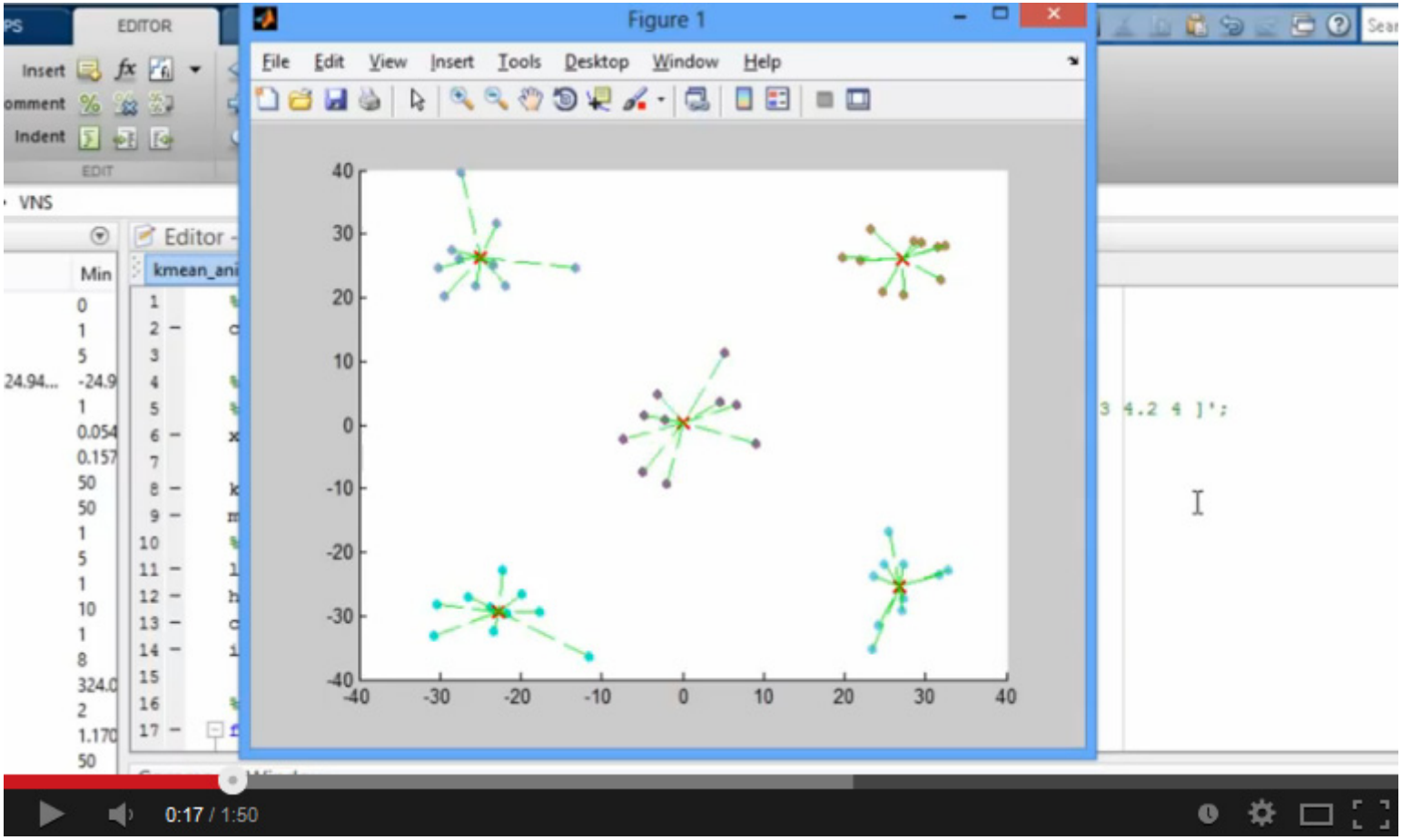
Clustering Methods – K-Means Usage

- 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;



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

[Video] K-Means Clustering



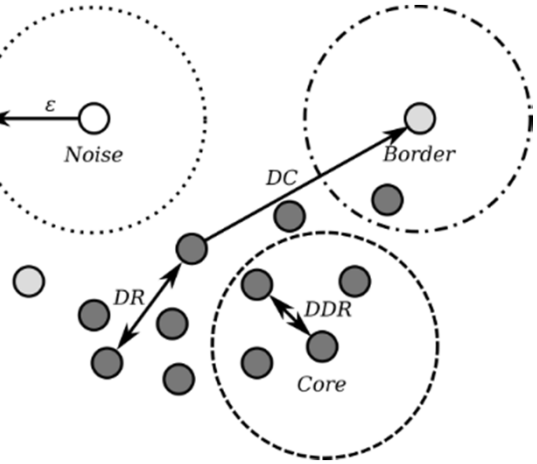
[4] Animation of the k-means clustering algorithm, YouTube Video

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 *[5] Ester et al.*
  - 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*)



(MinPoints = 4)

- 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*

(DR = Density Reachable)

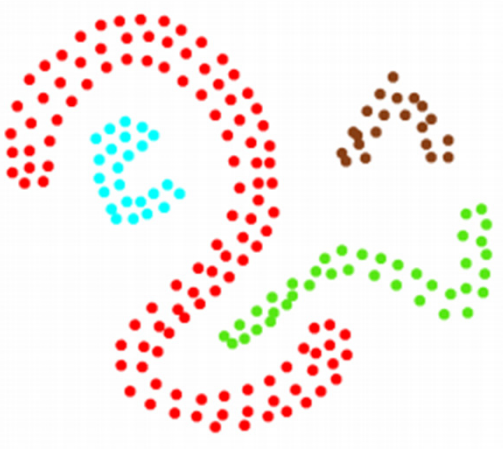
(DDR = Directly Density Reachable)

(DC = Density Connected)

DBSCAN Algorithm – Non-Trivial Example

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

Unclustered Data

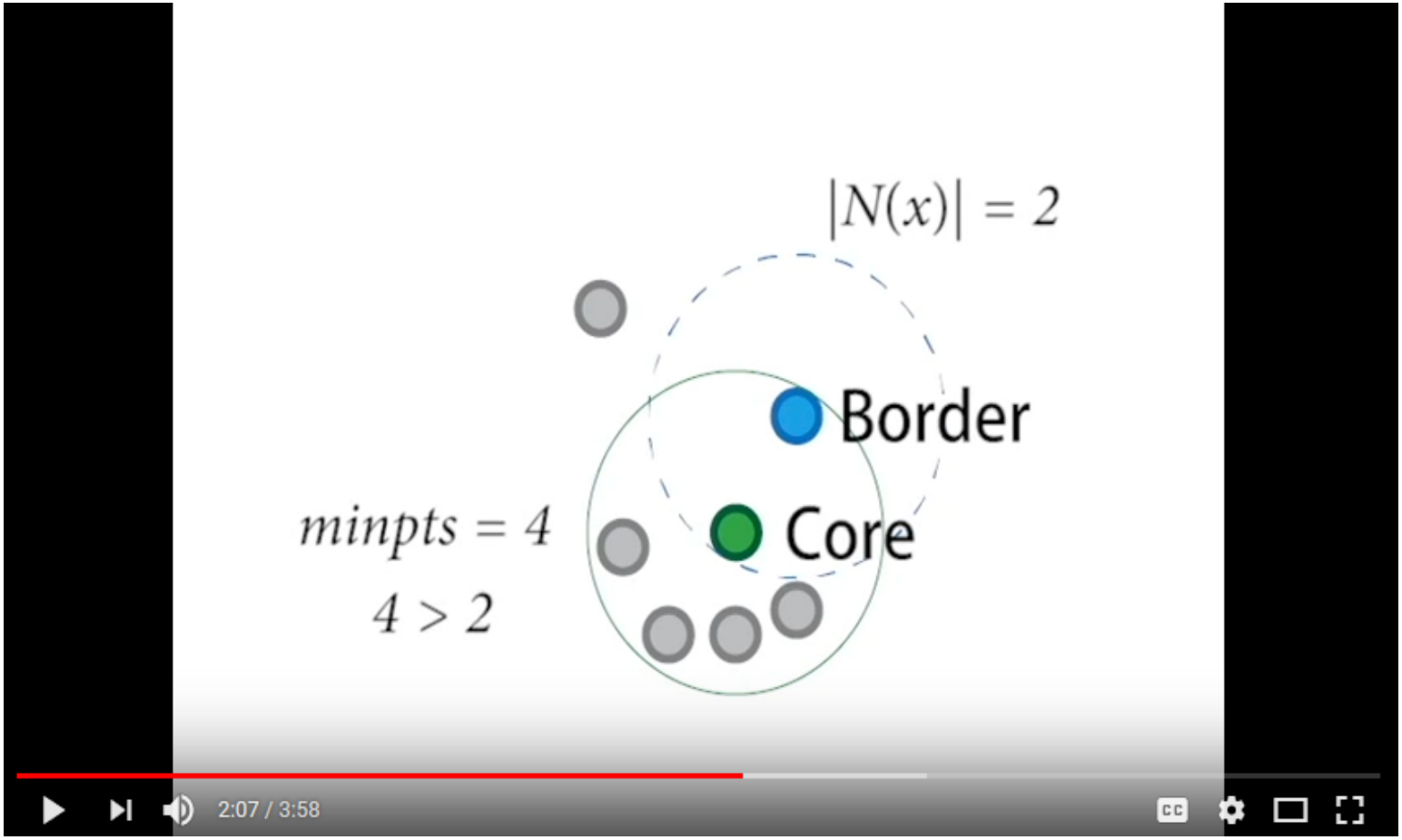


Clustered Data

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

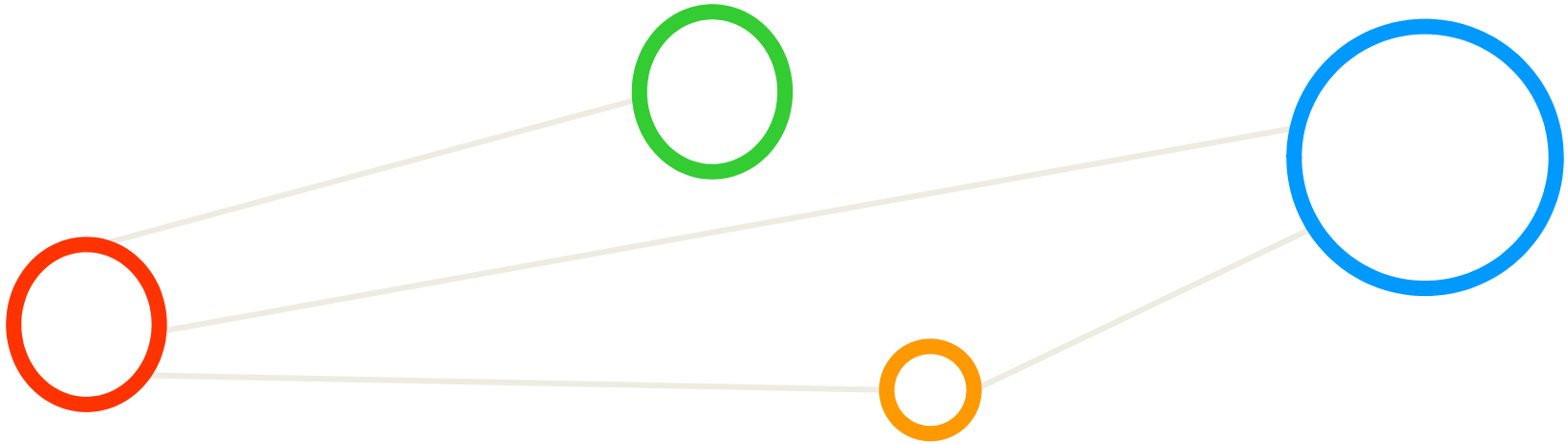
Exercises

[Video] DBSCAN Clustering



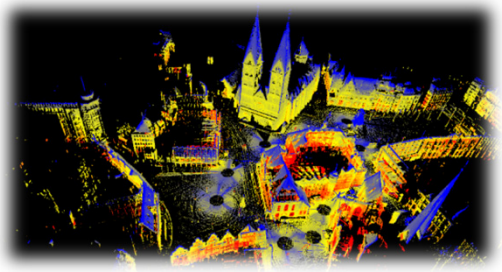
[6] DBSCAN, YouTube Video

Point Cloud Applications

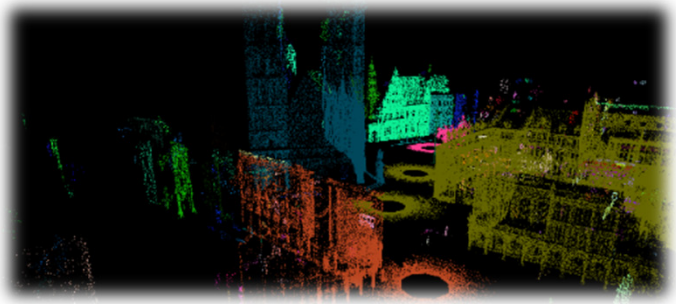


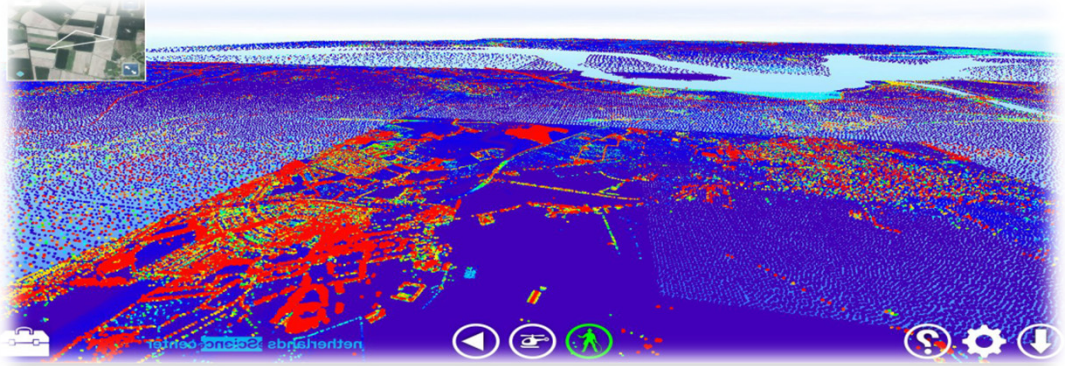
Point Cloud Applications

- ‘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





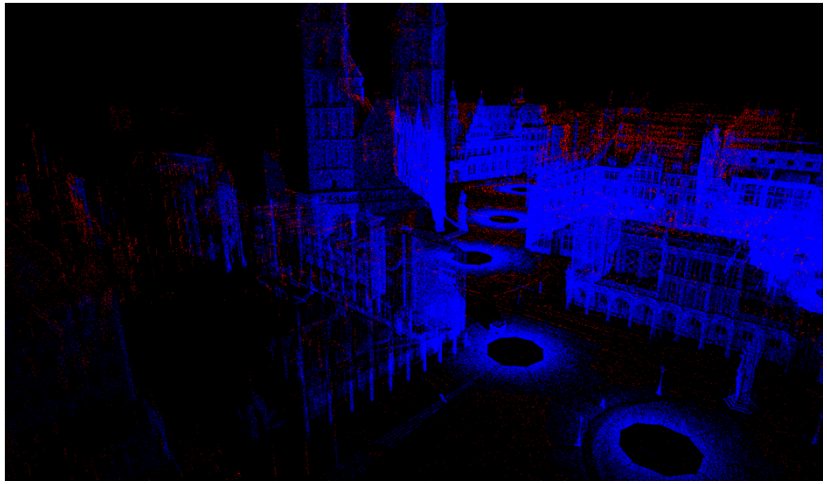
Point Cloud Application Example – Within Buildings

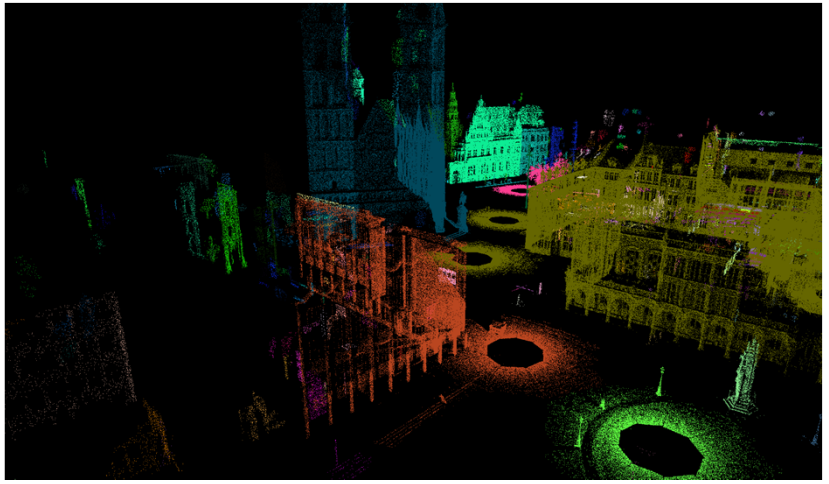
- Point based rendering example
  - Aachen Cathedral based on 3D laser scans and photos
  - Points are rendered as textured and blended splats
  - Visualisation can run in real-time on a desktop PC showing 6 million splats based of a 120 million point laser scan

[7] Aachen Cathedral Point Cloud Rendering, YouTube Video

Bremen Dataset & Locations

- Different clusterings of the inner city of Bremen
  - Using smart visualizations of the point cloud library (PCL)





```
[vsc42544@gligar01 Bremen]$ pwd
/apps/gent/tutorials/machine_learning/clustering/Bremen
[vsc42544@gligar01 Bremen]$ ls -al
total 2684416
drwxr-xr-x 2 vsc40003 vsc40003       4096 Nov 22 15:42 .
drwxr-xr-x 5 vsc40003 vsc40003       4096 Nov 22 15:44 ..
-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
```

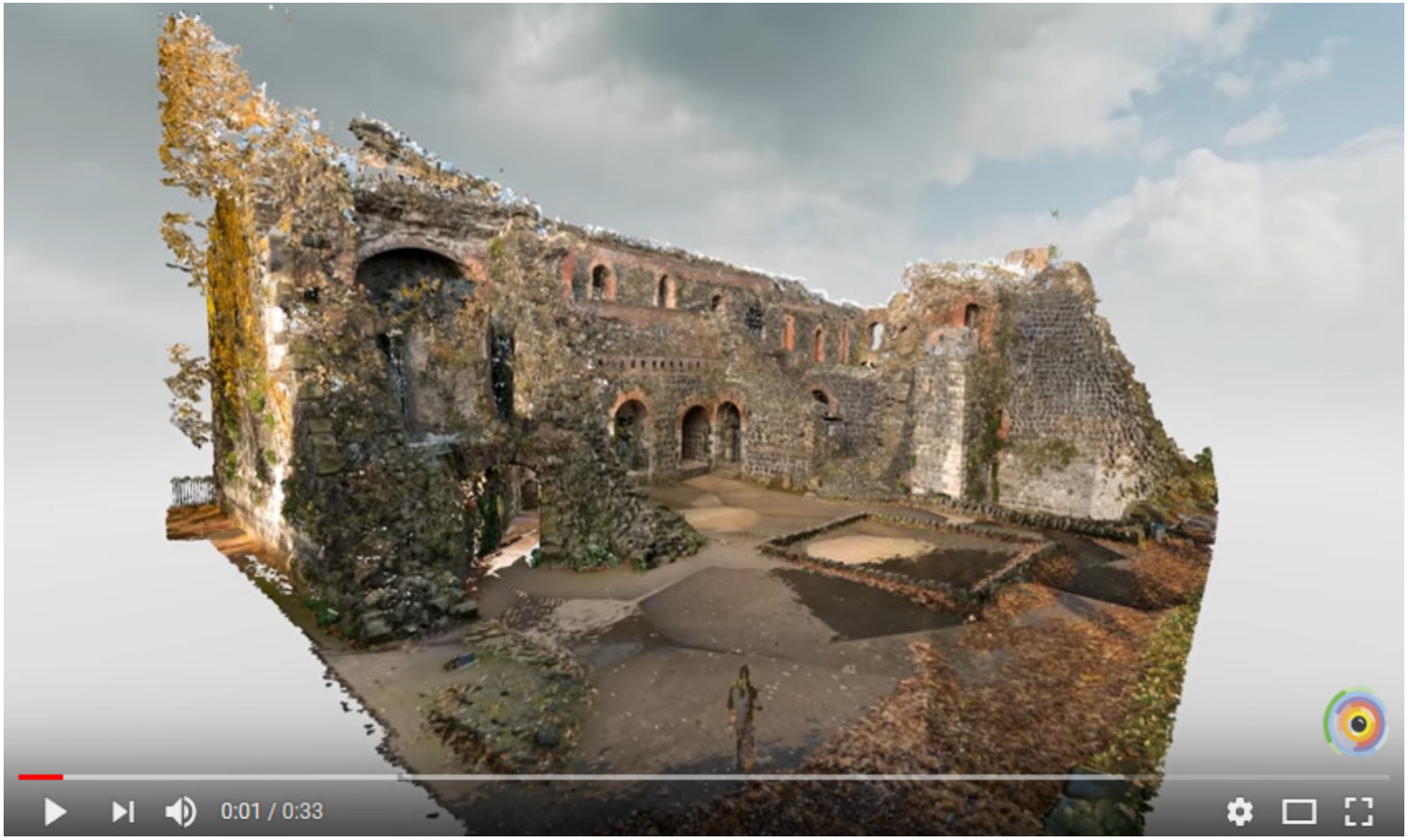
- **The Bremen Dataset is encoded in the HDF5 format**

[15] Bremen Dataset

B2SHARE
Store and Share Research Data

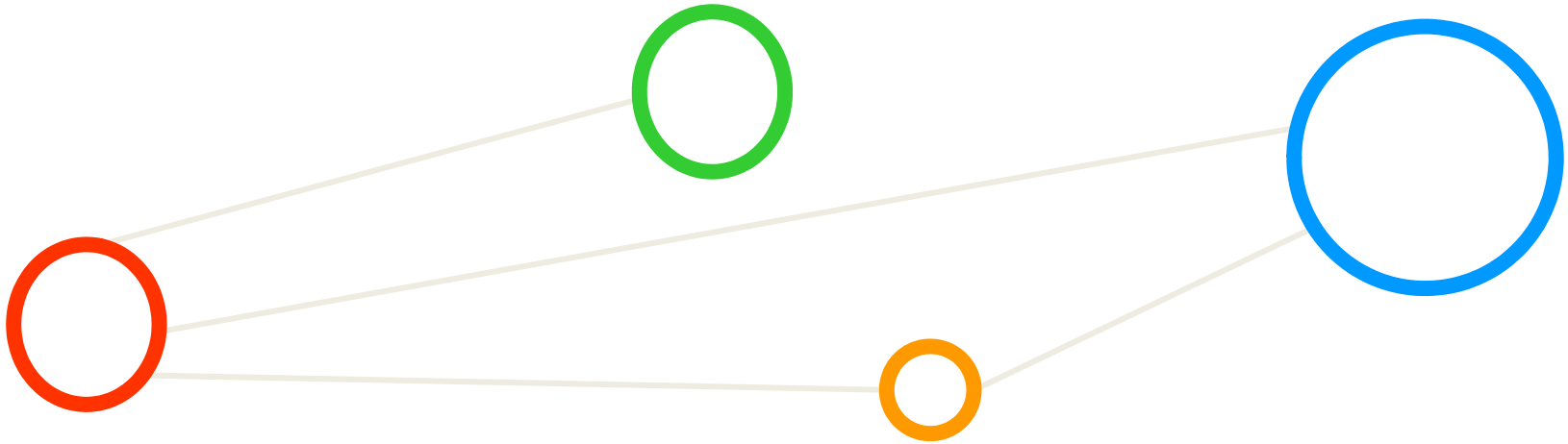
Exercises

[Video] Point Clouds



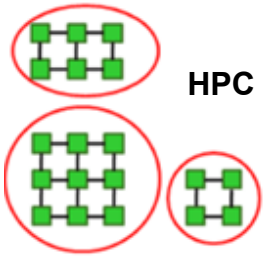
[8] Point Based Rendering of the Kaiserpfalz in Kaiserswerth, YouTube Video

Parallel Computing



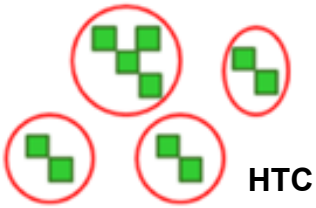
Understanding HPC vs. HTC

- **High Performance Computing (HPC) is based on computing resources that enable the efficient use of parallel computing techniques through specific support with dedicated hardware such as high performance cpu/core interconnections.**



focus in the next slides

- **High Throughput Computing (HTC) is based on commonly available computing resources such as commodity PCs and small clusters that enable the execution of 'farming jobs' without providing a high performance interconnection between the cpu/cores.**



Ugent Tier-2 Clusters

- Using Parallel Computing
  - Compiled from open source
  - Requires MPI library
  - Intended to be used by High Performance Computing system (i.e. good interconnects)

- Job runs
  - Use of job scripts
  - Depend on scheduler

- **Use our ssh keys to get an access and use reservation**
- **Put the private key into your ./ssh directory (UNIX)**
- **Use the private key with your putty tool (Windows)**

Vlaams Supercomputer Centrum

[14] UGent Tier-2 Clusters

UGent Tier-2 Clusters – GOLETT in the Tutorial

	#nodes	CPU	Mem/node	Diskspace/node	Network
Raichu	64	2 x 8-core Intel E5-2670 (Sandy Bridge @ 2.6 GHz)	32 GB	400 GB	GbE
Delcatty	160	2 x 8-core Intel E5-2670 (Sandy Bridge @ 2.6 GHz)	64 GB	400 GB	FDR InfiniBand
Phanpy	16	2 x 12-core Intel E5-2680v3 (Haswell-EP @ 2.5 GHz)	512 GB	3x 400 GB (SSD, striped)	FDR InfiniBand
Golett	200	2 x 12-core Intel E5-2680v3 (Haswell-EP @ 2.5 GHz)	64 GB	500 GB	FDR-10 InfiniBand
Swalot	128	2 x 10-core Intel E5-2660v3 (Haswell-EP @ 2.6 GHz)	128 GB	1 TB	FDR InfiniBand

Vlaams Supercomputer Centrum

[14] UGent Tier-2 Clusters

UGent Tier-2 Clusters – Login & Module Swap Cluster/golett

```
adminuser@linux-8djg:~> ssh vsc42544@login.hpc.ugent.be
Last login: Wed Nov 22 17:15:00 2017 from pool-216-7-zam606.vpn.kfa-juelich.de

STEVIN HPC-UGent infrastructure status on Wed, 22 Nov 2017 22:15:01

   cluster - full - free -  part - total - running - queued
             nodes  nodes   free   nodes    jobs      jobs
-----------------------------------------------------------------
  delcatty    153      0      4     159      N/A       N/A
    golett    102     35     57     196      N/A       N/A
    phanpy     11      0      5      16      N/A       N/A
    raichu     56      0      0      56      N/A       N/A
    swalot    107      0     21     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

[vsc42544@gligar01 ~]$ module swap cluster/golett

The following have been reloaded with a version change:
  1) cluster/delcatty => cluster/golett
```

Vlaams Supercomputer Centrum

[14] UGent Tier-2 Clusters

Exercises

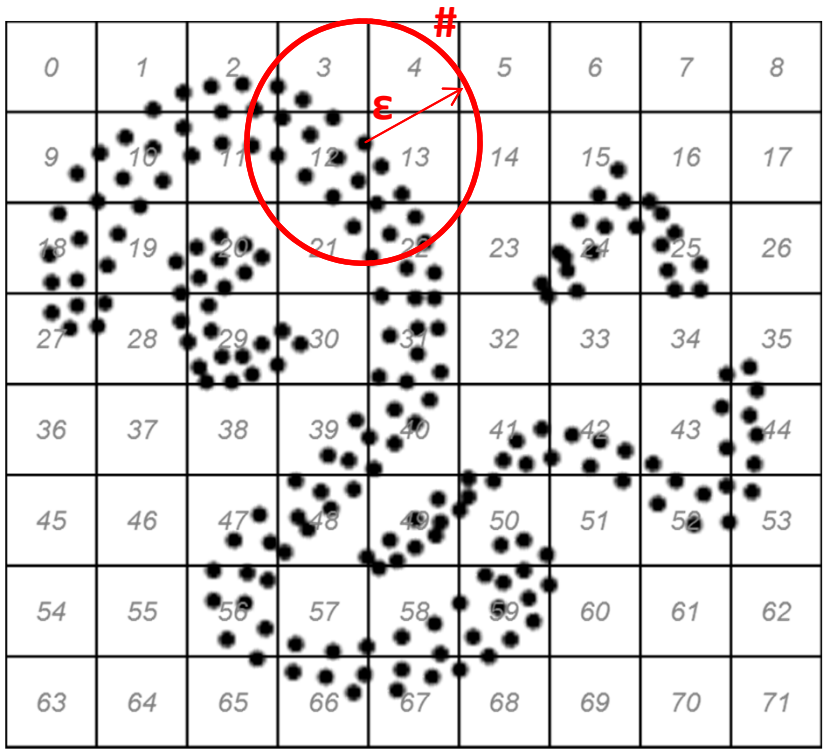
Review of Parallel DBSCAN Implementations

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

[9] 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

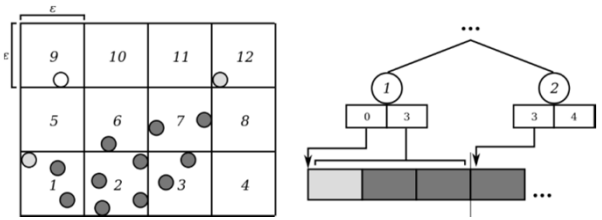
- Parallelization Strategy
  - Smart ‘Big Data‘ Preprocessing into Spatial Cells (‘indexed‘)
  - OpenMP standalone
  - MPI (+ optional OpenMP hybrid)
- Preprocessing Step
  - Spatial indexing and redistribution according to the point localities
  - Data density based chunking of computations
- Computational Optimizations
  - Caching of point neighborhood searches
  - Cluster merging based on comparisons instead of zone reclustering

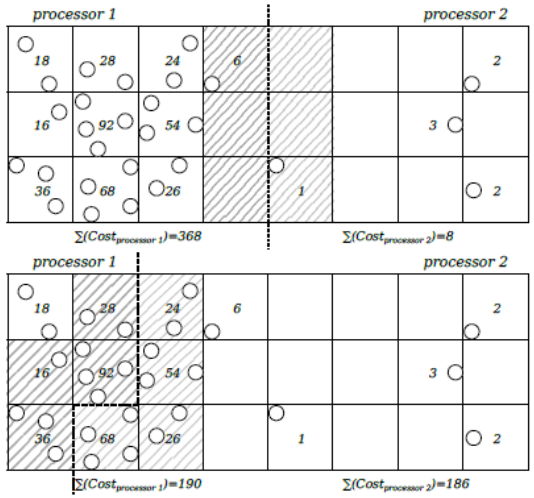


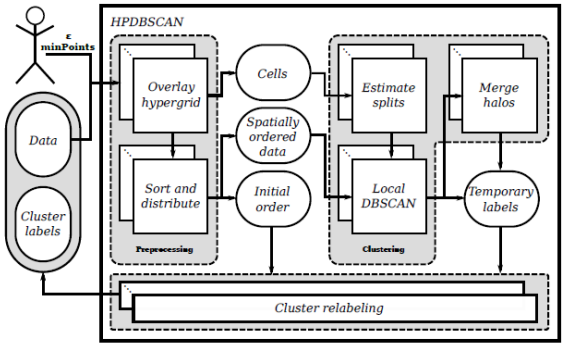
[10] M.Goetz, M. Riedel et al., ‘HPDBSCAN – Highly Parallel DBSCAN’, MLHPC Workshop at Supercomputing 2015

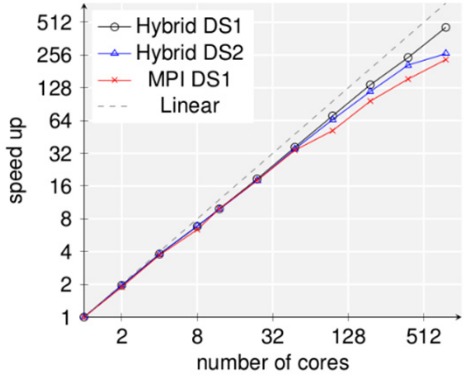
HPDBSCAN – Smart Domain Decomposition Example

- Parallelization Strategy
  - Chunk data space equally
  - Overlay with hypergrid
  - Apply cost heuristic
  - Redistribute points (data locality)
  - Execute DBSCAN locally
  - Merge clusters at chunk edges
  - Restore initial order
- Data organization
  - Use of HDF5
  - Cluster Id / noise ID stored in HDF5 file



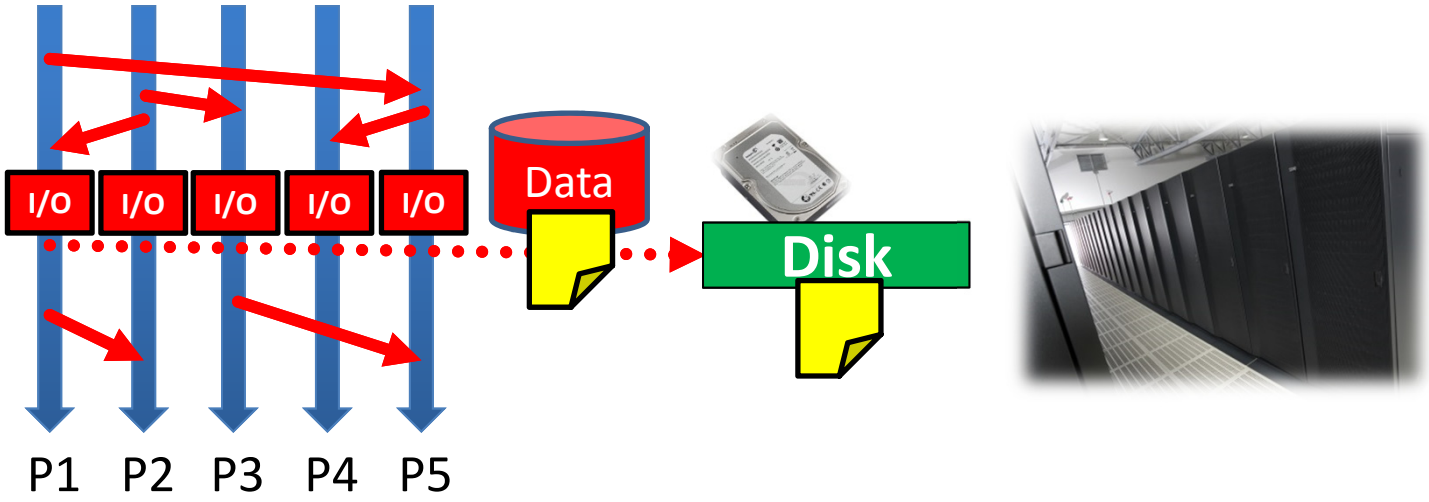






[10] M.Goetz, M. Riedel et al., 'HPDBSCAN – Highly Parallel DBSCAN', MLHPC Workshop at Supercomputing 2015

HPDBSCAN 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

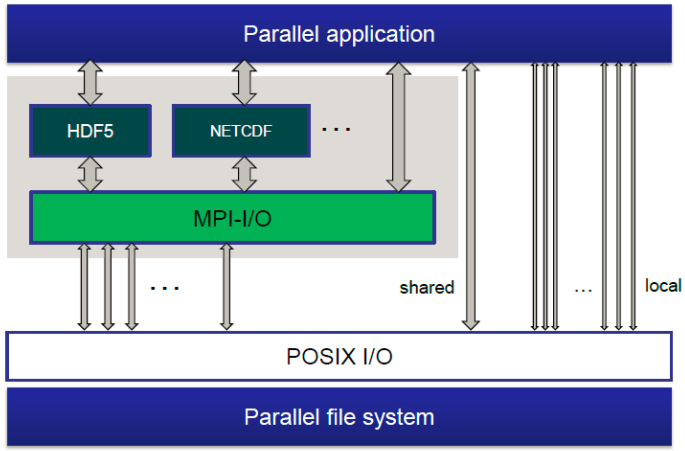
- **Parallel I/O: shared file means that processes can access their ‘own portion’ of a single file**
- **Parallel I/O with a shared file like MPI/IO is a scalable and even standardized solution**

HPDBSCAN 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**

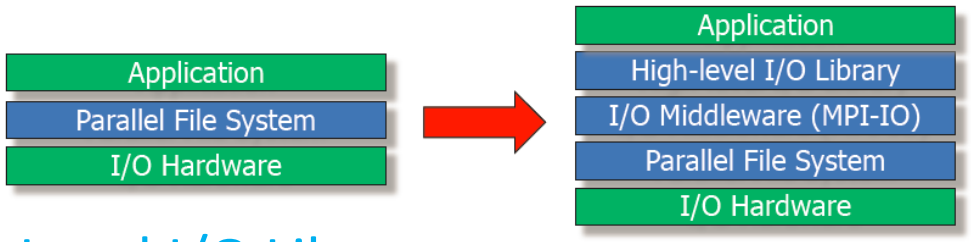
[16] HDF Group ***[17] Parallel NETCDF***

- 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**



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

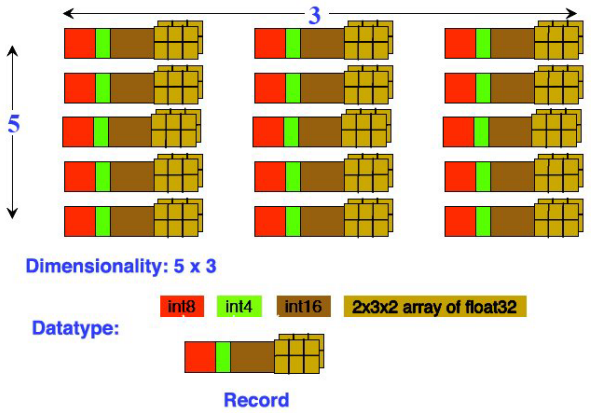
- High-Level I/O Library
  - Maps application abstractions to a structured portable file format
  - E.g. HDF-5, Parallel NetCDF
- I/O Middleware
  - E.g. MPI I/O
  - Deals with organizing access by many processes
- Parallel Filesystem
  - Maintains logical space and provides efficient access to data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

Hierarchical Data Format (HDF)

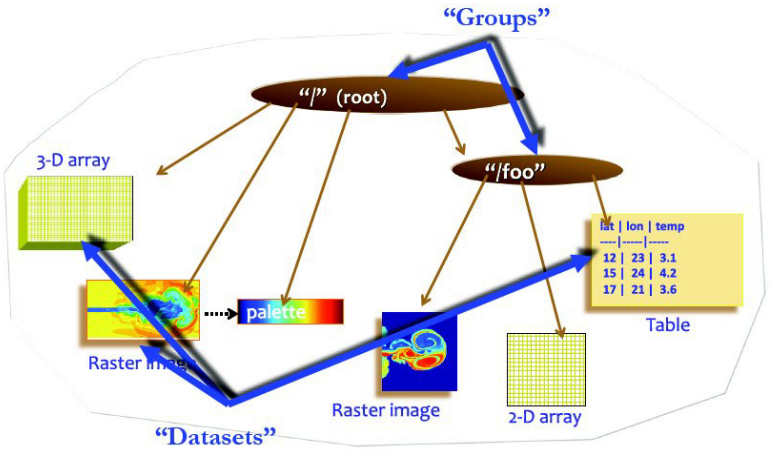
- **HDF is a technology suite that enables the work with extremely large and complex data collections**

[19] 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)

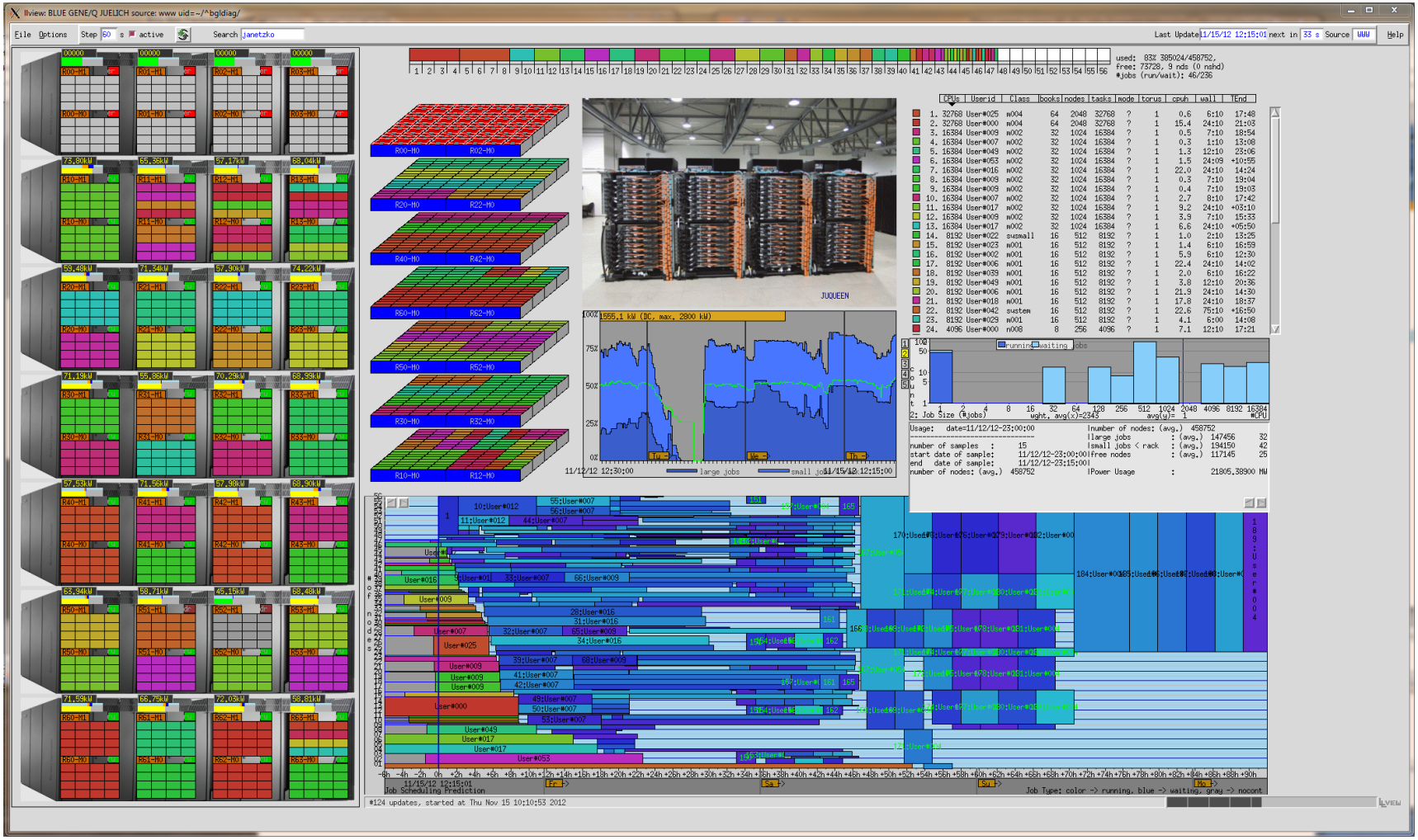


‘HDF5 file is a container’ to organize data objects



Cluster Golett – Why using a Scheduler in the Tutorial?

- Example: JUQUEEN concurrent usage shown in LLview



UGent Tier-2 Cluster Golett – HPDBSCAN Job Script

```
#!/bin/bash
#PBS -l walltime=1:0:0
#PBS -l nodes=1:ppn=all

module load HPDBSCAN/20171110-foss-2017b
module load vsc-mympirun

export WORKDIR=$VSC_SCRATCH/$PBS_JOBID
mkdir -p $WORKDIR
cd $WORKDIR

cp /apps/gent/tutorials/machine_learning/clustering/Bremen/bremenSmall.h5.h5 .

# by default, mympirun will use all available cores
# use --hybrid to only use a certain number of cores (per workernode)
mympirun --hybrid 6 dbscan -e 300 -m 100 -t 12 bremenSmall.h5.h5

echo "Results available in $WORKDIR"
```

- **Job submit using command: qsub <jobscript>**
- **Remember your <jobid> that is returned from the qsub command**
- **Show status of the job then with: qstat**

(parameters of DBSCAN and file to be clustered)

```
[vsc42544@gligar01 ~]$ qsub HPDBSCAN_example.sh
1173910.master19.golett.gent.vsc
[vsc42544@gligar01 ~]$ qstat
Job ID                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
1173910.master19.golett.gent. ...AN_example.sh vsc42544              0 Q short
```

Vlaams Supercomputer Centrum

- **Add for the tutorial reservation in the top of the file please this – check email with reservation!**

UGent Tier-2 Cluster Golett – HPDBSCAN Check Outcome

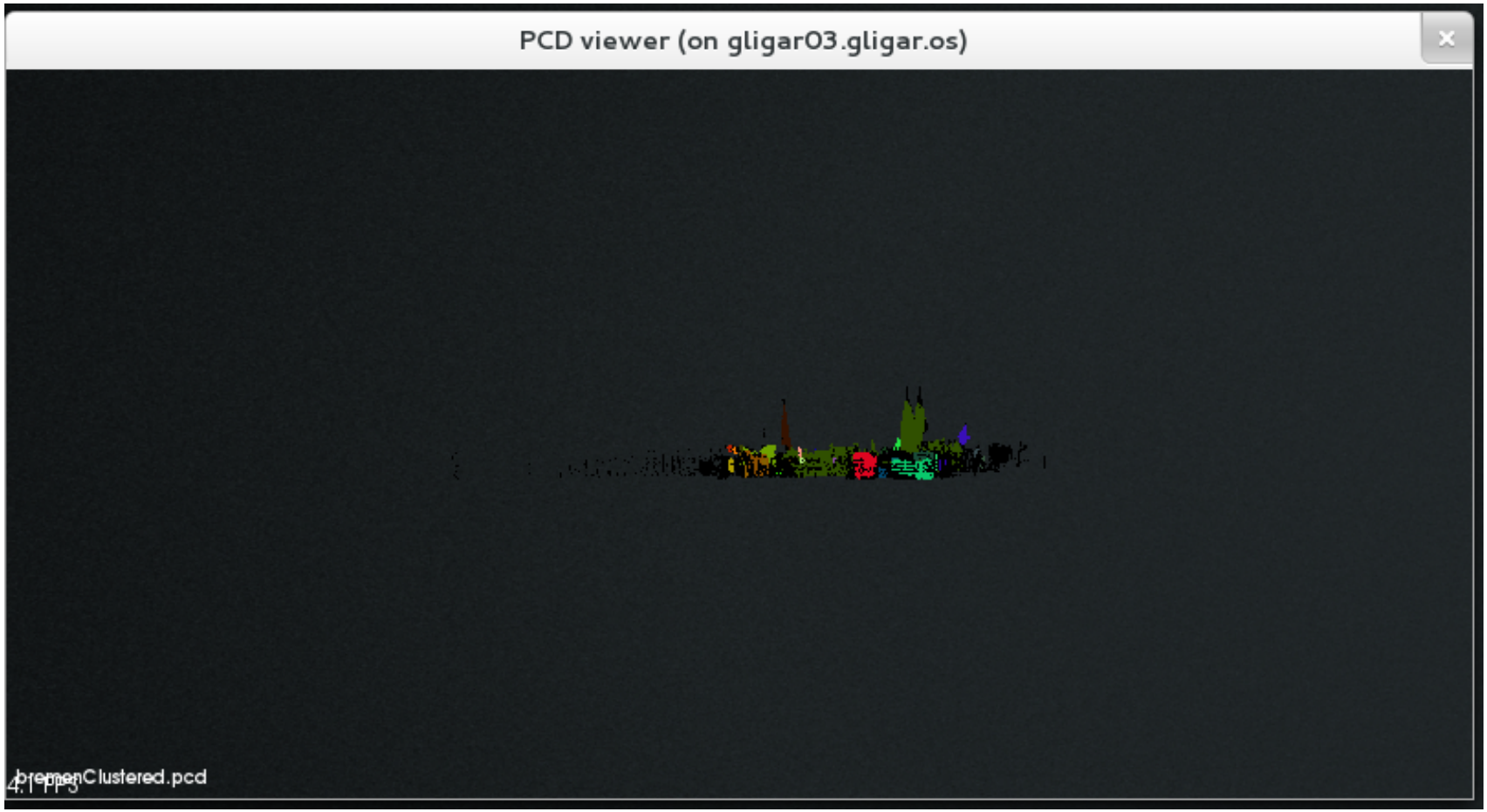
```
[vsc42544@gligar01 ~]$ more HPDBSCAN_example.sh.o1173910
Calculating Cell Space...
        Computing Dimensions... [OK] in 0.017713
        Computing Cells...      [OK] in 0.150861
        Sorting Points...       [OK] in 0.383150
        Distributing Points...  [OK] in 0.605419
DBSCAN...
        Local Scan...           [OK] in 235.598392
        Merging Neighbors...    [OK] in 0.080566
        Adjust Labels ...       [OK] in 0.101197
        Rec. Init. Order ...    [OK] in 1.048572
        Writing File ...        [OK] in 0.017491
Result...
        65      Clusters
        2973821 Cluster Points
        26179   Noise Points
        2953129 Core Points
Took: 238.233050s
Results available in /user/scratch/gent/vsc425/vsc42544/1173910.master19.golett.gent.vsc
[vsc42544@gligar01 ~]$

[vsc42544@gligar01 ~]$ cd /user/scratch/gent/vsc425/vsc42544/1173910.master19.golett.gent.vsc
[vsc42544@gligar01 1173910.master19.golett.gent.vsc]$ ls -al
total 70400
drwxrwxr-x  2 vsc42544 vsc42544     4096 Nov 22 22:33 .
drwx------ 13 vsc42544 vsc42544     4096 Nov 22 22:33
-rw-r--r--  1 vsc42544 vsc42544 72002416 Nov 22 22:37 bremenSmall.h5.h5
```

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

UGent Tier-2 Cluster Golett – Point Cloud Viewer Bremen

```
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
```

UGent Tier-2 Cluster Golett – Point Cloud Viewer Bremen



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

Exercises

Review of Parallel SVM Implementations

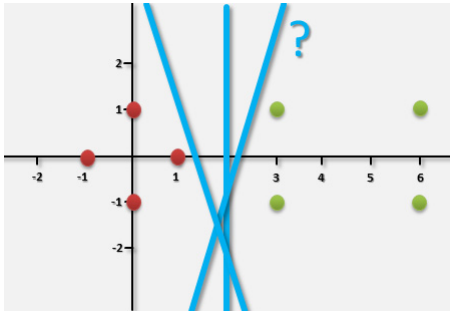
Technology	Platform Approach	Analysis
Apache Mahout	Java; Hadoop	No parallelization strategy for SVMs
Apache Spark/MLlib	Java; Spark	Parallel linear SVMs (no multi-class)
Twister/ParallelSVM	Java; Twister; Hadoop 1.0	Parallel SVMs, open source; developer version 0.9 beta
scikit-learn	Python	No parallelization 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

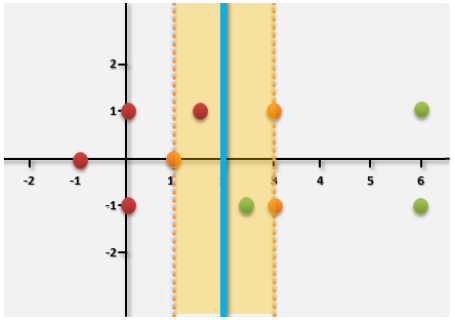
[9] M. Goetz, M. Riedel et al., **'On Parallel and Scalable Classification and Clustering Techniques for Earth Science Datasets',** ***6th Workshop on Data Mining in Earth System Science, International Conference of Computational Science***

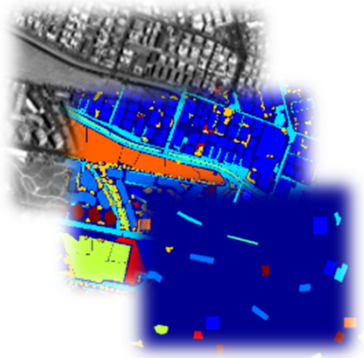
- **Lecture 3-6 will cover Supervised Classification using Support Vector Machines with piSVM**

Parallel and Scalable Machine Learning – piSVM

- ‘Different kind‘ of parallel algorithms
  - Goal is to ‘learn from data‘ instead of modelling/approximate the reality
  - Parallel algorithms often useful to reduce ‘overall time for data analysis‘
- E.g. Parallel Support Vector Machines (SVMs) Technique
  - Data classification algorithm PiSVM using MPI to reduce ‘training time‘
  - Example: classification of land cover masses from satellite image data







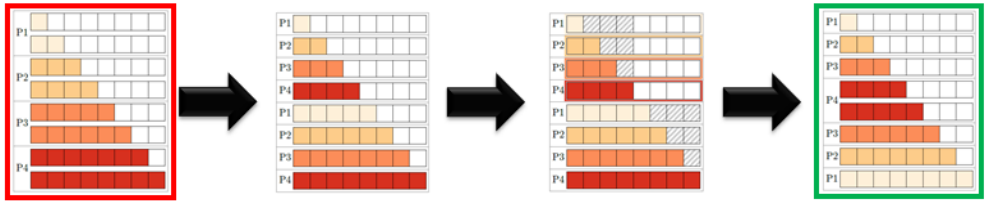
Class	Training	Test
Buildings	18126	163129
Blocks	10982	98834
Roads	16353	147176
Light Train	1606	14454
Vegetation	6962	62655
Trees	9088	81792
Bare Soil	8127	73144
Soil	1506	13551
Tower	4792	43124
Total	77542	697859

[11] G. Cavallaro & M. Riedel et al., ‘On Understanding Big Data Impacts in Remotely Sensed Image Classification Using Support Vector Machine Methods’, Journal of Applied Earth Observations and Remote Sensing

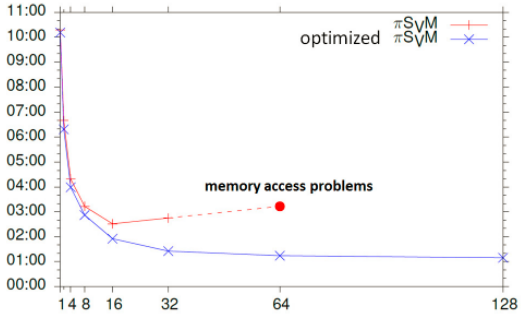
➢ **Lecture 3-6 will cover Supervised Classification using Support Vector Machines with piSVM**

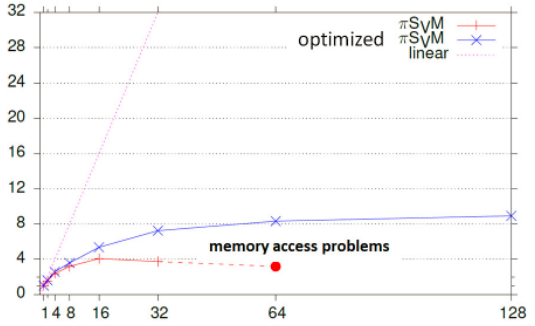
Parallel SVM with MPI Technique – piSVM Implementation

- Original piSVM 1.2 version (2011)
  - Open-source and based on libSVM library, C
  - Message Passing Interface (MPI)
  - New version appeared 2014-10 v. 1.3 (no major improvements)
  - Lack of 'big data' support (e.g. memory, layout)
- 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

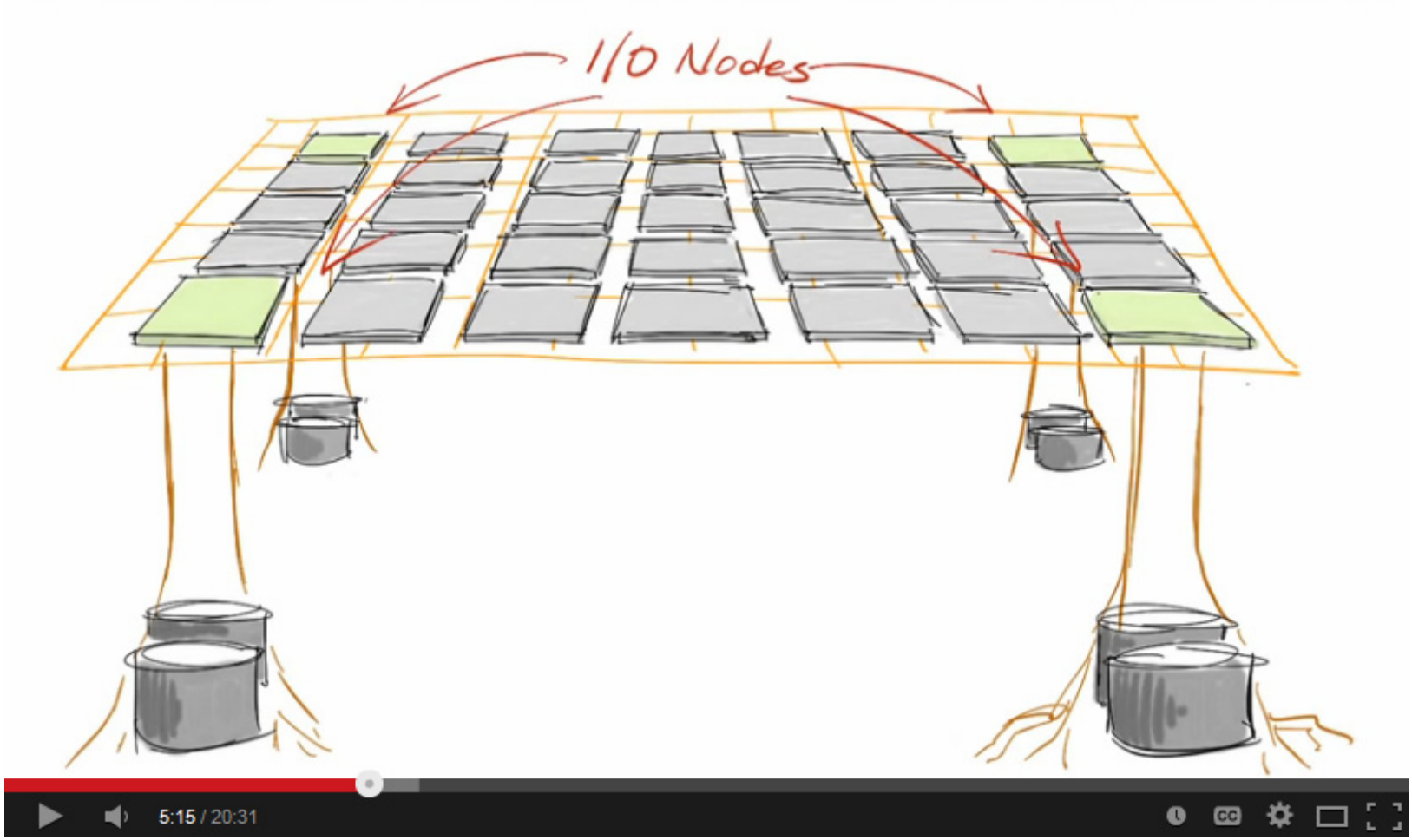


[12] piSVM on SourceForge, 2008



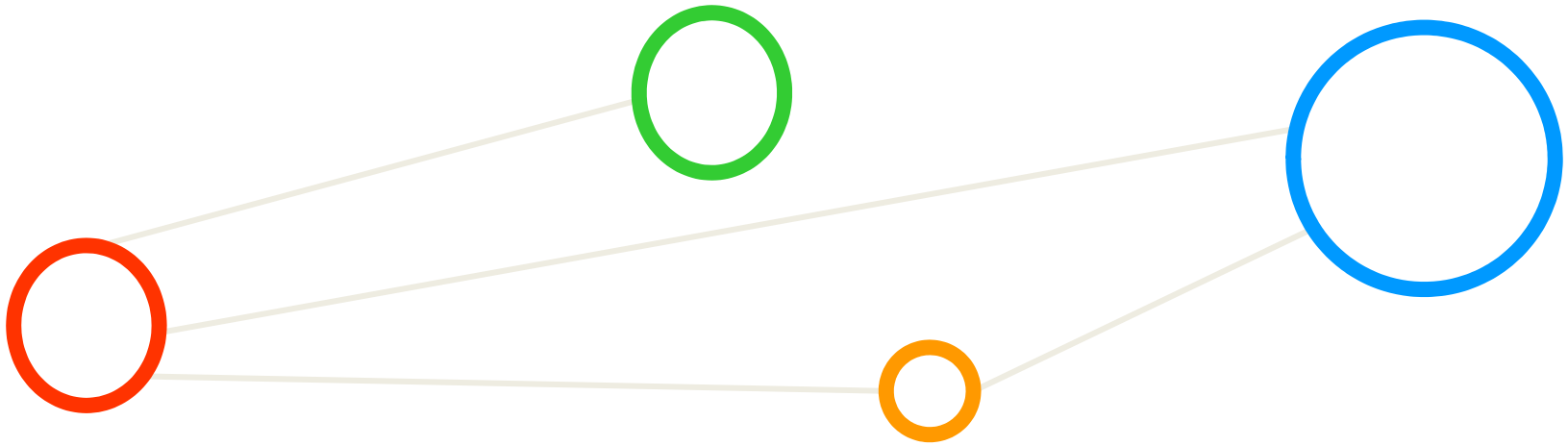


[Video] Parallel I/O with I/O Nodes



[13] Simplifying HPC Architectures, YouTube Video

Lecture Bibliography



Lecture Bibliography (1)

- [1] An Introduction to Statistical Learning with Applications in R, Online: http://www-bcf.usc.edu/~gareth/ISL/index.html
- [2] PANGAEA Data Collection, Data Publisher for Earth & Environmental Science, Online: http://www.pangaea.de/
- [3] Judy Qiu,'Harp: Collective Communication on Hadoop', 2014
- [4] Animation of the k-means algorithm using Matlab 2013, YouTube Video, Online: http://www.youtube.com/watch?v=5FmnJVv73fU
- [5] Ester, Martin, et al. "A density-based algorithm for discoveringclusters in large spatial databases with noise." Kdd. Vol. 96. 1996.
- [6] YouTube Video, 'CSCE 420 Communication Project – DBSCAN', Online: https://www.youtube.com/watch?v=5E097ZLE9Sg
- [7] YouTube Video ,'Point Based Rendering of the Aachen Cathedral' Online: https://www.youtube.com/watch?v=X_wyoroo4co
- [8] YouTube Video, 'Point Based Rendering of the Kaiserpfalz in Kaiserswerth', Online: https://www.youtube.com/watch?v=KvDb58YvlvQ
- [9] 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
- [10] M.Goetz, M. Riedel et al.,'HPDBSCAN – Highly Parallel DBSCAN', accepted for MLHPC Workshop at Supercomputing 2015, Online: http://www.wikicfp.com/cfp/servlet/event.showcfp?eventid=46948

Lecture Bibliography (2)

- [11] G. Cavallaro, M. Riedel, M. Richerzhagen, J.A. Benediktsson, A. Plaza, '*On Understanding Big Data Impacts in Remotely Sensed Image Classification Using Support Vector Machine Methods',* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Issue 99, pp. 1-13, 2015
- [12] Original piSVM tool, Online: http://pisvm.sourceforge.net/
- [13] YouTube Video, 'Big Ideas: Simplifying High Performance Computing Architectures', Online: https://www.youtube.com/watch?v=ISS_OGVamBk
- [14] Ugent Tier-2 Clusters, Online: https://www.vscentrum.be/infrastructure/hardware/hardware-ugent
- [15] B2SHARE, 'HPDBSCAN Benchmark test files', Online: http://hdl.handle.net/11304/6eacaa76-c275-11e4-ac7e-860aa0063d1f
- [16] HDF Group, Online: http://www.hdfgroup.org/
- [17] Parallel NETCDF, Online: http://trac.mcs.anl.gov/projects/parallel-netcdf
- [18] E. Hartnett, 2010-09: NetCDF and HDF5 - HDF5 Workshop 2010
- [19] 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


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