Introduction to Big Data in HPC, Hadoop and HDFS – Part One

Dr. – Ing. Morris Riedel

Adjunct Associated Professor, University of Iceland

Jülich Supercomputing Centre, Germany

Head of Research Group High Productivity Data Processing

Cy-Tera/LinkSCEEM HPC Administrator Workshop, Nicosia, The Cyprus Institute, 19th January – 21th January 2015

HELMHOLTZ | ASSOCIATION

Research Field Key Technologies

Jülich Supercomputing Centre

Supercomputing & Big Data

JÜLICH FORSCHUNGSZENTRUM

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

Research Centre Juelich – Forschungszentrum Juelich

JUELICH in Numbers

Area: 2.2 km^2

Staff: 5236
- Scientists: 1658
- Technical staff: 1662
- Trainees: 303

Budget: 557 Mio. €
- incl. 172 Mio. € third party funding

Located in Germany, Koeln – Aachen Area

Institutes at JUELICH

- Institute of Complex Systems
- Institute for Advanced Simulation
- Juelich Supercomputing Center
- Juelich Center for Neutron Science
- Peter-Grünberg Institute
- Institute for Neuroscience and Medicine
- Institute for Nuclear Physics
- Institute for Bio and Geosciences
- Institute for Energy and Climate Research
- Central Institute for Engineering, Electronics, and Analytics

Research for generic key technologies of the next generation

Scientific & Engineering Application-driven Problem Solving

University of Iceland School of Engineering & Natural Sciences

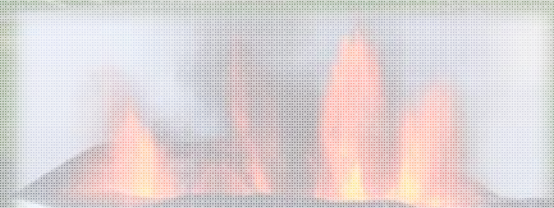
Schools of the University

School of Education
School of Humanities
School of Engineering and Natural Sciences
School of Social Sciences
School of Health Sciences
Interdisciplinary Studies

Full programmes taught in English
Staff: ~ 1259
Students: ~14.000
Located in Reykjavik Capital Center, Iceland

Faculties of the School

Civil and Environmental Engineering
Earth Sciences
Electrical and Computer Engineering
Industrial Engineering
Mechanical Engineering
Computer Science
Life and Environmental Sciences
Physical Sciences



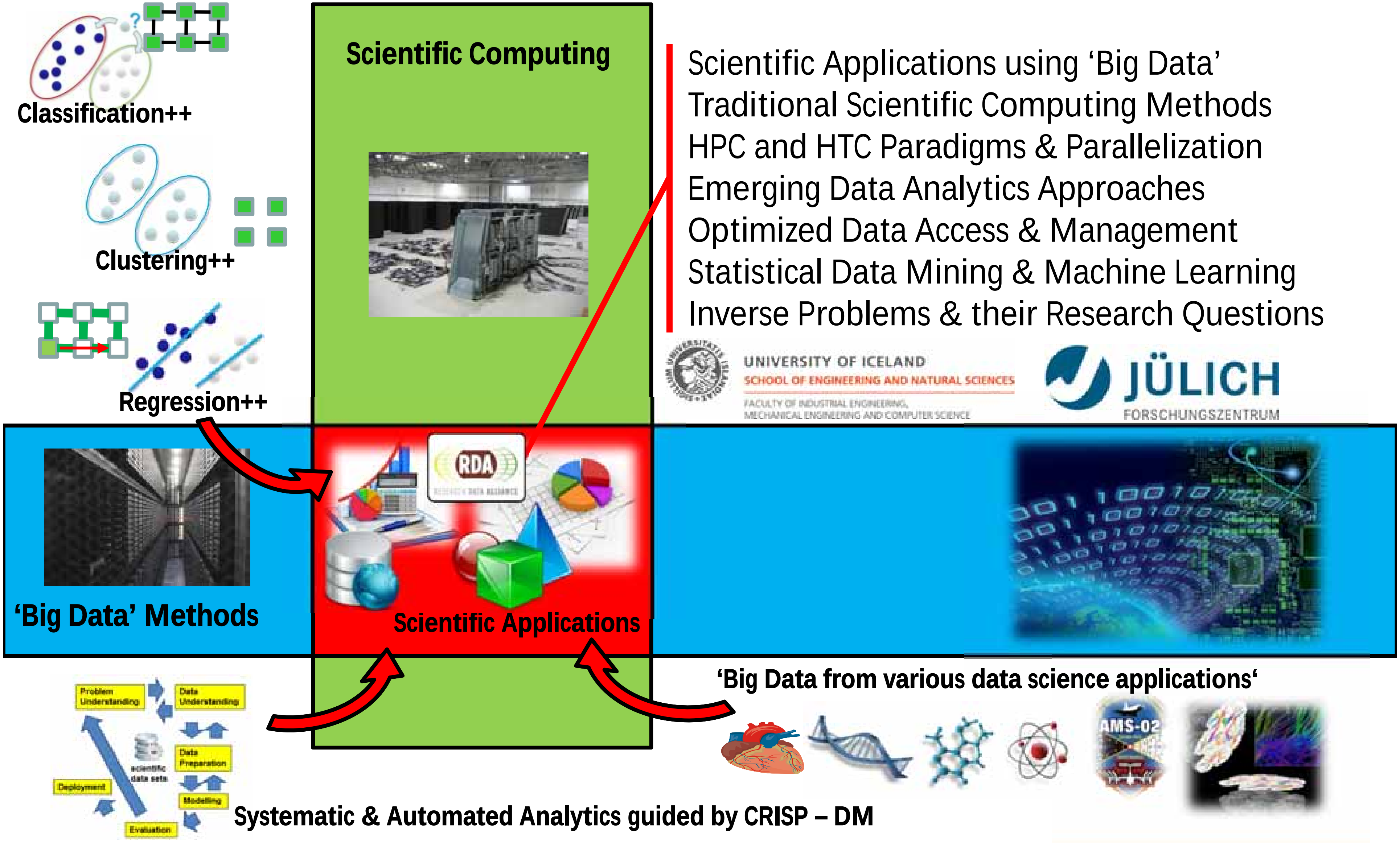
Teaching of key technologies in engineering & sciences

University Courses: Statistical Data Mining & HPC

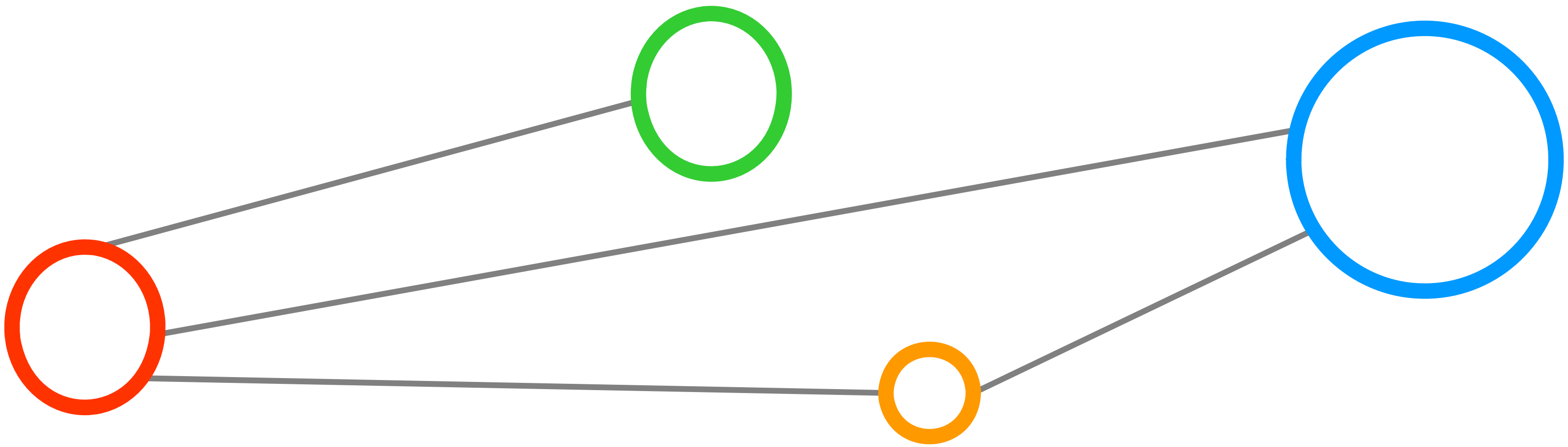
UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

High Productivity Data Processing Research Group – Focus

- Scientific Applications using ‘Big Data’
- Traditional Scientific Computing Methods
- HPC and HTC Paradigms & Parallelization
- Emerging Data Analytics Approaches
- Optimized Data Access & Management
- Statistical Data Mining & Machine Learning
- Inverse Problems & their Research Questions



Overall Course Outline



Overall Course Outline

Part One ‘Big Data‘ Challenges & HPC Tools

- Understanding ‘Big Data‘ in Science & Engineering
- Statistical Data Mining and Learning from ‘Big Data‘
- OpenMP/MPI Tool Example for Clustering ‘Big Data‘
- MPI Tool Example for Classification of ‘Big Data‘

coffee break

Part Two ‘Big Data‘ & Distributed Computing Tools

- Exploring Parallel & Distributed Computing Approaches
- Examples of Map-Reduce & ‘Big Data‘ Processing with Hadoop
- Tools for handling ‘Big Data‘ storage & replication methods
- Technologied for Large-scale distributed ‘Big Data‘ Management

Part One

Part One – Outline

Part One 'Big Data' Challenges & HPC Tools

- Understanding 'Big Data' in Science & Engineering
- Statistical Data Mining and Learning from 'Big Data'
- OpenMP/MPI Tool Example for Clustering 'Big Data'
- MPI Tool Example for Classification of 'Big Data'

coffee break

Part Two 'Big Data' & Distributed Computing Tools

- Exploring Parallel & Distributed Computing Approaches
- Examples of Map-Reduce & 'Big Data' Processing with Hadoop
- Tools for handling 'Big Data' storage & replication methods
- Technologies for Large-scale distributed 'Big Data' Management

Understanding ‘Big Data‘

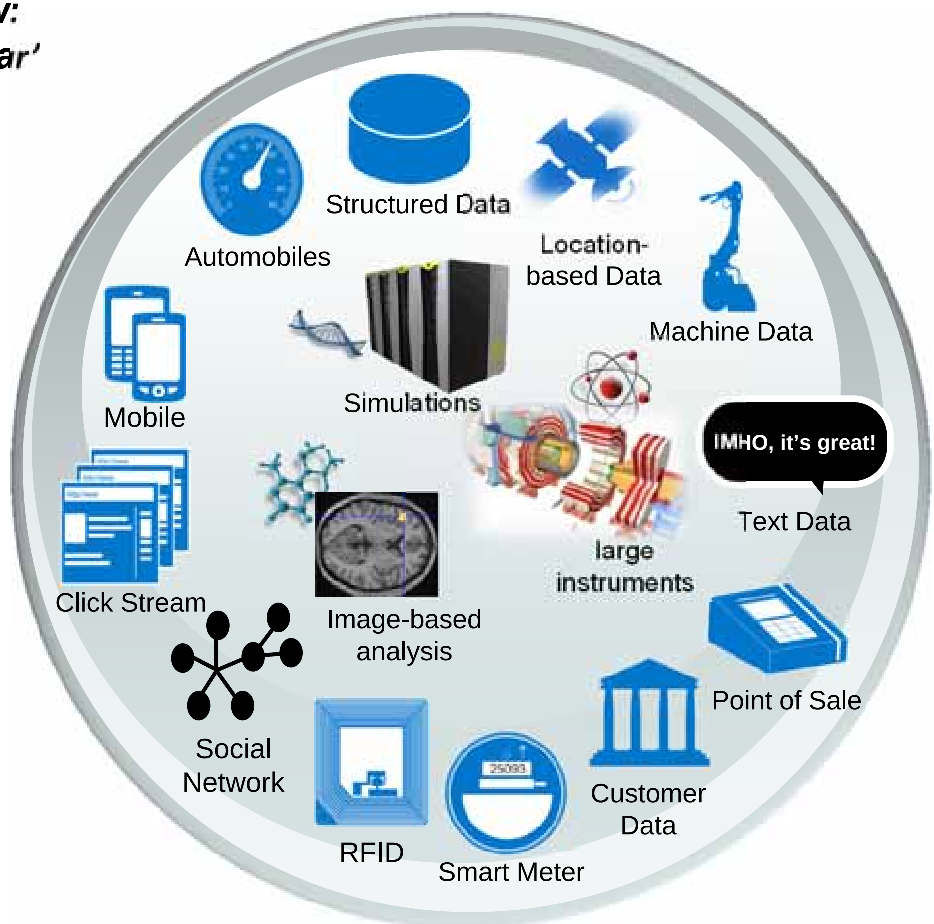
Volume

Variety

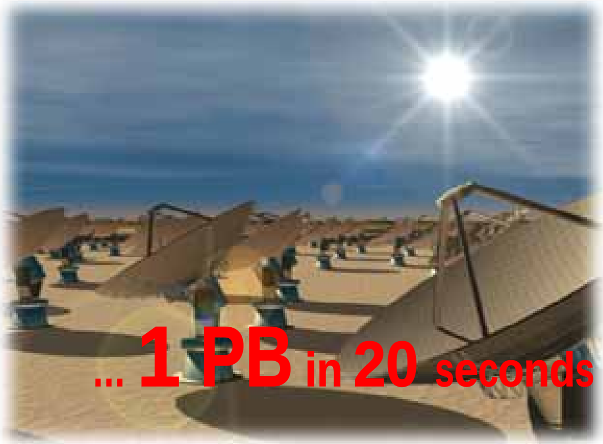
Velocity

- **Big data is data that becomes large enough that it cannot be processed using conventional methods**

[1] O’Reilly Radar Team, ‘Big Data Now: Current Perspectives from O'Reilly Radar’



Context & Multi-Scales (in science & engineering)



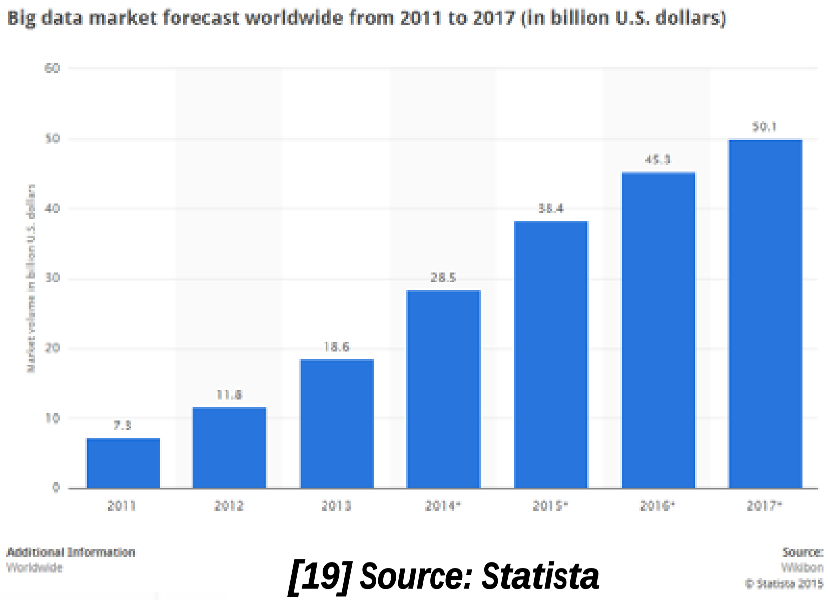
Broad Interest in ‘Big Data‘

Study: Facebook Sharing Comparable To Enjoyment From Sex, Food

June 23, 2012 10:26 AM

Share 8.8k Tweet 398 Share 1.4K View Comments

[17] Facebook Sharing Study



[19] Source: Statista

Stanford Report, January 12, 2015

New Stanford research finds computers are better judges of personality than friends and family

Stanford researchers have found that computers can judge personality traits more accurately than one's friends and colleagues. In fact, artificial intelligence can draw inferences about a person as accurately as a spouse, according to Stanford postdoctoral fellow Michal Kosinski.

BY CLIFTON B. PARKER

[18] Stanford Facebook like study

.... ‘The results showed that a computer could more accurately <u>predict the subject's personality</u> than a work colleague by analyzing just 10 likes; more than a friend or a roommate with 70; a family member with 150; and a spouse with 300 likes.’....

The ‘Big Data Hype‘ in the Media

Nicely summarized in this book

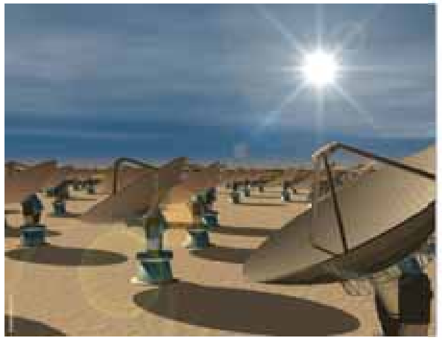
- Includes ‘stories‘ of how ‘big data‘ made/makes a difference in life
- Not a truly scientific source, but line of argumentation is: ‘The data speaks for itself‘

- **One aim of this course is to use concrete tools and analyse data**
- **You will learn to answer: Is bigger data really always better data?**

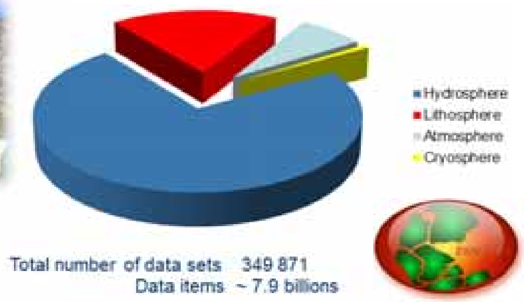
[2] V. Mayer-Schonberger, K. Cukier, ‘Big Data’, 2013

Selected ‘Big Data Types‘ in Science & Engineering

Large Hadron Collider
~12 PB / year

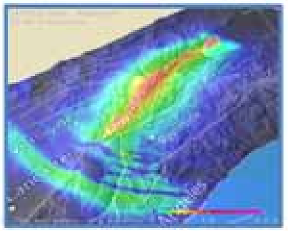


Square Kilometre Array
~1 PB / 20 seconds

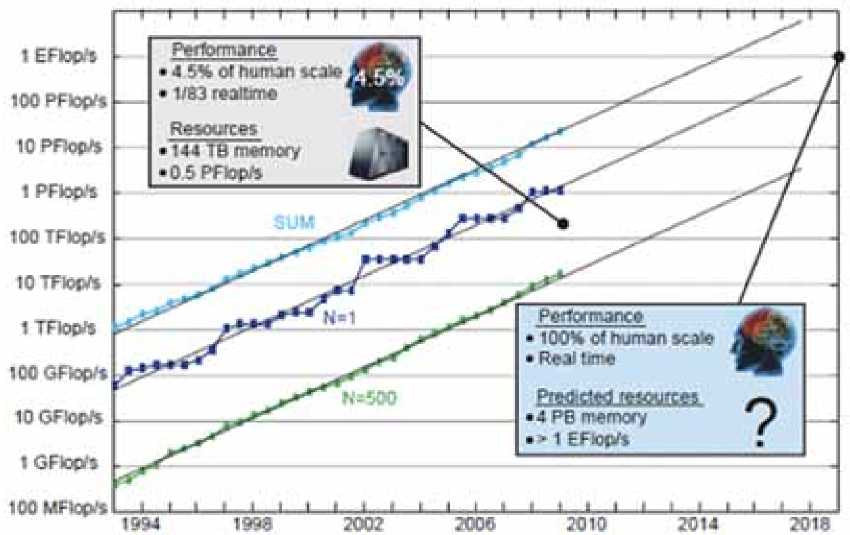


Earth Science Data
Measurement Collections

Estimated figures for simulated 240 second period, 100 hour run-time	TeraShake domain (600x300x80 km^3)	PetaShake domain (800x400x100 km^3)
Fault system interaction	NO	YES
Inner Scale	200m	25m
Resolution of terrain grid	1.8 billion mesh points	2.0 trillion mesh points
Magnitude of Earthquake	7.7	8.1
Time steps	20,000 (.012 sec/step)	160,000 (.0015 sec/step)
Surface data	1.1 TB	1.2 PB
Volume data	43 TB	4.9 PB



[15] F. Berman et al.



[16] Human Brain Project

‘Big Data‘ Key Challenges – Analysis

Scalability

- GBs, TBs, and PBs datasets that fit not into memory
- E.g. algorithms necessary with out-of-core/CPU strategies

High Dimensionality

- Datasets with hundreds or thousand attributes become available
- E.g. bioinformatics with gene expression data with thousand of features

Heterogenous and Complex Data

- More complex data objects emerge and unstructured data sets
- E.g. Earth observation time-series data across the globe (e.g. PANGAEA)

Data Ownership and Distribution

- Distributed datasets are common (e.g. security and transfer challenges)

- **Key challenges faced when performing traditional data analysis and data mining are scalability, high dimensionality of datasets, heterogenous and complex data, data ownership & distribution**

[3] Introduction to Data Mining

‘Big Data‘ Key Challenges for HPC Administrators

Ever increasing volumes, varieties, velocities

- Shift from tape to active disks → active processing
- Data transfer-aware scheduling → transfer takes time
- Different copies of ‘same data‘ → sharing data necessary
- Different copies of ‘same data‘ in different representations → delete some data (e.g. tool-dependent data types, e.g. libsvm format vs. Original image, etc.)

Publication process changes

- Open referencable data is required for journals → data publicly available
- Long-lasting copies years after HPC users finished projects → archiving
- Technology changes, links need to persist in papers → handle systems

New toolsets

- Data replication, in-memory & data sharing tools, different filesystems, etc.
- Statistical data mining codes for classification, clustering, applied statistics, etc.

Part One – Questions

Part One – Outline

Part One 'Big Data' Challenges & HPC Tools

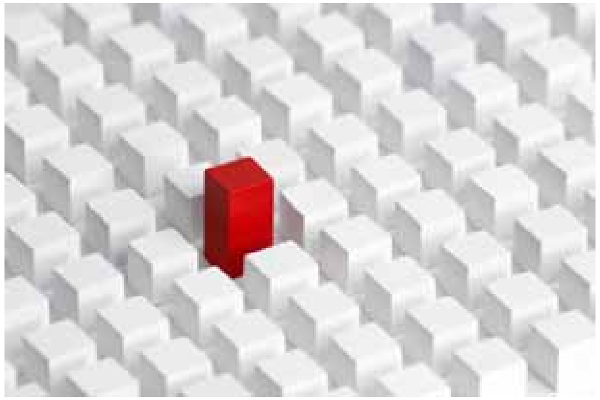
- Understanding 'Big Data' in Science & Engineering
- Statistical Data Mining and Learning from 'Big Data'
- OpenMP/MPI Tool Example for Clustering 'Big Data'
- MPI Tool Example for Classification of 'Big Data'

coffee break

Part Two 'Big Data' & Distributed Computing Tools

- Exploring Parallel & Distributed Computing Approaches
- Examples of Map-Reduce & 'Big Data' Processing with Hadoop
- Tools for handling 'Big Data' storage & replication methods
- Technologied for Large-scale distributed 'Big Data' Management

Get Meaning out of ‘Big Data‘



Rapid advances in data collection and storage technologies in the last decade

- Extracting useful information is a challenge considering ever increasing massive datasets
- Traditional data analysis techniques cannot be used in growing cases (e.g. memory limits in R, Matlab, …)

> - **Data Mining is a technology that blends traditional data analysis methods with sophisticated algorithms for processing large volumes of data**
> - **Data Mining is the process of automatically discovering useful information in large data repositories ideally following a systematic process**

[3] Introduction to Data Mining

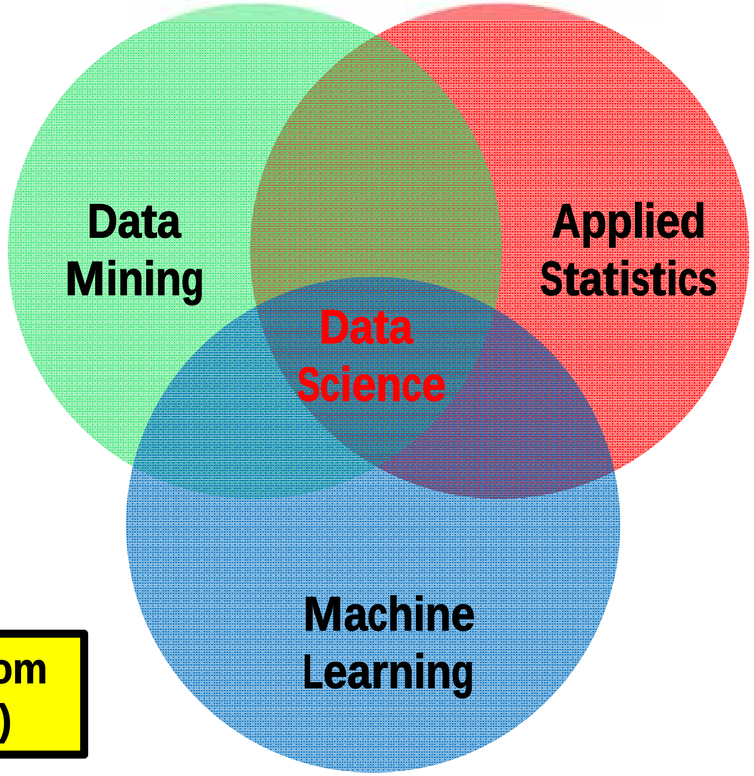
- Statistical Data Mining
  - Traditional statistical approaches are still very useful to consider
  - E.g. in order to reduce large quantities of data to most expressive datasets

Statistical Data Mining

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.

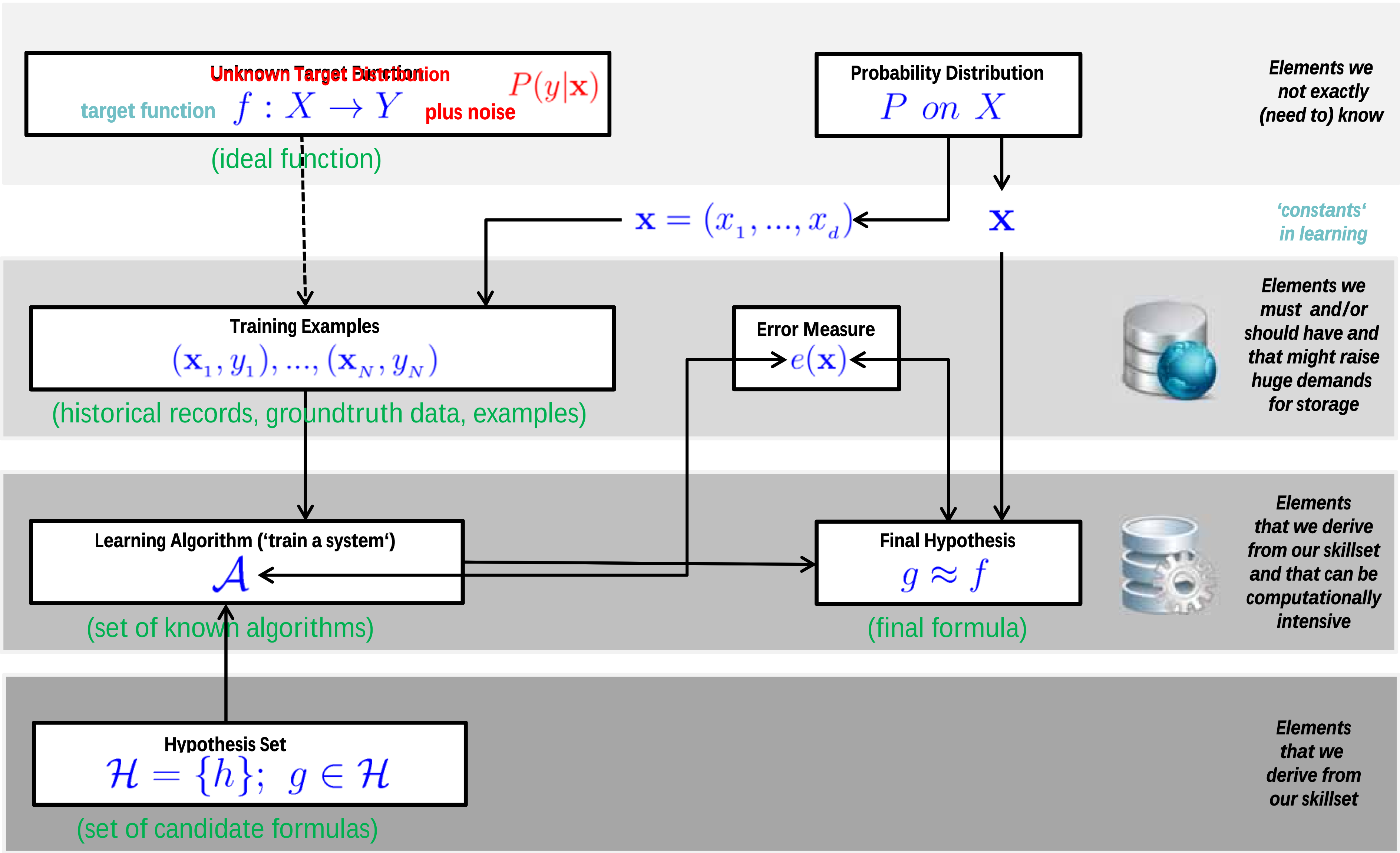


'People with statistical learning skills are in high demand.'

[4] An Introduction to Statistical Learning

- **Statistical Data Mining is a very broad subject and goes from very abstract theory to extreme practice ('rules of thumb')**

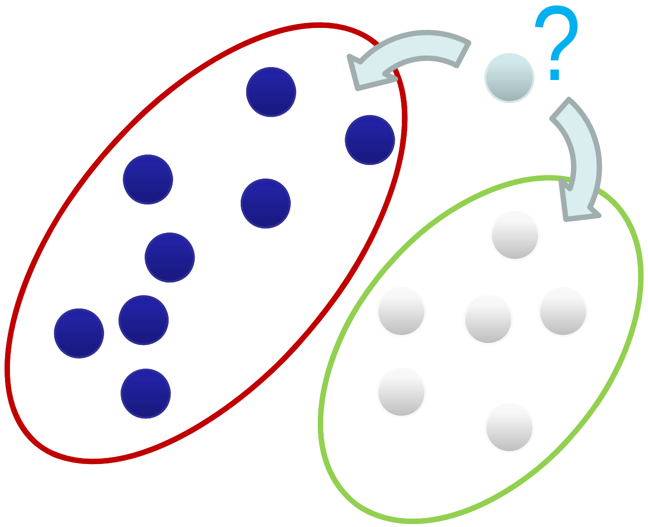
Learning from Data – Computing & Storage Views



Learning from Data – Methods

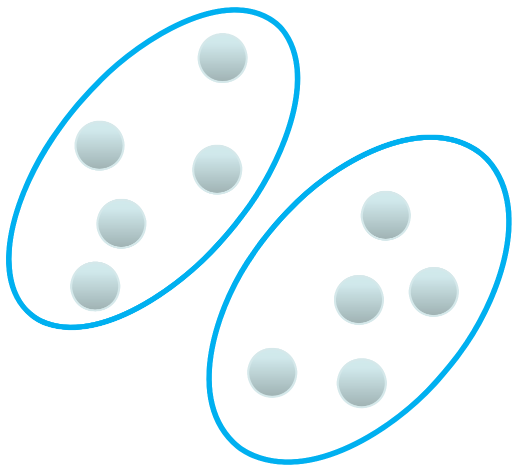
- **Statistical data mining methods can be roughly categorized in classification, clustering, or regression augmented with various techniques for data exploration, selection, or reduction**

Classification



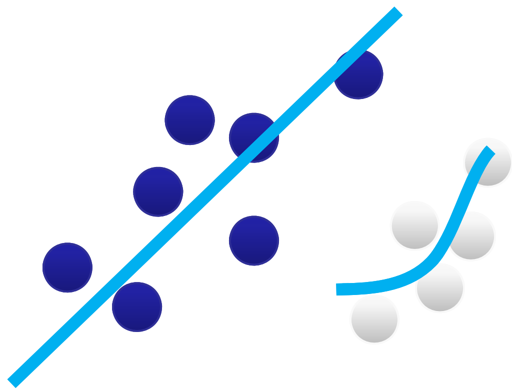
- Groups of data exist
- New data classified to existing groups

Clustering



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

Regression



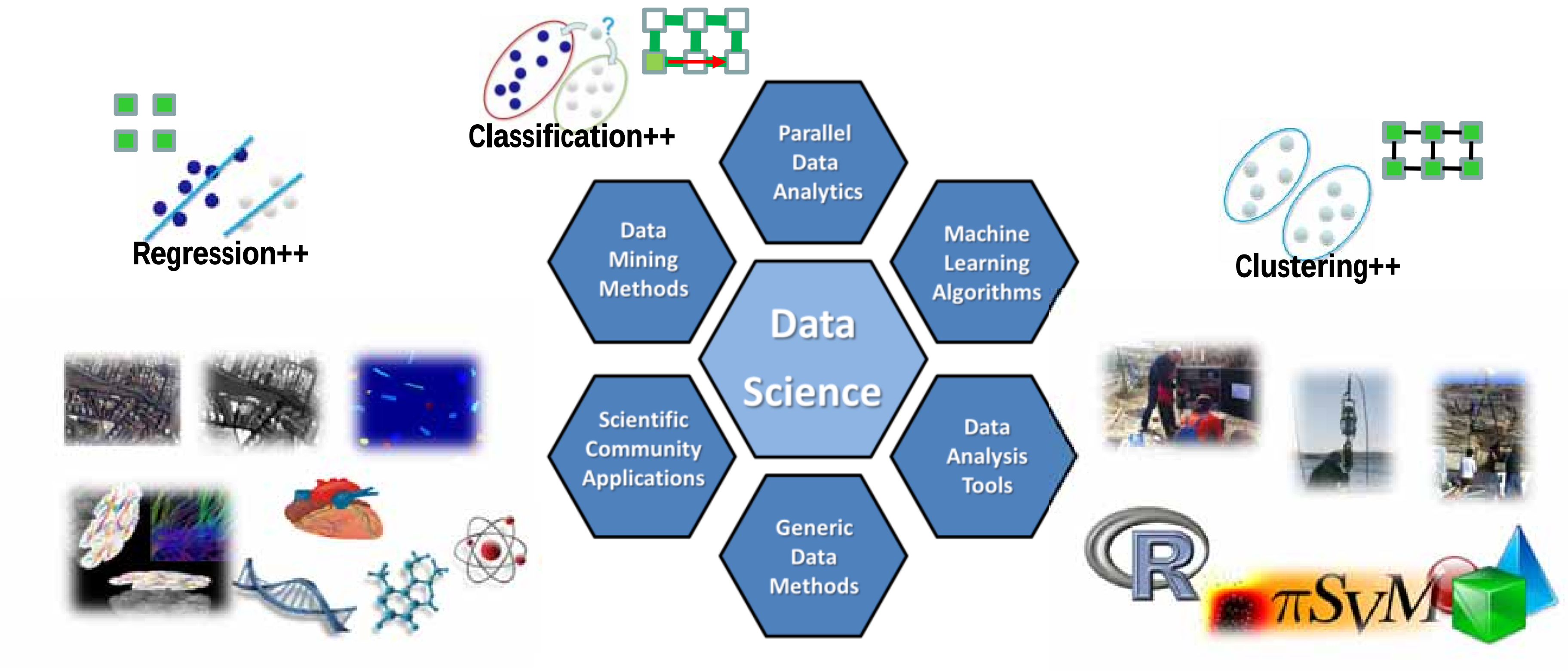
- Identify a line with a certain slope describing the data

- **Learning from Data is much more complicated as shown in this presentation – needs expertise**
- **It requires significant understanding of statistical learning theory, e.g. to prevent overfitting**

Big Data Analytics

'Big Data Analytics' is an 'interesting mix' of different approaches

- E.g. parallel computing techniques with machine learning/data mining methods
- E.g. requires scalable processing methods and underlying infrastructures



Part One – Questions

Part One – Outline

Part One 'Big Data' Challenges & HPC Tools

- Understanding 'Big Data' in Science & Engineering
- Statistical Data Mining and Learning from 'Big Data'
- OpenMP/MPI Tool Example for Clustering 'Big Data'
- MPI Tool Example for Classification of 'Big Data'

coffee break

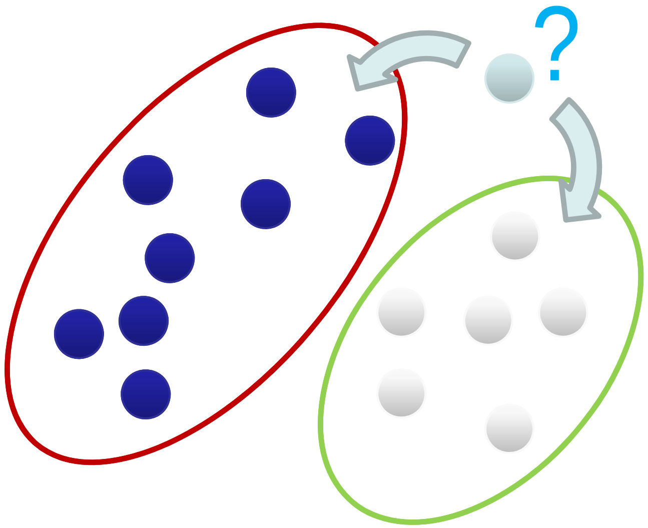
Part Two 'Big Data' & Distributed Computing Tools

- Exploring Parallel & Distributed Computing Approaches
- Examples of Map-Reduce & 'Big Data' Processing with Hadoop
- Tools for handling 'Big Data' storage & replication methods
- Technologied for Large-scale distributed 'Big Data' Management

Learning From Data – Clustering

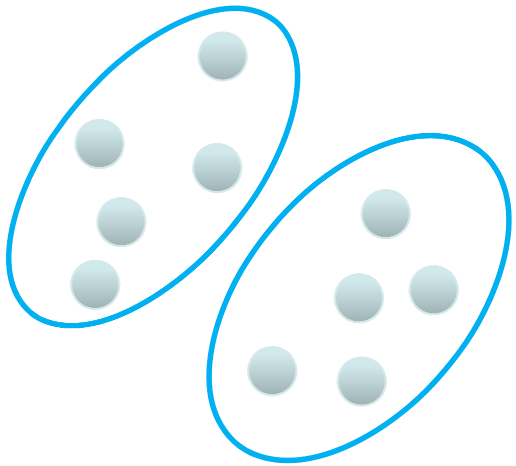
- **Statistical data mining methods can be roughly categorized in classification, clustering, or regression augmented with various techniques for data exploration, selection, or reduction**

Classification



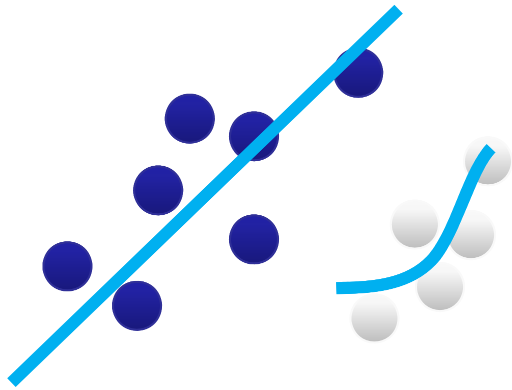
- Groups of data exist
- New data classified to existing groups

Clustering



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

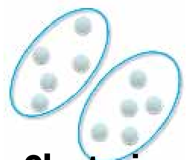
Regression



- Identify a line with a certain slope describing the data

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 (‘supervising‘) 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

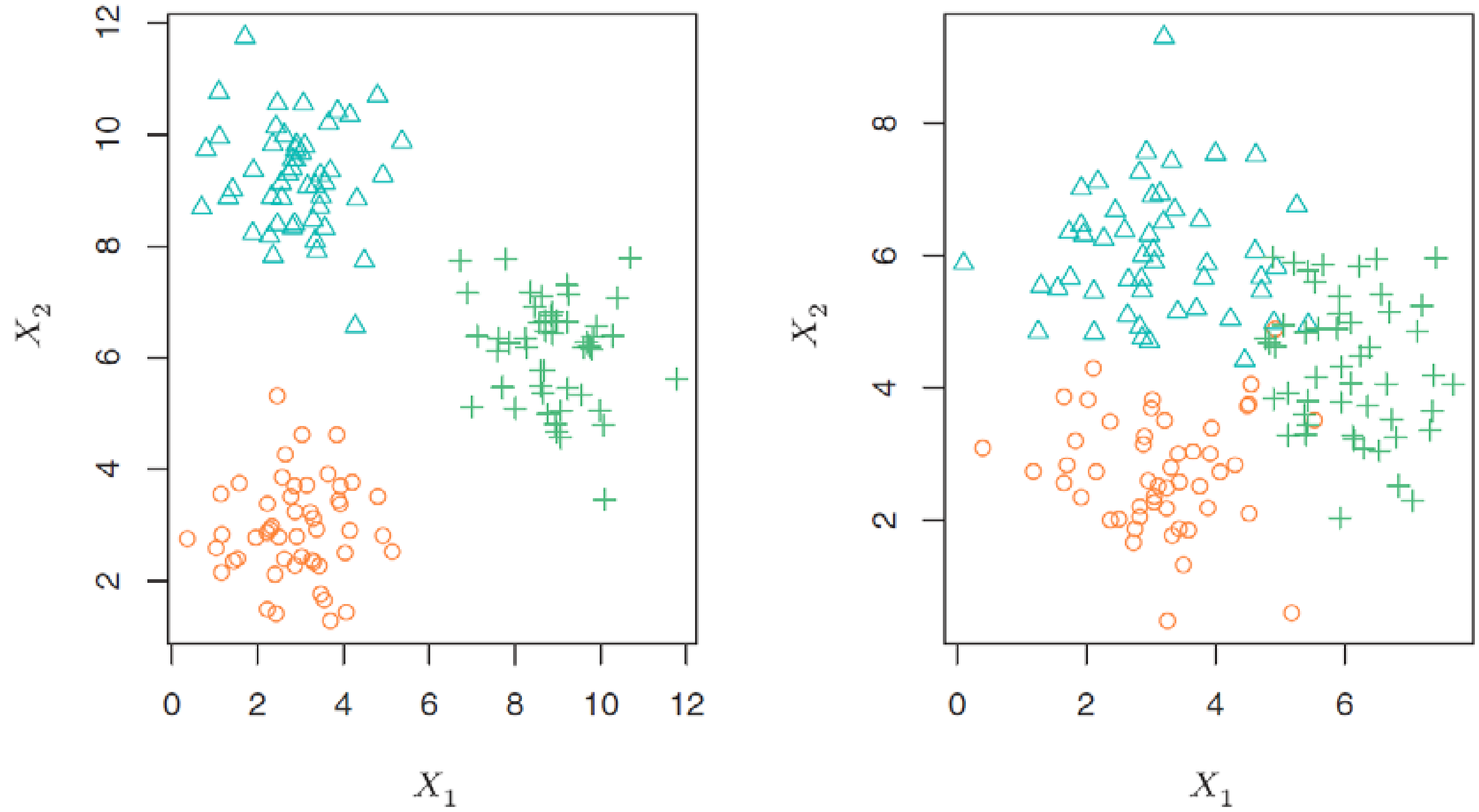
- No response/output that could supervise our data analysis
- Clustering groups that overlap might be hardly recognized as distinct group

- **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, ---]**

[4] An Introduction to Statistical Learning

Learning Approaches – Unsupervised Learning Example

Practice: The number of clusters can be ambiguities



[4] An Introduction to Statistical Learning

Unsupervised Learning – Clustering Methods

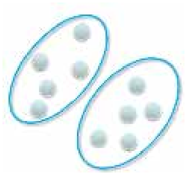
Characterization of clustering tasks

- 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

Parallel & Scalable DBSCAN MPI/OpenMP Tool (1)

DBSCAN Algorithm

- Introduced 1996 by Martin Ester et al. ***[7] Ester et al.***
- Groups number of similar points into clusters of data
- Similarity is defined by a distance measure (e.g. *euclidean distance*)

Unclustered Data

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*

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

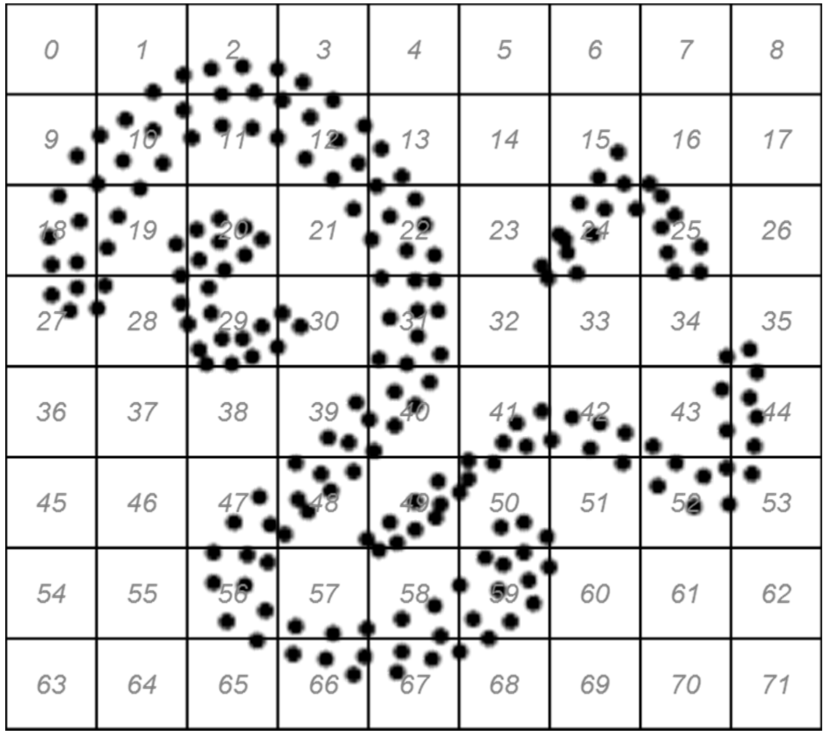
Parallel & Scalable DBSCAN MPI/OpenMP Tool (2)

Parallelization Strategy

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

Preprocessing Step

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



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

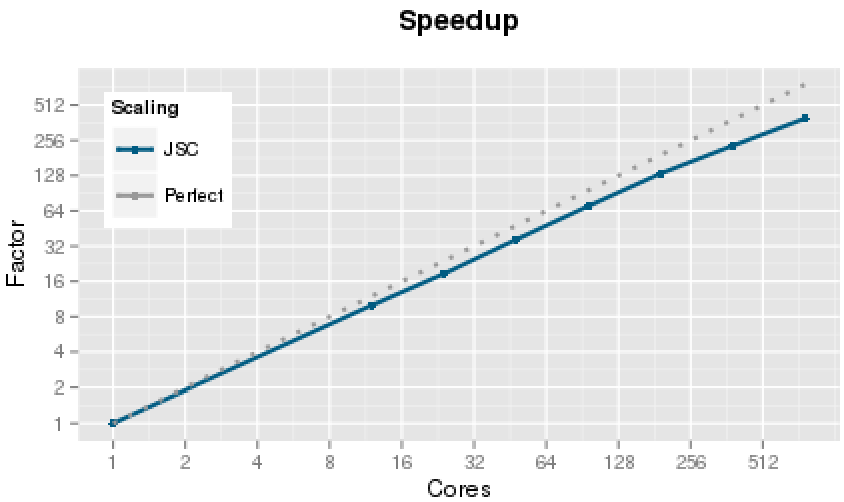
Computational Optimizations

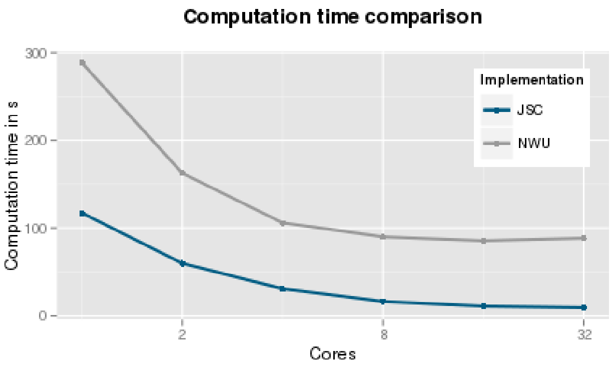
- Caching of point neighborhood searches
- Cluster merging based on comparisons instead of zone reclustering

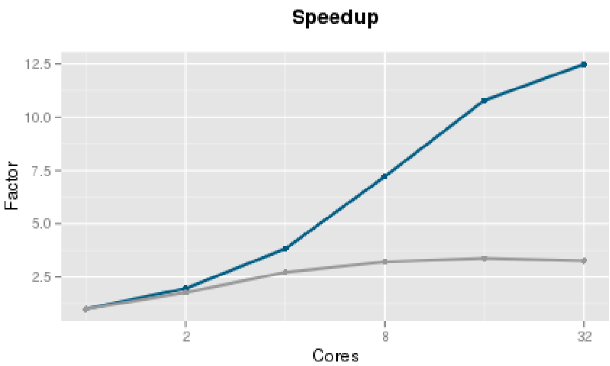
Parallel & Scalable DBSCAN MPI/OpenMP Tool (3)

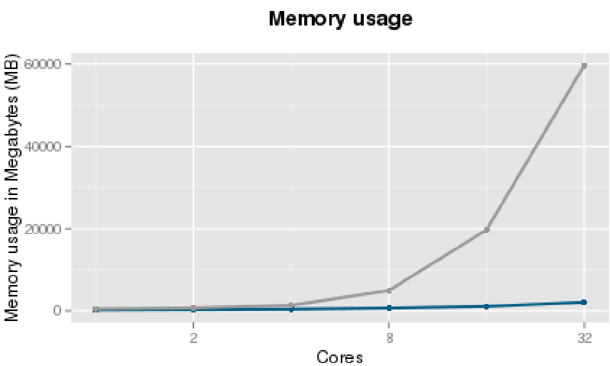
Performance Comparisons

- With another open-source parallel DBSCAN implementation (aka 'NWU') ***[6] Patwary et al.***
- 3.7056.351 data points (2 dimensions)
- Use of Hierarchical Data Format (HDF) v.5 for scalable input/output of 'big data'









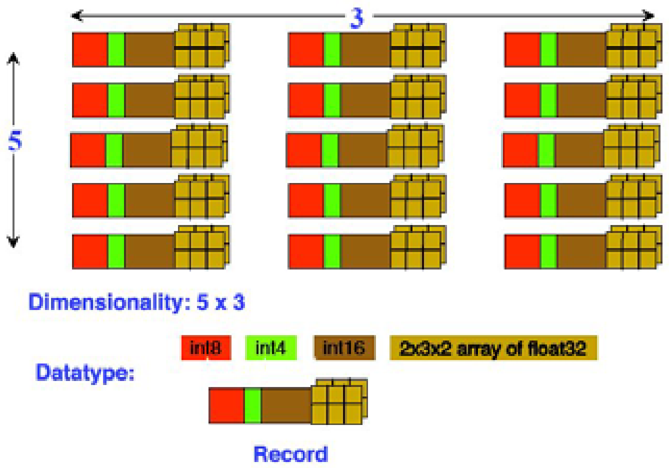
Parallel & Scalable DBSCAN MPI/OpenMP Tool (4)

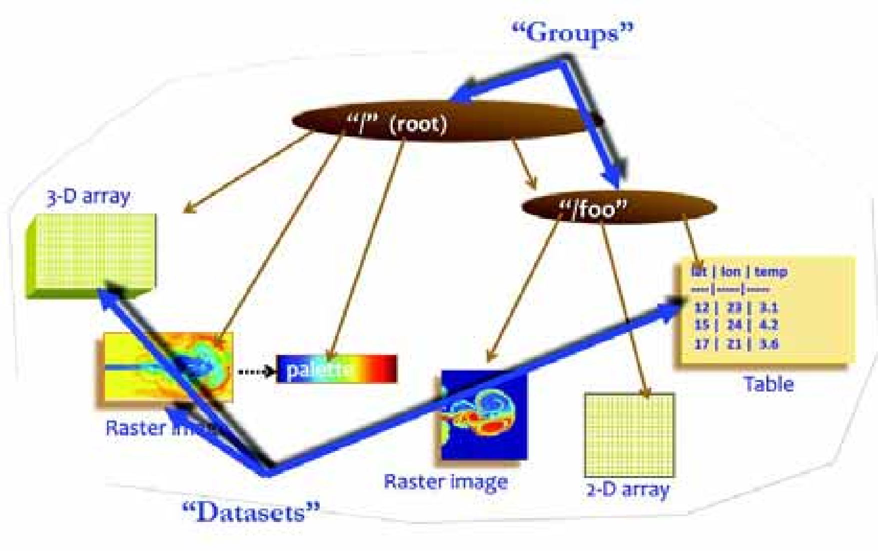
- **Hierarchical Data Format (HDF) is designed to store & organize large amounts of numerical data**
- **HDF is a technology suite that enables the work with extremely large and complex data collections**

E.g. simple HDF compound type data example

[9] M. Stephan et al., HDF@ I/O workshop

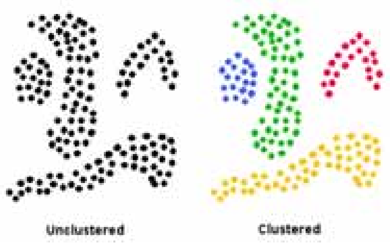
- Array of data records with some descriptive information (5x3 dimension)
- HDF5 data structure type with int(8); int(4); int(16); 2x3x2 array (float32)





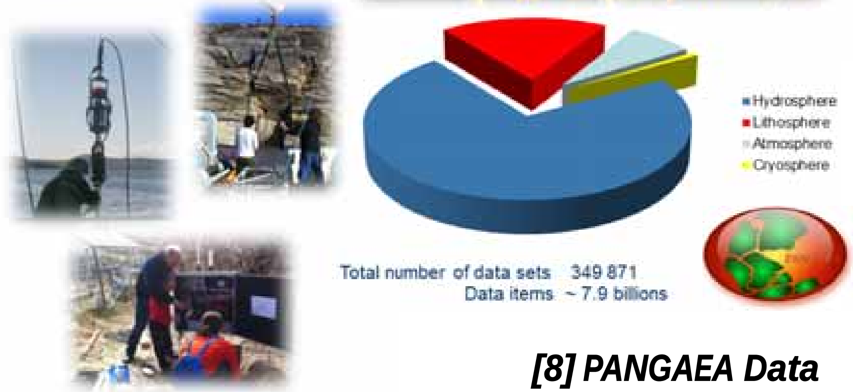
'HDF5 file is a container' to organize data objects

Parallel & Scalable DBSCAN MPI/OpenMP Tool (5)



Selected ‘Big Data‘ Applications

- London twitter data
  (goal: find density centers of tweets)
- Bremen thermo point cloud data
  (goal: noise reduction)
- PANGAEA earth science datasets
  (goal: automated outlier detection)



[8] PANGAEA Data

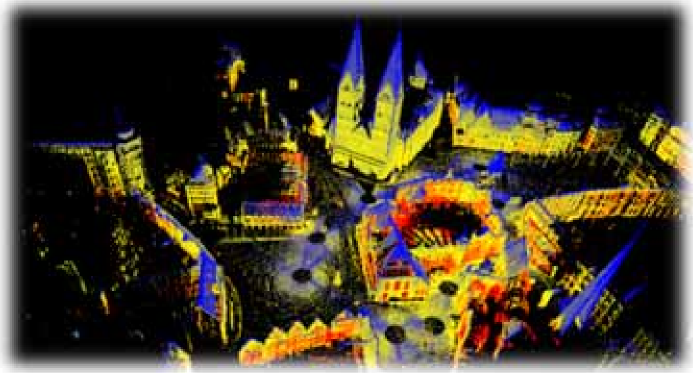
	Cores					
Computation time	1	2	4	8	16	32
JSC-HPDBSCAN	117,18 s	59,64 s	30,68 s	16,25 s	10,86 s	9,39 s
NWU-PDSDBSCAN	288,35 s	162,47 s	105,94 s	89,87 s	85,37 s	88,42 s
Speed-Up						
JSC-HPDBSCAN	1,00 x	1,96 x	3,82 x	7,21 x	10,79 x	12,48 x
NWU-PDSDBSCAN	1,00 x	1,77 x	2,72 x	3,21 x	3,38 x	3,26 x
Memory						
JSC-HPDBSCAN	251,064 MB	345,276 MB	433,340 MB	678,248 MB	1,101 GB	2,111 GB
NWU-PDSDBSCAN	500,512 MB	725,104 MB	1,370 GB	4,954 GB	19,724 GB	59,685 GB

Parallel & Scalable DBSCAN MPI/OpenMP Tool (6)

Tool openly available

- Public bitbucket account – open-source
- https://bitbucket.org/markus.goetz/hpdbscan
- Preparation of tool Website in progress

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



3D Point Cloud of Bremen/Germany

→ Configuration & Demonstration

Usage

- module load hdf5/1.8.13
- mpiexec -np 1 ./dbscan -e 300 -m 100 -t 12 bremenSmall.h5

Parameter *epsilon*

Parameter *minPoints*

Parallel & Scalable DBSCAN MPI/OpenMP Tool (7)

Usage via jobscript

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

JUDGE @ Juelich

```
mriedel@judge:/homeb/zam/analytic/bigdata/hpdbscan/jsc_mpi/mriruns> more datajobscript.sh
#!/bin/bash
#MSUB -N HPDBSCAN_BremenSmall_1_12
#MSUB -l nodes=1:ppn=12:gpus=0:performance
#MSUB -l walltime=00:03:00
#MSUB -M m.riedel@fz-juelich.de
#MSUB -m abe
#MSUB -v tpt=12
#MSUB -l vmem=64gb
#MSUB -q devel

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
```

DBSCAN Parameters

Parallel & Scalable DBSCAN MPI/OpenMP Tool (8)

Output with various information

- Run-times of different stages
- Clustering task information
  (e.g. number of identified clusters)
- Noise identification
- Data volume (small Bremen): ~72 MB
- Data volume (large Bremen): ~1.9 GB

JUDGE @ Juelich

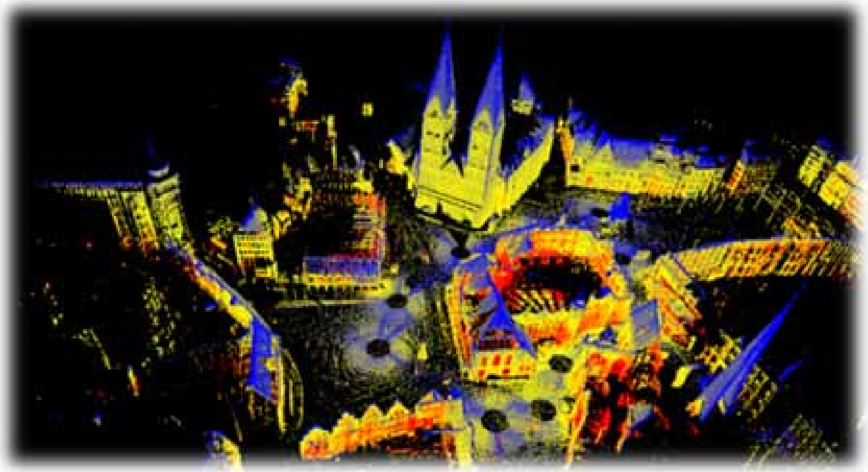
```
mriedel@judge:/homeb/zam/analytic/bigdata/hpdbscan/jsc_mpi/mriruns> more HPDBSCAN_BremenSmall_1_12.o2208066
Calculating Cell Space...
        Computing Dimensions... [OK] in 0.011853
        Computing Cells...      [OK] in 0.073445
        Sorting Points...       [OK] in 0.124476
        Distributing Points...  [OK] in 0.000000
DBSCAN...
        Local Scan... I am ready 0
 in 90.606330           [OK] in 90.606364
        Merging Neighbors...    [OK] in 0.000000
        Adjust Labels ...       [OK] in 0.004972
        Rec. Init. Order ...    [OK] in 1.255420
        Writing File ...        [OK] in 0.019120
Result...
        65      Clusters
        2973821 Cluster Points
        26179   Noise Points
        2953129 Core Points
Took: 92.214843s
```

Output results written in same input data:
cluster number & noise label
(depends on parameters)

Parallel & Scalable DBSCAN MPI/OpenMP Tool (9)

Visualization Example

- Using Point Cloud Library (PDL) toolset
- Transformation of Data to PCD format (python script on the right)



Usage

- python H5toPCD.py bremenSmall.h5
- pcl_viewer bremenClustered.pcd

H5toPCD.py python script

```
import h5py as h5
import numpy as np
import sys

if len(sys.argv) < 2:
    INPUT="bremen.h5"
else:
    INPUT = sys.argv[1]
FILE = "bremenClustered.pcd"

print"loading H5"
bremen = h5.File("bremenSmall.h5")
points = bremen["DBSCAN"]
clusters = bremen["Clusters"]
colors = bremen["COLORS"]

print "Transform to numpy"
points = np.array(points)
clusters = np.array(clusters)
colors = np.array(colors)

#print "Remove Noise"
#points = points[clusters!=0]
#clusters = clusters[clusters!=0]

#data = np.concatenate((points,colors.reshape((-1,1))),axis=1)
data = np.concatenate((points,clusters.reshape((-1,1))),axis=1)

clusters[clusters!=0]=1
data = np.concatenate((data,clusters.reshape((-1,1))),axis=1)

print "Write PCD"
with open(FILE, "w+") as out:
    out.write("""# .PCD v0.7 - Point Cloud Data file format
VERSION 0.7
FIELDS x y z rgb noise
SIZE 4 4 4 4 4
TYPE F F F F F
COUNT 1 1 1 1 1
WIDTH %d
HEIGHT 1
VIEWPOINT 0 -50000 -50000 1 0 0 0
POINTS %d
DATA ascii
""" % (len(data),len(data),))
    np.savetxt(out, data)
```

Take advantage of NumPy library

Part One – Questions

Part One – Outline

Part One 'Big Data' Challenges & HPC Tools

- Understanding 'Big Data' in Science & Engineering
- Statistical Data Mining and Learning from 'Big Data'
- OpenMP/MPI Tool Example for Clustering 'Big Data'
- MPI Tool Example for Classification of 'Big Data'

coffee break

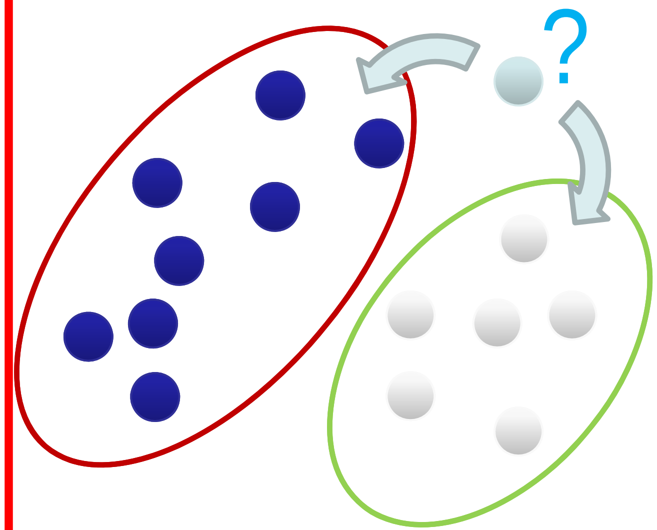
Part Two 'Big Data' & Distributed Computing Tools

- Exploring Parallel & Distributed Computing Approaches
- Examples of Map-Reduce & 'Big Data' Processing with Hadoop
- Tools for handling 'Big Data' storage & replication methods
- Technologied for Large-scale distributed 'Big Data' Management

Learning From Data – Classification

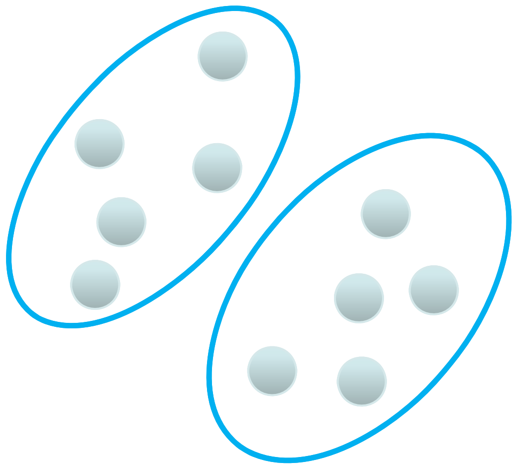
- **Statistical data mining methods can be roughly categorized in classification, clustering, or regression augmented with various techniques for data exploration, selection, or reduction**

Classification



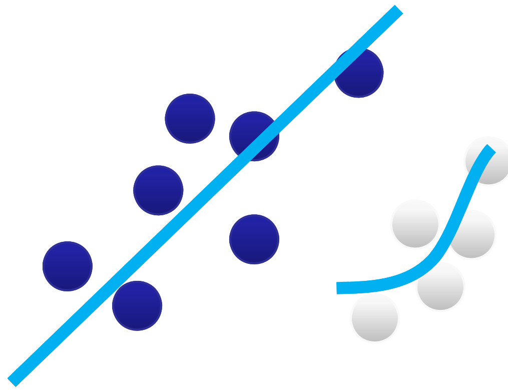
- Groups of data exist
- New data classified to existing groups

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

Learning Approaches – Supervised Learning


Classification

Each observation of the predictor measurement(s) has an associated response measurement:

- Input $\mathbf{x} = x_1, ..., x_d$
- ('Supervising') Output $y_i, i = 1, .., n$
- Data $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$

Goal: Fit a model that relates the response to the predictors

- Prediction: Aims of accurately predicting the response for future observations
- Inference: Aims to better understanding the relationship between the response and the predictors

- **Supervised learning approaches fits a model that related the response to the predictors**
- **Supervised learning approaches are used in linear/logistic regression or support vector machines**
- **Supervised learning works with data = [input, correct output]**

[4] An Introduction to Statistical Learning

Learning Approaches – Supervised Learning Example

- **Build 'reconstructed brain (one 3d volume) that matches with sections & block images (2d images)**
- **Understanding the 'sectioning of the brain' and support automation of reconstruction**

1. Some 'pattern' exists
   - Image content <u>classification</u> : '[1] brain part; [2] not brain part'
2. No exact mathematical formula exists
   - No precise formula for 'contour of the brain'; a non-linear class boundary
3. Towards '~Big Data' (higher resolutions soon)
   - Block face images (of frozen tissue)
   - Every 20 micron (cut size)
   - Resolution: 3272 x 2469
   - ~ 14 MB / RGB image
   - ~ 8 MB / corresponding mask image ('groundtruth')
   - ~700 images

➢ ~ 40 GB dataset – we can apply 'supervised learning' due to labelled data

Supervised Learning – Classification Methods

Characterization of classification tasks

- Using an associated response Y to given inputs X to learn a classifier (system)
- Train and learn via iterations and minimize an error function/measure
- Testing new unseen data with the classifier to find out to which class it belongs
- Risking overfitting of classifier (→ use regularization and validation methods)

Considered often as part of 'data modelling'

- Assessing the results is straightforward, because real validation data exists
- In many cases 'dimensionality reduction' techniques can reduce necessary data (e.g. apply principle component analysis on training data, cut dimensions off)

- **Classification is a broad class of methods that train a classifier system (learning run-time matters)**
- **Classifiers can perform with low/high accuracy when unseen data is tested for class membership**

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‘)

Decision Trees & Ensemble Methods – tree-based classification

- Grows trees for class decisions, ensemble methods average a number of 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

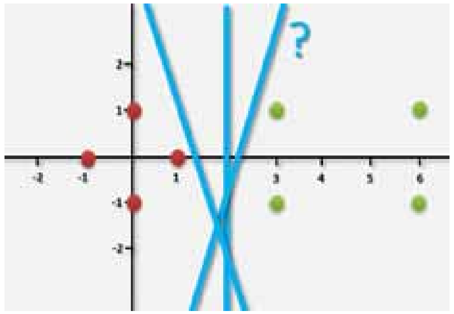
- Use of the Bayes theorem with strong/naive independence between features

Parallel & Scalable SVM MPI Tool (1)

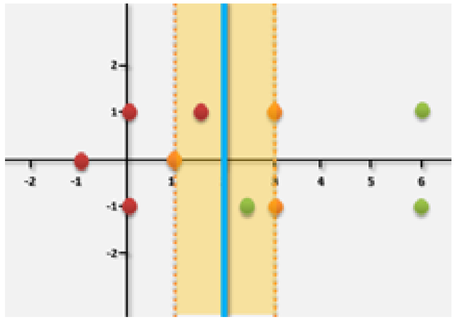
SVM Algorithm Approach

[10] 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
- Uses quadratic programming & Lagrangian method with N x N



(linear example)



(‘maximal margin clasifier‘ example)

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

(maximizing hyperplane turned into optimization problem, minimization, dual problem)

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

(max. hyperplane → dual problem, using quadratic programming method)

$$\begin{bmatrix} y_1 y_1 x_1^T x_1 & y_1 y_2 x_1^T x_2 & \dots y_1 y_N x_1^T x_N \\ \dots & \dots & \ddots \\ y_N y_1 x_N^T x_1 & y_N y_2 x_N^T x_2 & \dots y_N y_N x_N^T x_N \end{bmatrix}$$

(quadratic coefficients)

- **Intuition tells us just ‘furthest away’ from the closest points is a good position for the line**

Parallel & Scalable SVM MPI Tool (2)

- **True Support Vector Machines are Support Vector Classifiers combined with a non-linear kernel**
- **Non-linear kernels exist - mostly known are polynomial & Radial Basis Function (RBF) kernels**

[4] An Introduction to Statistical Learning

Understanding the MPI tool parameters

- Selecting non-linear kernel function K type as RBF → parameter –t 2
- Setting RBF Kernel configuration parameter γ → e.g. parameter –g 16
- Setting SVM allowed errors parameter → e.g. parameter –c 10000

Major benefit of Kernels: Computing done in original space

- Linear Kernel $$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$$ (linear in features)
- Polynomial Kernel $$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$$ (polynomial of degree d)
- RBF Kernel $$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$ (large distance, small impact)

Parallel & Scalable SVM MPI Tool (3)

Original parallel piSVM tool 1.2

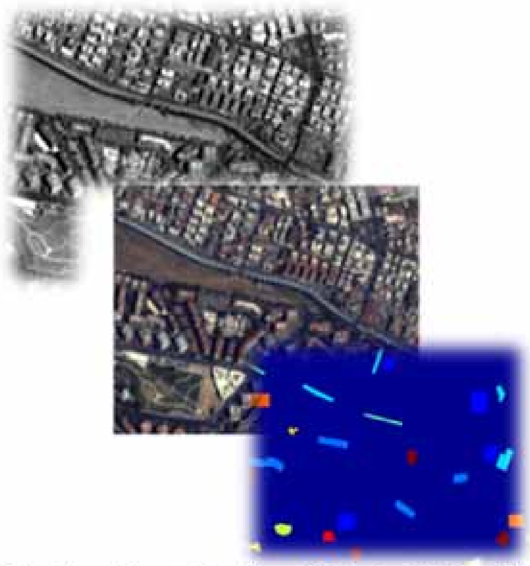
- Open-source and based on libSVM library
- Old C code (was not maintained, 2011)
- Message Passing Interface (MPI)
- Lack of 'big data' support (memory, layout, etc.)
- New version appeared 2014-10 1.3 (currently in investigation)

[11] piSVM Website, 2011 code

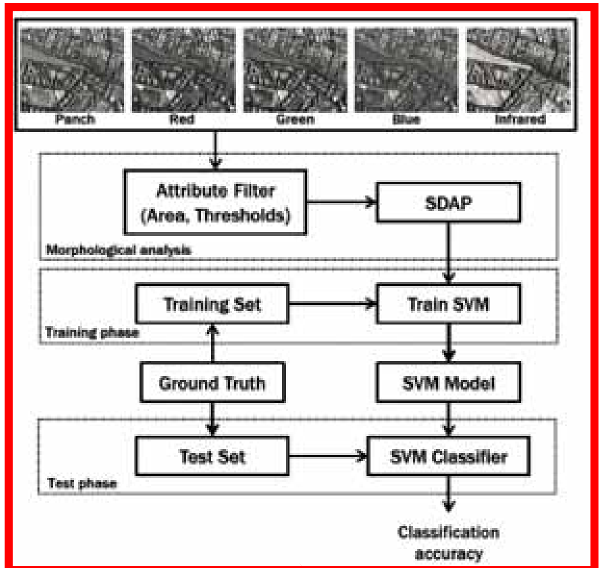
Tuned scalable parallel piSVM tool 1.2.1

- Open-source (repository to be created)
- Based on piSVM tool 1.2
- Optimization w.r.t. load imbalance with increasing cores
- Tunings w.r.t. collective MPI operations
- Cooperation with piSVM developer is planned
- Contact: m.richerzhagen@fz-juelich.de

Parallel & Scalable SVM MPI Tool (3)



Sattelite Data (Quickbird)



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

Parallel Support Vector Machines (SVM)

HPC / MPI

Classification Study of Land Cover Types

„Reference Data Analytics“ for reusability & learning

CRISP-DM Report	Openly Shared Datasets	Running Analytics Code

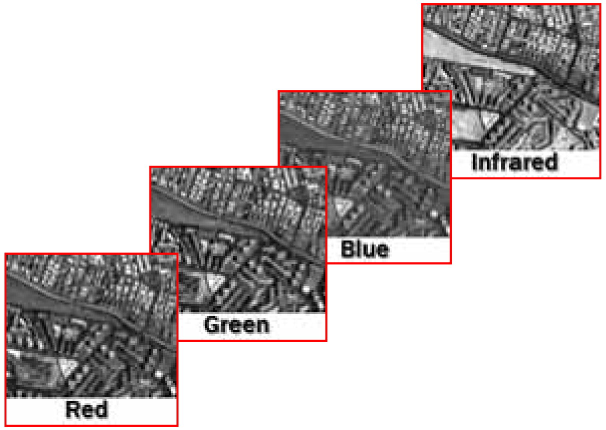
Parallel & Scalable SVM MPI Tool (4)

Example dataset: Geographical location: Image of Rome, Italy

- Remote sensor data obtained by Quickbird satellite

High-resolution (0.6m) panchromatic image

Pansharpened (UDWT) low-resolution (2.4m) multispectral images



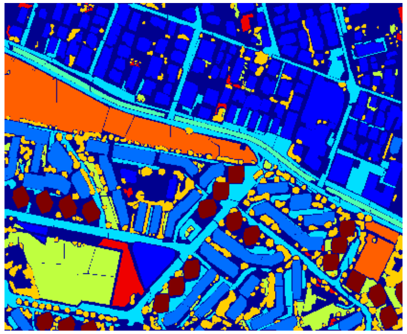
[12] Rome Image dataset

B2SHARE
Store and Share Research Data

Parallel & Scalable SVM MPI Tool (5)

Labelled data available for train/test data

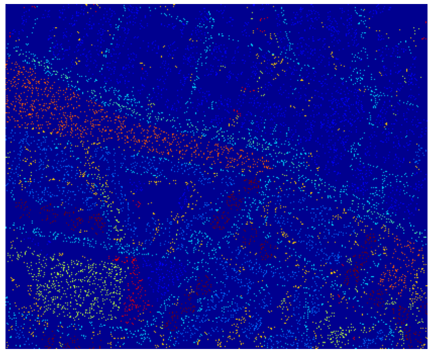
- Groundtruth data of 9 different land-cover classes available



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

Data preparation

- We generated a set of training samples by randomly selecting 10% of the reference samples (with labelled data)
- Generated set of test samples from the remaining labels (labelled data, 90% of reference samples)



Training Image
(10% pixels/class)

[12] Rome Image dataset

B2SHARE Store and Share Research Data

Parallel & Scalable SVM MPI Tool (6)

Based on 'LibSVM data format'

- Add 'Self-Dual Attribute Profile (SDAP) on Area' on all images training file

Area

Std Dev

Moment of Inertia

[14] G. Cavallaro, M. Mura, J.A. Benediktsson, L. Bruzzone et al.

Class — **Number Feature** — **Gray Level**

Each line is a training vector with gray levels

each line is a pixel

```
3 1:0.105882 2:0.109804 3:0.101961  ........  54:0.121569 55:0.130952
2 1:0.364706 2:0.360784 3:0.356863  ........  54:0.356863 55:0.349206
6 1:0.152941 2:0.34902  3:0.454902  ........  54:0.466667 55:0.460317
........
........
........
.......
9 1:0.247059 2:0.247059 3:0.227451  ........  54:0.227451 55:0.218254
7 1:0.411765 2:0.411765 3:0.415686  ........  54:0.415686  55:0.40873
```

Buildings
Blocks
Roads
Light Train
Vegetation
Trees
Bare Soil
Soil
Tower

#77542 samples

55 features

[12] Rome Image dataset

B2SHARE Store and Share Research Data

Parallel & Scalable SVM MPI Tool (6)

Usage via jobscript

- Using MOAB job scheduler
- np = number of processors; o/q partitioning

JUDGE @ Juelich

```
#!/bin/bash
#MSUB -N Train-tune-rec86-4-16-32
#MSUB -l nodes=4:ppn=16:performance
#MSUB -l walltime=03:00:00
#MSUB -M m.riedel@fz-juelich.de
#MSUB -m abe
#MSUB -W x=naccesspolicy:singlejob
#MSUB -v tpt=2
#MSUB -q devel


### jobscript

cd $PBS_O_WORKDIR
echo "workdir: $PBS_O_WORKDIR"

NSLOTS=32

echo "running on $NSLOTS cpus..."

### 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
```

→ Configuration & Demonstration

SVM Parameters

[13] Rome Analytics Results

B2SHARE Store and Share Research Data

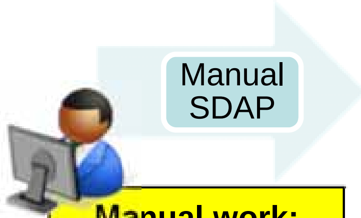
Parallel & Scalable SVM MPI Tool (7)

Training speed-up is possible when number of features is ‘high‘

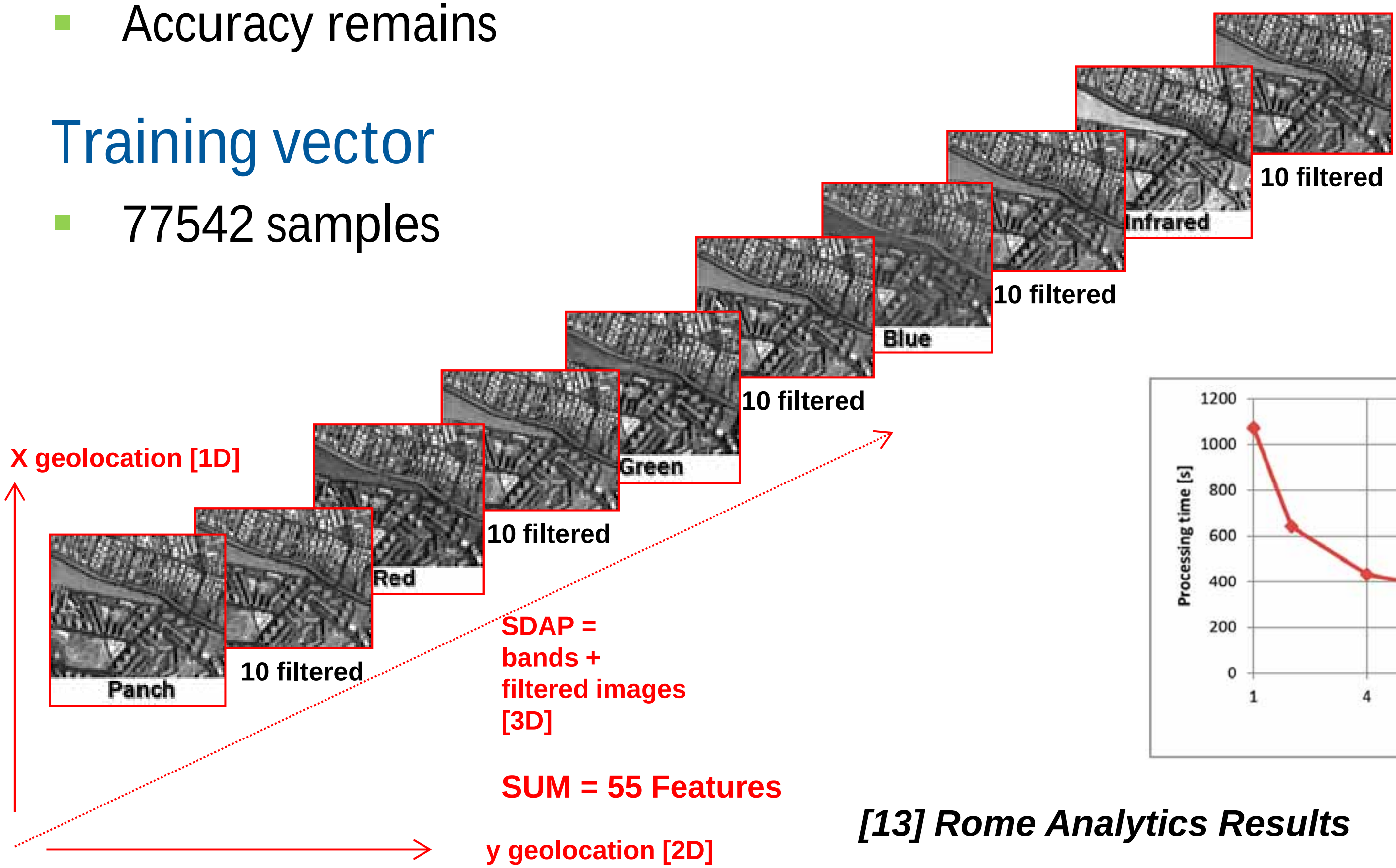
- Serial Matlab: ~1277 sec (~21 minutes)
- Parallel (16) Analytics: 220 sec (3:40 minutes)
- Accuracy remains

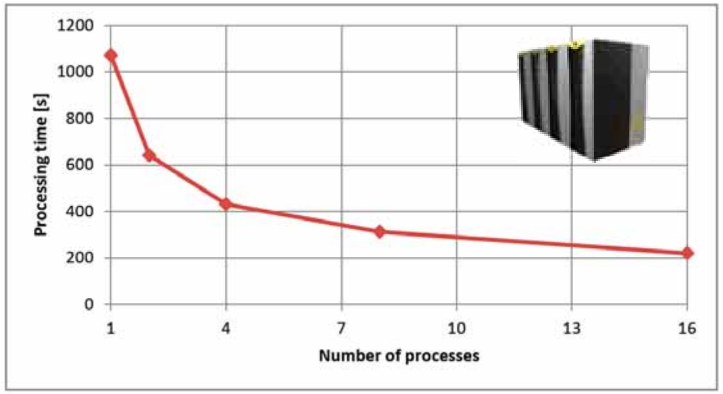
Training vector

- 77542 samples



Manual work: Obtain the SDAP for all image bands using attribute ‘area‘ (10 thresholds)



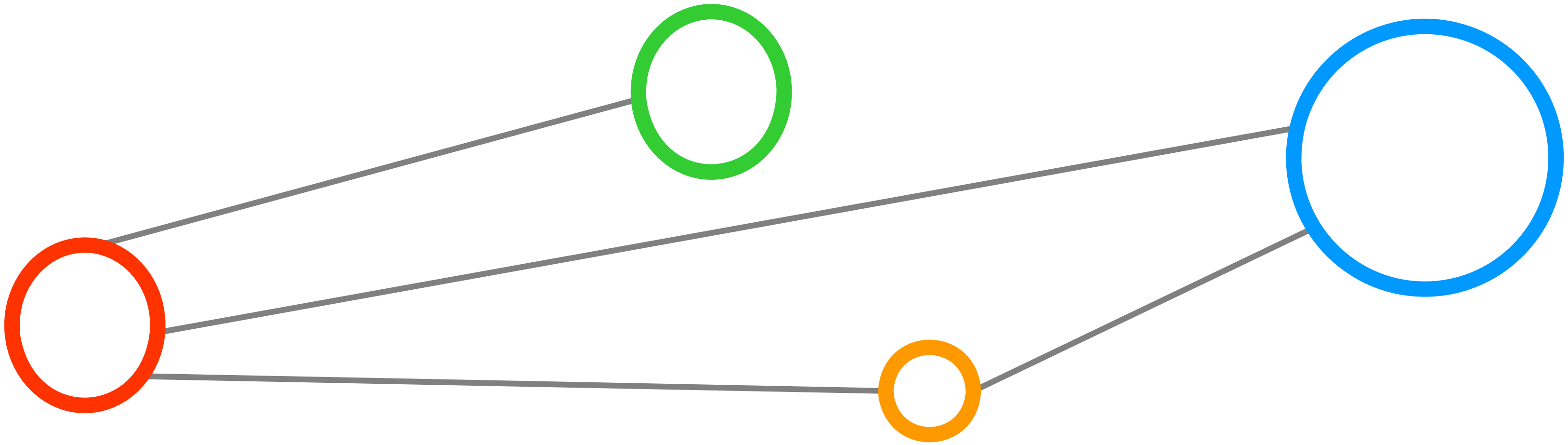


[13] Rome Analytics Results

B2SHARE Store and Share Research Data

Part One – Questions

References



References (1)

[1] O’Reilly Radar Team, ‘Big Data Now: Current Perspectives from O'Reilly Radar’

[2] Viktor Mayer-Schonberger, Kenneth Cukier, ‘Big Data: A Revolution That Will Transform How We Live, Work, and Think’, ISBN 0544002695, 2013

[3] Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Addison Wesley, ISBN 0321321367, English, ~769 pages, 2005

[4] An Introduction to Statistical Learning with Applications in R, Online: http://www-bcf.usc.edu/~gareth/ISL/index.html

[5] 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

[6] Patwary, Md Mostofa Ali, et al. "A new scalable parallel dbscanalgorithm using the disjoint-set data structure." High PerformanceComputing, Networking, Storage and Analysis (SC), 2012 InternationalConference for. IEEE, 2012

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

[8] PANGAEA Earth Science Data Collection, Online: http://www.pangaea.de/

[9] 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

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

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

[12] B2SHARE data collection Rome Dataset, online: http://hdl.handle.net/11304/4615928c-e1a5-11e3-8cd7-14feb57d12b9

[13] B2SHARE data collection piSVM1.2 Analytics JUDGE Cluster Rome Images 55 Features, online: http://hdl.handle.net/11304/6880662c-1edf-11e4-81ac-dcbd1b51435e

References (2)

[14] G. Cavallaro, M. Mura, J.A. Benediktsson, L. Bruzzone 'A Comparison of Self-Dual Attribute Profiles based on different filter rules for classification', IEEE IGARSS2014, Quebec, Canada

[15] Fran Berman, Maximising the Potential of Research Data

[16] Human Brain Project, online: https://www.humanbrainproject.eu/de

[17] Facebook Sharing Study, online: http://washington.cbslocal.com/2012/06/23/study-facebook-sharing-comparable-to-enjoyment-from-sex-food/

[18] Stanford Facebook like study, online:

http://news.stanford.edu/news/2015/january/personality-computer-knows-011215.html

[19] Big Data Market, Source Statista, online: http://www.statista.com/statistics/254266/global-big-data-market-forecast/

Acknowledgements

PhD Student Gabriele Cavallaro, University of Iceland

Tómas Philipp Runarsson, University of Iceland

Kristján Jonasson, University of Iceland

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

Selected Members of the Research Group on High Productivity Data Processing

Ahmed Shiraz Memon
Mohammad Shahbaz Memon
Markus Goetz
Christian Bodenstein
Philipp Glock
Matthias Richerzhagen

Thanks

Slides available at http://www.morrisriedel.de/talks