Selected Parallel and Scalable Methods for Scientific Big Data Analytics

Dr.-Ing. Morris Riedel et al.

Research Group Leader, Juelich Supercomputing Centre
Adjunct Associated Professor, University of Iceland

ZIH Kolloquium, 21th May 2015
Technical University of Dresden
JUELICH in Numbers

Area: 2.2 km²

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

**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

**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

- Full programmes taught in English
- Staff: ~1259
- Students: ~14,000
- Located in Reykjavik Capital Center, Iceland

---

**Teaching of key technologies in engineering & sciences**

**University Courses:** Statistical Data Mining & HPC-A/B
Outline

- Data Analytics @ Juelich
  - Driven by Scientific & Engineering Demands
  - Understanding of Terms & Key Focus
- Scalable & Parallel Tools
  - Clustering – DBSCAN
  - Classification – SVM
  - Scientific Applications in Context
- Recent Research Directions
  - ‘Brain Analytics‘
  - Deep Learning
- Conclusions
- References & Backup Slides
Data Analytics @ Juelich
Data Analytics – Context JSC

- Research data-intensive science and engineering applications
- Explore computing that is more intertwined with data analysis
- Tackle Inverse Problems
- Sharing, re-use, towards reproducibility
‘Data Analytics’ is an ‘interesting mix’ of different approaches

- Analytics: Whole methodology; Analysis: data investigation process itself
- ‘Big’ requires scalable processing methods and underlying infrastructure

Concrete ‘big data’: large medical data

Concrete ‘big data’: large earth science data
Data Analytics – Research Key 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

Scientific Applications

‘Big Data’ Methods

Classification++
Clustering++
Regression++

Systematic & Automated Analytics guided by CRISP – DM

‘(Big) Data from various data science applications’
Data Analytics – Selected Research Group Activities

John von Neumann Institute for Computing (NIC)
- Peer-review of scientific big data analytics (SBDA) proposals
- Jointly work with SBDA users (first projects starting, prototyping process)

Research Data Alliance (RDA)
- Chairing activities of the Big Data Analytics Interest Group
- Collaboration with a variety of EU and US partners
- Geoffrey Fox, UoIndiana (map-reduce), Kuo Kwo-Sen (NASA, SciDB)

Smart Data Innovation Lab (SDIL)
- Driving activities in the personalised medicine community (with Bayer)
- Collaboration with partners from industry (e.g. IBM, SAP, Siemens, etc.)
Data Analytics – Selected Research Expertise

Key expertise making algorithms parallel & scalable for ‘big data’

- Driven by scientific and engineering cases, e.g. understanding the human brain, remote sensing applications, marine measurements analysis, …
- Automate and/or support the data analysis process
- Example codes: Density-based Spatial Clustering of Applications with Noise (DBSCAN), Support Vector Machines (SVMs),

**Parallel & Scalable DBSCAN clustering tool**

**Parallel & Scalable SVM classification tool**

Problem: Automatic outlier detection for data quality
- Tailor solution for community
- Scalability towards Big Data
- Design and improve automatic data analytics approaches

Problem: Classification of buildings from multi-spectral images
- Enable smooth transition from ‘manual Matlab SVM scripts’
- Research on parallel SVM methods (map-reduce, HPC)


Scalable & Parallel Tools: Clustering
Learning From Data – Clustering Technique

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

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

**Regression**
- Identify a line with a certain slope describing the data
Selected Clustering Methods

K-Means Clustering – Centroid based clustering
- Partitions a data set into K distinct clusters (centroids can be artificial)

K-Medoids Clustering – Centroid based clustering (variation)
- Partitions a data set into K distinct clusters (centroids are actual points)

Sequential Agglomerative hierarchic nonoverlapping (SAHN)
- Hierarchical Clustering (create tree-like data structure → ‘dendrogram’)

Clustering Using Representatives (CURE)
- Select representative points / cluster; as far from one another as possible

Density-based spatial clustering of applications + noise (DBSCAN)
- Reasoning: density similarity measure helpful in our driving applications
- Assumes clusters of similar density or areas of higher density in dataset
# Technology Review of Open & Available Tools

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

*M. Goetz, M. Riedel et al., 6th Workshop on Data Mining in Earth System Science, International Conference of Computational Science (ICCS), Reykjavik, to be published*
Parallel & Scalable DBSCAN MPI/OpenMP Tool (1)

DBSCAN 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
- Identifies outliers/noise

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 $minPoints$
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

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

Performance Comparisons

- With another open-source parallel DBSCAN implementation (aka ‘NWU’)
- 3,705,635 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)

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)

<table>
<thead>
<tr>
<th>Computation time</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSC-HPDBSCAN</td>
<td>117.18 s</td>
<td>59.64 s</td>
<td>30.68 s</td>
<td>16.25 s</td>
<td>10.86 s</td>
<td>9.39 s</td>
</tr>
<tr>
<td>NWU-PDSDBSCAN</td>
<td>288.35 s</td>
<td>162.47 s</td>
<td>105.94 s</td>
<td>89.87 s</td>
<td>85.37 s</td>
<td>88.42 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Speed-Up</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>JSC-HPDBSCAN</td>
<td>1.00 x</td>
<td>1.96 x</td>
<td>3.82 x</td>
<td>7.21 x</td>
<td>10.79 x</td>
<td>12.48 x</td>
</tr>
<tr>
<td>NWU-PDSDBSCAN</td>
<td>1.00 x</td>
<td>1.77 x</td>
<td>2.72 x</td>
<td>3.21 x</td>
<td>3.38 x</td>
<td>3.26 x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>JSC-HPDBSCAN</td>
<td>251,064 MB</td>
<td>345,276 MB</td>
<td>433,340 MB</td>
<td>678,248 MB</td>
<td>1,101 GB</td>
<td>2,111 GB</td>
</tr>
<tr>
<td>NWU-PDSDBSCAN</td>
<td>500,512 MB</td>
<td>725,104 MB</td>
<td>1,370 GB</td>
<td>4,954 GB</td>
<td>19,724 GB</td>
<td>59,685 GB</td>
</tr>
</tbody>
</table>

[6] Open PANGAEA Earth Science Data Collection
Parallel & Scalable DBSCAN MPI/OpenMP Tool (5)

Free tool available
- Public bitbucket account – open-source
- Tool Website with more information
- Maintained on best effort basis

Usage via simple jobscripts

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

Parameter \( \epsilon \) and \( \text{minPoints} \)

Parallel & Scalable DBSCAN MPI/OpenMP Tool (6)

Usage via jobs script

- 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

```bash
mriedel@judge:/home/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=/home/zam/analytic/bigdata/hpdbscan/gcc-4.9.2/lib64:$LD_LIBRARY_PATH
DBSCAN=/home/zam/analytic/bigdata/hpdbscan/jsc_mpi/dbscan
SMALLBREMENDATA=/home/zam/analytic/bigdata/hpdbscan/jsc_mpi/mriruns/bremenSmall.h5

cd /home/zam/analytic/bigdata/hpdbscan/jsc_mpi/mriruns
mpilexec -rp 1 $DBSCAN -e 300 -m 100 -t 12 $SMALLBREMENDATA
```
Parallel & Scalable DBSCAN MPI/OpenMP Tool (7)

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

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

```python
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([data, clusters.reshape((-1,1))], axis=1)

clusters[clusters!=0] = 1

print "Write PCD"
with open(FILE, "w") as out:
    out.write("""
    VERSION 0.7
    FIELDS x y z rgb noise
    SIZE 4 4 4 4
    TYPE F F F F I I
    COUNT 1 1 1 1 1
    WIDTH 1
    HEIGHT 1
    VIEWPOINT 0 0 0 1 0 0 0
    POINTS %d
    DATA ascii
    """
    np.savetxt(out, data)
```
Earth Science Application
‘Automated outlier detection in time series’
- Collaboration with MARUM, Bremen (work in progress)
- Example: water quality data of Koljofjords
- Connected underwater device
- Measurements: oxygen, temperature, salinity, ...

Use of HPBSCAN algorithm
- Detect outliers and anomalies/events (e.g. water mixing)
- Compare with manually annotated data by domain-scientist
- Needs automation
Parallel & Scalable DBSCAN MPI/OpenMP Tool (10)

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

Use of HPBSCAN algorithm
- First 2d results detect various clusters
- Work in progress, not very good results
- Approach: Several iterations (with 3D) with potentially different parameter values
- Investigate other methods (e.g. OPTICS)

Research activities jointly with T. Dickscheid et al. (Juelich Institute of Neuroscience & Medicine)
Parallel & Scalable DBSCAN MPI/OpenMP Tool (11)

Neuroscience Application – Work in progress (e.g. 3120x3288)
‘Cell nuclei detection and tissue clustering‘ – varying parameters

Research activities jointly with T. Dickscheid et al. (Juelich Institute of Neuroscience & Medicine)
Scalable & Parallel Tools: Classification
Learning From Data – Classification Technique

Classification

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

Clustering

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

Regression

- Identify a line with a certain slope describing the data
Selected Classification Methods

Perceptron Learning Algorithm – simple linear classification
   - Enables binary classification with ‘a line’ between classes of separable data

Support Vector Machines (SVMs) – non-linear (‘kernel’) classification
   - Enables non-linear classification with maximum margin (best ‘out-of-the-box’)
   - Reasoning: achieves often better results than other methods in remote sensing application

Decision Trees & Ensemble Methods – tree-based classification
   - Grows trees for class decisions, ensemble methods average n trees

Artificial Neural Networks (ANNs) – brain-inspired classification
   - Combine multiple linear perceptrons to a strong network for non-linear tasks

Naive Bayes Classifier – probabilistic classification
   - Use of the Bayes theorem with strong/naive independence between features
## Technology Review of Open & Available Tools

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

*M. Goetz, M. Riedel et al., 6th Workshop on Data Mining in Earth System Science, International Conference of Computational Science (ICCS), Reykjavik, to be published*
SVM Algorithm Approach

- 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 \( N \times N \)

\[
\mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \langle x_n^T x_m \rangle
\]

(\text{max. hyperplane } \rightarrow \text{dual problem, using quadratic programming method})

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

\[
y_i (w \cdot x_i - b) \geq 1 - \xi_i, \quad \xi_i \geq 0
\]

(\text{mini}\text{mizing hyperplane turned into optimization problem, minimization, dual problem})

\[
k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle
\]

(\text{kernel trick, quadratic coefficients – Computational Complexity & Big Data Impact})
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

Understanding the MPI tool parameters

- Selecting non-linear kernel function $K$ type as RBF $\rightarrow$ parameter $-t \ 2$
- Setting RBF Kernel configuration parameter $\gamma$ $\rightarrow$ e.g. parameter $-g \ 16$
- Setting SVM allowed errors parameter $\rightarrow$ 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, C, 2011
- Message Passing Interface (MPI)
- New version appeared 2014-10 v. 1.3 (no major improvements)
- Lack of ‘big data‘ support (memory, layout, etc.)

Tuned scalable parallel piSVM tool 1.2.1

- Open-source (repository to be created)
- Based on piSVM tool 1.2
- Optimizations: load balancing; MPI collectives
- Contact: m.richerzhagen@fz-juelich.de

Parallel & Scalable SVM MPI Tool (4)

Classification Study of Land Cover Types

Reference Data Analytics for reusability & learning

CRISP-DM Report
Openly Shared Datasets
Running Analytics Code

<table>
<thead>
<tr>
<th>Class</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buildings</td>
<td>18126</td>
<td>163129</td>
</tr>
<tr>
<td>Blocks</td>
<td>10982</td>
<td>98834</td>
</tr>
<tr>
<td>Roads</td>
<td>16353</td>
<td>147176</td>
</tr>
<tr>
<td>Light Train</td>
<td>1606</td>
<td>14454</td>
</tr>
<tr>
<td>Vegetation</td>
<td>6962</td>
<td>62655</td>
</tr>
<tr>
<td>Trees</td>
<td>9088</td>
<td>81792</td>
</tr>
<tr>
<td>Bare Soil</td>
<td>8127</td>
<td>73144</td>
</tr>
<tr>
<td>Soil</td>
<td>1506</td>
<td>13551</td>
</tr>
<tr>
<td>Tower</td>
<td>4792</td>
<td>43124</td>
</tr>
<tr>
<td>Total</td>
<td>77542</td>
<td>697859</td>
</tr>
</tbody>
</table>

Satellite Data (Quickbird)

Parallel Support Vector Machines (SVM)

HPC / MPI

[10] Rome Image dataset
Parallel & Scalable SVM MPI Tool (5)

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

[Rome Image dataset](#)
Labelled data available for train/test data

- Groundtruth data of 9 different land-cover classes available

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)

<table>
<thead>
<tr>
<th>Class</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buildings</td>
<td>18126</td>
<td>163129</td>
</tr>
<tr>
<td>Blocks</td>
<td>10982</td>
<td>98834</td>
</tr>
<tr>
<td>Roads</td>
<td>16353</td>
<td>147176</td>
</tr>
<tr>
<td>Light Train</td>
<td>1606</td>
<td>14454</td>
</tr>
<tr>
<td>Vegetation</td>
<td>6962</td>
<td>62655</td>
</tr>
<tr>
<td>Trees</td>
<td>9088</td>
<td>81792</td>
</tr>
<tr>
<td>Bare Soil</td>
<td>8127</td>
<td>73144</td>
</tr>
<tr>
<td>Soil</td>
<td>1506</td>
<td>13551</td>
</tr>
<tr>
<td>Tower</td>
<td>4792</td>
<td>43124</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>77542</strong></td>
<td><strong>697859</strong></td>
</tr>
</tbody>
</table>

[10] Rome Image dataset
Parallel & Scalable SVM MPI Tool (7)

Based on ‘LibSVM data format‘ (using feature extraction method)

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

![Area](image)

<table>
<thead>
<tr>
<th>Class</th>
<th>Feature</th>
<th>Gray Level</th>
<th>Std Dev</th>
<th>Moment of Inertia</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Buildings</td>
<td>3:0.105882</td>
<td>0:1.09804</td>
<td>3:0.101961 ......</td>
</tr>
<tr>
<td></td>
<td>Blocks</td>
<td>2:0.364706</td>
<td>2:0.360784</td>
<td>3:0.356863 ......</td>
</tr>
<tr>
<td></td>
<td>Roads</td>
<td>6:1.0152941</td>
<td>2:0.34902</td>
<td>3:0.454902 ......</td>
</tr>
<tr>
<td></td>
<td>Light Train</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td></td>
<td>Vegetation</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td></td>
<td>Trees</td>
<td>9:0.247059</td>
<td>2:0.247059</td>
<td>3:0.227451 ......</td>
</tr>
<tr>
<td></td>
<td>Bare Soil</td>
<td>7:1.0411765</td>
<td>2:0.411765</td>
<td>3:0.415686 ......</td>
</tr>
<tr>
<td></td>
<td>Soil</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td></td>
<td>Tower</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
</tbody>
</table>

Each line is a pixel

Number Feature: 55 features

Std Dev: 55 samples

Rome Image dataset

[10] Rome Image dataset

Usage via jobscript

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

```bash
#!/bin/bash
#MOAB -N Train-tune-rec06-4-16-32
#MOAB -l nodes=4:ppn=16:performance
#MOAB -l walltime=03:00:00
#MOAB -M m.riedel@fz-juelich.de
#MOAB -m abc
#MOAB -W x=accesspolicy:singlejob
#MOAB -v tpt=2
#MOAB -q devel

### jobscript

cd $PBS_O_WORKDIR
echo "workdir: $PBS_O_WORKDIR"

NSLOTS=32
echo "running on $NSLOTS cpus..."

### location
PISVM=/home/azs/mriedel/pisvm-1.2/pisvm-1.2/pisvm-train

TRAINDATA=/home/azs/mriedel/bigdata/86-romeok/sdap_area_all_training.sl

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

→ Usage via simple jobscripts

SVM Parameters

[12] Rome Analytics Results & job scripts
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)
Another more challenging dataset: high number of classes

- Parallelization challenges: unbalanced class representations

G. Cavallaro, M. Riedel et al., Remote Sensing Journal – Big Data Special Issue, to be published

[20] Indian pines dataset, processed and raw
Another example dataset: high number of classes

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

Manual work: Trade-off between raw data processing and using feature extraction methods

Can we automate feature extraction mechanism to some degree?

G. Cavallaro, M. Riedel et al., Remote Sensing Journal – Big Data Special Issue, to be published

[21] Analytics Results (raw)
[22] Analytics Results (processed)
Parallel & Scalable SVM MPI Tool (12)

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

- Evaluation between Matlab (aka serial) and parallel piSVM
- 10x cross-validation (RBF kernel parameter and C, gridsearch)

G. Cavallaro, M. Riedel et al., Remote Sensing Journal – Big Data Special Issue, to be published

[23] Analytics 10 fold cross-validation Results (raw)
[24] Analytics 10 fold cross-validation Results (processed)
Recent Research Directions
Recent Research Directions – Brain Data Classification

1. Some ‘pattern’ exists
   - Image content classification (e.g. SVMs, RandomForst, etc.)

2. No exact mathematical formula exists
   - No precise formula for ‘contour of the brain’

3. Dataset (next: 5 brains, >100,000 pixels, 2PB raw)
   - Block face images (of frozen brain 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

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

➢ Research activities jointly with T. Dickscheid et al. (Juelich Institute of Neuroscience & Medicine)
Recent Research Directions – Deep Learning

1. Some ‘pattern’ exists
   - Image content classification & clustering

2. No exact mathematical formula exists
   - No precise formula for ‘brain layers’

3. Dataset – raw images exist
   - Needs to be properly prepared
   - Generate labeled data to learn from (manual tool supporting scientists)
   - Use Deep Learning (deep convolutional neural network, GPGPUs) to classify cell nuclei
   - Extract cell nuclei into 2D/3D point cloud
   - Cluster different brain areas by cell density (parallel DBSCAN)

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

Scientific Peer Review is essential to progress in the field

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

Selected SBDA benefit from parallelization

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

Number of Data Analytics et al. Technologies incredible high

- Thorough analysis and evaluation hard (needs different infrastructures)
- (Less) open source & working versions available, often paper studies
- Still evaluating approaches: HPC, map-reduce, Spark, SciDB, MaTex, …
References
[1] R. Huber, M. Riedel et al., ‘PANGAEA – Data Publisher for Earth & Environmental Science - Research data enters scholarly communication and big data analysis’, presentation at EGU 2014, Vienna
[6] PANGAEA Earth Science Data Collection, Online: http://www.pangaea.de/
[8] An Introduction to Statistical Learning with Applications in R, Online: http://www-bcf.usc.edu/~gareth/ISL/index.html
[9] Original piSVM tool, online: http://pisvm.sourceforge.net/
[10] B2SHARE data collection Rome Dataset, online: http://hdl.handle.net/11304/4615928c-e1a5-11e3-8cd7-14feb57d12b9
[12] 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] UNICORE, online: http://www.unicore.eu
[20] B2SHARE data collection, remote sensing indian pines images, Online: http://hdl.handle.net/11304/7e8eec8e-ad61-11e4-ac7e-860aa0063d1f
[21] B2SHARE data collection, piSVM remote sensing indian pines analytics results (raw), Online: http://hdl.handle.net/11304/c06a8c7e-fe6c-11e4-8a18-f31aa6f4d448
[22] B2SHARE data collection, piSVM remote sensing indian pines analytics results (processed), Online: http://hdl.handle.net/11304/c528999e-ff7c-11e4-8a18-f31aa6f4d448
[23] B2SHARE data collection, Analytics 10 fold cross-validation (raw), Online: http://hdl.handle.net/11304/163ba8e8-fe60-11e4-8a18-f31aa6f4d448
[24] B2SHARE data collection, Analytics 10 fold cross-validation (processed), Online: http://hdl.handle.net/11304/5bba8e36-fe63-11e4-8a18-f31aa6f4d448
Acknowledgements

PhD Student Gabriele Cavallaro, University of Iceland
Tómas Philipp Runarsson, University of Iceland
Kristján Jonasson, University of Iceland

Timo Dickscheid, 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
Selected Backup Slides for Discussions
Distributed Large-scale Data Management
In-Situ Analytics for HPC & Exascale

Exascale computer with access to exascale storage/archives

In-situ correlations & data reduction

Correlations

e.g. map-reduce jobs, R-MPI

In-situ statistical data mining

e.g. clustering, classification

analytics part visualization part computational simulation part

visual analytics
distributed archive
key-value pair DB
interactive
Scalable I/O
in-memory

Exascale application

[15] Inspired by ASCAC DOE report
Tools for Large-scale Distributed Data Management

- Useful tools for data-driven scientists & HPC users

Need for Sharing & Reproducability in HPC – Example

- Sharing different datasets is key
- One tend to lose the overview of which data is stored on which platform
- How do we gain trust to delete data when duplicates on different systems exist

- bachelor thesis activities, e.g. improving code (same data)
- Research & PhD thesis activities & papers, and....

Teach class with good AND bad examples!

- Student Classes
- Bachelor thesis Student
- PhD Student
- Professor
- another collaborator

B2SHARE: Store and Share Research Data
Smart Data Analytics Process

Big Data Analytics
- Simple Data Preprocessing
- Automated Parallel Data Analytics
- Data Postprocessing
- Data Analysis

Traditional Data Analysis
- Manual Feature Reduction
- Manual Feature Extraction
- Manual Feature Selection

Concrete Datasets (& source/sensor)
(parallel) Algorithms & Methods
Technologies & Ressources

Scientific Data Applications

Combine both: Smart Data Analytics

Time to solution

Concrete Datasets


CRISP-DM report

"Reference Data Analytics"
for reusability & learning

Report for joint Usage
Openly Shared Datasets
Running Analytics Code

Manual Feature Extraction
Manual Feature Selection

Manual Feature Reduction

Smart Data Analytics Process

JSC: Data Analytics: m.riedel@fz-juelich.de
Selected Research Data Alliance (RDA) Activities

- Big Data Analytics Interest Group – Establish something like UCI machine learning repository, but for big data analytics...


Satellite Data (Quickbird)
Parallel Support Vector Machines (SVM)
HPC & MPI

Classification Study of Land Cover Types

Research activities with Gabriele Cavallaro (PhD thesis, UoIceland) on Self Dual Attribute Profile
Reproducability Example in Data-driven Science (1)

- Having this tool available on the Web helps tremendously to save time for no research tasks
- Using the tool enables to focus better on the research tasks
Reproducability Example in Data-driven Science (2)

- Sharing pre-processed data
- LibSVM format
- Training and Testing Datasets
- Different setups for analysis (SDAP on All or SDAP on Panchromatic)
Reproducability Example in Data-driven Science (3)

Simple download from http using the wget command

```
mriedel@judge:~/bigdata> ls -al
```

```
total 640
drwxrwxrwx 21 mriedel zam 32768 2014-09-17 22:20`
drwxr-xr-x 19 mriedel zam 32768 2014-09-18 11:49 ,
drwxr-xr-x 2  mriedel zam 32768 2014-06-19 07:17 102-salinasindian
```

Well defined directory structures

- Simple Download from http using wget
- ...before adopting B2SHARE regularly
Reproducability Example in Data-driven Science (4)

Make a short note in your directory linking back to B2SHARE

- Enables the trust to delete data if necessary (working against big data)
- Link back to B2SHARE for quick checks and file that links back fosters trust

mrriedel@judge:~/bigdata> cd 86-romeok/
mrriedel@judge:~/bigdata/86-romeok> ls -al
total 580320
  drwxr-xr-x   2 mriedel zam       512 2014-07-09 11:03 .
  drwxr-xr-x 21 mriedel zam 32768 2014-06-17 22:20 
-rw-r--r--  1 mriedel zam     35 2014-07-09 11:01 b2share.txt
-rw-r--r--  1 mriedel zam 410974972 2014-05-22 13:26 sdap_area_all_test.el
-rw-r--r--  1 mriedel zam   45662874 2014-05-22 13:36 sdap_area_all_training.el
-rw-r--r--  1 mriedel zam 114763982 2014-05-22 13:36 sdap_area_panch_test.el
-rw-r--r--  1 mriedel zam 12745692 2014-05-22 13:36 sdap_area_panch_training.el
mrriedel@judge:~/bigdata/86-romeok> more b2share.txt
https://b2share.eudat.eu/record/86
mrriedel@judge:~/bigdata/86-romeok>
Reproducability Example in Data-driven Science (5)

True reproducability needs: (1) datasets; (2) technique parameters (here for SVM); and (3) correct versions of algorithm code.
Classical Machine Learning

Dealing with Big Data in traditional Machine Learning

- Define Features to learn from ?!
- Transform data into supported format ?!
- How to reduce dimensions ?!
- How to parallelize ?!

![Diagram of data processing flow](Image)
Deep Learning (2)

Deep Learning

Dealing with Big Data in Deep Learning

- Define Features to learn from
  - Automatically learn how to define features
- Transform data into supported format
  - Adopt the model to your data
- How to reduce dimensions
  - Automatically reduce dimensions in every hidden layer
- How to parallelize
  - Naturally the brain is parallel, so Artificial Neural Networks are!

- A. Ng, Google Brain
Deep Learning in Computational Biomedicine

Genome Analysis
- Find high level features on low level –omics data

Medical Image Analysis
- Use 2D (or 3D) structure of the data for classification

Unstructured Data Analysis
- Use DL for text analysis to classify patient data, drug recommendations by users, …

Etc…
Deep Learning Packages

There exists several frameworks for deep neural networks

- **Pylearn2**
  - Python tool on the top of the Theano python library
  - Easy configuration of data, model, learning via YAML files
  - CUDA support for accelerated calculations
  - Jobman for parallel cross validation

- **Caffe**
  - C++ implementation with python & matlab wrappers
  - CUDA acceleration

- **DL4J**
  - Java implementation of Deep Learning
  - CUDA + Hadoop support
Chances and Pitfalls for ‘Scientific Big Data Analytics’

~2009 – H1N1 Virus Made Headlines
- Nature paper from Google employees
- Explains how Google is able to predict fast winter flu
- Not only on national scale, but down to regions
- Possible via logged big data – ‘search queries’

~2014 – The Parable of Google Flu
- Large errors in flu prediction & lessons learned
  (1) Dataset: Transparency & replicability impossible
  (2) Study the algorithm since they keep changing
  (3) It’s not just about size of the data

- Big data is not always better data – Think about difference of causality vs. correlation
Location-based Social Network-based Health Analytics

Scientific Domain Area

- Smart Cities approaches combined with Health Analytics Research

Scientific Outcome

- Traffic density estimation
- Network emission model

Location-based Social Networks (LBSN) Data

- Open data sources: Twitter & Foursquare
- Plan: Validation with real measurements in cities

➢ Research activities with Markus Goetz (PhD thesis) – Juelich Supercomputing Centre, UOlceland