Parallel & Scalable Machine Learning

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 4

Unsupervised Clustering, Challenges & Solutions

January 15th, 2018
JSC, Germany
Review of Lecture 3

- **Unsupervised Clustering**
  - Input \( x = x_1, \ldots, x_d \)
  - No output (unsupervised)
  - Data \( (x_1), \ldots, (x_N) \)

- **K-Means Clustering**
  - Problem of assigning \( K \) as numbers of clusters
  - Simple, works only sometimes

- **DBSCAN Clustering**
  - Finds arbitrary shapes and numbers of clusters
  - Still two parameters
  - Parallel version scales for big pointcloud datasets

(serial statistical computing with R tool)  
(no guiding output)

(number of cluster can be amiguities sometimes)

[1] An Introduction to Statistical Learning
4. **Unsupervised Clustering Challenges & Solutions**
Outline

- Parallel & Scalable Clustering
  - Using HPDBSCAN for Bremen Big Dataset
  - Selected HPDBSCAN Implementation Details
  - Domain Decomposition
  - Sorting & Index
  - Hybrid Code

- Clustering Challenges & Solutions
  - Adjust Walltime to Complexity & Size
  - Change Number of Nodes
  - Change Parameters Epsilon
  - Change Parameters minPoints
  - Twitter Dataset & Interpretation
Parallel & Scalable Clustering
Bremen Dataset & Locations – Revisited

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

The Bremen Dataset is encoded in the HDF5 format (binary)
- You need your own copy of the file in your home directory to cluster!
Exercises – Bremen Big HPDBSCAN Runs
HPC Environment – Modules Revisited

- **Module environment tool**
  - Avoids to manually setup environment information for every application
  - Simplifies shell initialization and lets users easily modify their environment

- **Module avail**
  - Lists all available modules on the HPC system (e.g. compilers, MPI, etc.)

- **Module spider**
  - Find modules in the installed set of modules and more information

- **Module load** \(\rightarrow\) **needed before HPDBSCAN run**
  - Loads particular modules into the current work environment, E.g.:

  ```bash
  [train001@jrl12 ~]$ module load GCC
  
  Due to MODULEPATH changes, the following have been reloaded:
  1) binutils/.2.29
  
  The following have been reloaded with a version change:
  1) GCCcore/.5.4.0 => GCCcore/.7.2.0
  
  [train001@jrl12 ~]$ module load ParaStationMPI/5.2.0-1
  [train001@jrl12 ~]$ module load HDF5/1.8.19
  ```
JURECA HPC System – HPDBSCAN Job Script

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

export OMP_NUM_THREADS=4

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

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

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

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

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

Note the tutorial reservation with `-reservation=ml-hpc-1` just valid for today on JURECA

(parameters of DBSCAN and file to be clustered)
The outcome of the clustering process is written directly into the HDF5 file using cluster IDs and noise IDs.
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

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

HPDBSCAN – Scaling

- **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


(DS1 = Bremen; DS2 = Twitter)
Clustering Challenges & Solutions
Bremen Big Dataset – ‘Running against the Wall’ (1)

- Configured walltime
  - 1:00 hour in jobscript; 2 nodes (4 tasks per node)
- Check job status (shortly before the hour)
  ```sh
  [train001@jrl07 hpdbscan]$ squeue -u train001
  JOBID PARTITION     NAME       USER   ST       TIME  NODES NODELIST(REASON)
  4629896  batch    HPDBSCAN train001 R       59:36    2 jrc[1250-1251]
  ```
- Job gets automatically cancelled by scheduler
  ```sh
  [train001@jrl07 hpdbscan]$ squeue -u train001
  JOBID PARTITION     NAME       USER   ST       TIME  NODES NODELIST(REASON)
  4629896  batch    HPDBSCAN train001 CG    1:00:27    2 jrc[1250-1251]
  ```

In parallel & scalable machine learning one needs to adjust the walltimes of jobs to the complexity in processing time and/or size of the dataset (cf. Bremen small vs. Bremen big)

- Determining the right amount of walltime is not easy and mostly be best obtained by test runs
- The required walltime depends on the number of used nodes (and tasks) and is directly linked
Bremen Big Dataset – ‘Running against the Wall’ (2)

- **Check outcome of the job**
  
  ```bash
  [train001@jrl07 hpdbscan]$ more HPDBSCAN-4629896.out
  Calculating Cell Space...
  Computing Dimensions... [OK] in 0.043040
  Computing Cells... [OK] in 0.157041
  Sorting Points... [OK] in 1.041985
  Distributing Points... [OK] in 2.126353
  DBSCAN...
  Local Scan...
  ```

- **Check error report of the job**
  
  ```bash
  [train001@jrl07 hpdbscan]$ more HPDBSCAN-4629896.err
  HDF5-DIAG: Error detected in HDF5 (1.8.19) MPI-process 0:
  #000: H5F.c line 772 in H5Fclose(): not a file ID
  major: Invalid arguments to routine
  minor: Inappropriate type
  error: *** step 4629896 CANCELLED DUE TO TIME LIMIT ***
  srun: Job step aborted: Waiting up to 6 seconds for job step to finish.
  srun: error: irc1250: tasks 0-1: Terminated
  ```

- The partial result of clustering when terminated is not useful and should be not used anymore
- In case of termination by scheduler even HDF problems might occur that corrupt the file
Exercises – Increasing Number of Nodes
JURECA HPC System – HPDBSCAN Job Script

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

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

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

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

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

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

Note the tutorial reservation with `--reservation=ml-hpc-1` just valid for today on JURECA
Exercises – Changing Epsilon & MinPoints Parameters
JURECA HPC System – HPDBSCAN Job Script

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

export OMP_NUM_THREADS=4

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

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

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

srun $HPDBSCAN -m 100 -e 300 -t 12 $BREMENSMAILDATA

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

Note the tutorial reservation with –reservation=ml-hpc-1 just valid for today on JURECA
Exercises – Twitter Dataset
Twitter Dataset & Locations – Revisited

- Twitter streaming API data
  - Contains 1% of all geo-tagged of the UK in June 2014 (e.g. London)

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

```
[train001@jrl04 twitter]$ pwd
/homea/hpclab/train001/data/twitter
[train001@jrl04 twitter]$ ls -al
total 317312
drwxr-xr-x 2 train001 hpclab 512 Jan 14 23:00 .
drwxr-xr-x 8 train001 hpclab 512 Jan 14 22:06 ..
-rw-r--r-- 1 train001 hpclab 265636608 Jan 13 2017 twitter.h5
-rw-r--r-- 1 train001 hpclab 59272032 Jan 13 2017 twitterSmall.h5
```
#!/bin/bash
#SBATCH --job-name=HPDBSCAN
#SBATCH -o HPDBSCAN-%j.out
#SBATCH -e HPDBSCAN-%j.err
#SBATCH --nodes=4
#SBATCH --ntasks=4
#SBATCH --ntasks-per-node=4
#SBATCH --time=01:00:00
#SBATCH --cpus-per-task=4
#SBATCH --reservation=ml-hpc-1

export OMP_NUM_THREADS=4

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

# your own copy of bremen small
TWITTERSMALLDATA=/homea/hpclab/train001/twitterSmall.h5

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

srun $HPDBSCAN -m 40 -e 0.0001 -t 12 $TWITTERSMALLDATA (parameters of DBSCAN and file to be clustered)

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

Note the tutorial reservation with –reservation=ml-hpc-1 just valid for today on JURECA.
The outcome of the clustering process is written directly into the HDF5 file using cluster IDs and noise IDs.

```
[train001@jrl04 hpdbscan]$ more HPDBSCAN-4632910.out
Calculating Cell Space...
  Computing Dimensions... [OK] in 0.002393
  Computing Cells...     [OK] in 0.498816
  Sorting Points...      [OK] in 0.891462
  Distributing Points... [OK] in 2.576206
DBSCAN...
  Local Scan...          [OK] in 1.375779
  Merging Neighbors...   [OK] in 0.000586
  Adjust Labels...       [OK] in 0.013686
  Rec. Init. Order...    [OK] in 0.640681
  Writing File...        [OK] in 0.006232
Result...
  8976  Clusters
  906807 Cluster Points
  2797544 Noise Points
  757369 Core Points
Took: 6.189666s
```
Lecture Bibliography
Lecture Bibliography

- [1] An Introduction to Statistical Learning with Applications in R,
  Online: http://www-bcf.usc.edu/~gareth/ISL/index.html
  Online: http://www.wikicfp.com/cfp/servlet/event.showcfp?eventid=46948
  Online: http://hdl.handle.net/11304/6eacaa76-c275-11e4-ac7e-860aa0063d1f
  Online: http://hdl.handle.net/11304/6eacaa76-c275-11e4-ac7e-860aa0063d1f
Slides Available at http://www.morrisriedel.de/talks