Comparison of clustering techniques for hybrid rocket fuel combustion data

International Workshop on Scientific Machine Learning
Cologne, January 8-10, 2020

Alexander Rüttgers*, Charlotte Debus*, Martin Siggel*
Anna Petrarolo**, Mario Kobald**

*Institute for Simulation and Software Technology
** Institute for Space Propulsion
German Aerospace Center (DLR)
Motivation (ATEK research rocket flight in June 2019)

https://www.youtube.com/watch?v=JlcReUwZXFU
Outline

1. Rocket engine combustion analysis at DLR
2. Helmholtz Analytics Toolkit (HeAT) for distributed ML
3. Clustering results with HeAT
Outline

1. Rocket engine combustion analysis at DLR
2. Helmholtz Analytics Toolkit (HeAT) for distributed ML
3. Clustering results with HeAT
Rocket engine combustion analysis

- **Aim:** Cost reduction of rocket engines, be competitive with e.g. Space-X

**Traditional liquid rocket engine:**

- 2 pumps transporting fluid fuel and oxidizer at very high pressure and flow
- **Advantages**
  - Burning rate can be controlled precisely
- **Disadvantages**
  - Pumps are mechanically very complex
  - Expensive
Rocket engine combustion analysis

- **Aim:** Cost reduction of rocket engines, be competitive with e.g. Space-X

**Solid propellant rocket engine**

- Fuel and oxidizer are mixed in solid form
- Advantage
  - Cheap
- Disadvantage
  - Burning rate can not be varied during flight
Rocket engine combustion analysis

- **Aim**: Cost reduction of rocket engines, be competitive with e.g. Space-X

**Hybrid rocket engine**

- Pressurized fluid oxidizer
- Solid fuel
- A valve controls, how much oxidizer gets into the combustion chamber

- Advantages
  - Cheap
  - Controllable
Project ATEK: Experiments on new hybrid rocket fuels at DLR

- DLR investigates new hybrid rocket fuels on a paraffin basis at Institute of Space Propulsion in Lampoldshausen.

- About 300 combustion tests were performed with single-slab paraffin-based fuel with 20° forward facing ramp angle + gaseous oxygen.

- Two different fuel compositions:
  - pure paraffin 6805
  - paraffin 6805 + 5% polymer

Fig. 1: Fuel slap configuration before (top) and after (bottom) combustion test.
Combustion chamber set-up

• Optically accessible combustion chamber is 450 mm long, 150 mm wide and 90 mm high.

• Tests were performed with different configurations (e.g. fuel, oxidizer mass flow, filters)

• Combustion is captured with high-speed video camera with 10 000 frames / second

Fig. 2: Side view of combustion chamber

Fig. 3: Test matrix used for data analysis
<table>
<thead>
<tr>
<th>Video extract of test 284</th>
<th>fuel</th>
<th>oxidizer mass flow</th>
<th>CH*-filter</th>
<th>duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignition, steady</td>
<td>pure paraffin 6805</td>
<td>50 g/s,</td>
<td>yes, only wavelengths emitted from CH* are filmed</td>
<td>3 s = 30 000 frames / 8GB raw data per test</td>
</tr>
<tr>
<td>combustion, extinction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Clustering of combustion image data

- Clustering of combustion data = identify different phases of the flow.

- Various clustering algorithms exist in the literature (DBSCAN, spectral clustering, k-means, …).

- **Start:** Comparison of algorithms on two features \((\mu, \bar{x})_j\) for all \(j = 1, \ldots, 30000\) images of test 284.
Comparison of clustering algorithms for presented application

<table>
<thead>
<tr>
<th></th>
<th>K-means</th>
<th>Spectral clustering</th>
<th>DBSCAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>approach</td>
<td>- Iteratively minimize the within-cluster sum of squares</td>
<td>- Construct similarity matrix $A$ of size ($nr_of_points \times nr_of_points$)</td>
<td>- Find points in $\varepsilon$-environment of every point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Build graph Laplacian matrix $L = D - A$ with diagonal matrix $D_{ii} = \sum_j A_{ij}$</td>
<td>- If environment contains enough $minPts$ points, a new cluster is started</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Compute first K eigenvectors of $L$</td>
<td>- Otherwise, it’s noise or belongs to other cluster.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cluster low-dimensional data representation with e.g. K-means</td>
<td></td>
</tr>
<tr>
<td>pros*</td>
<td>- Scales to large data sets</td>
<td>- Reduces curse of dimensionality</td>
<td>- Does not require number of clusters $K$</td>
</tr>
<tr>
<td></td>
<td>- Guarantees convergence</td>
<td>- Does not make strong assumptions on clusters (e.g. spherical shape)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Very simple</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cons*</td>
<td>- Choosing $K$ manually</td>
<td>- Choosing $K$ manually</td>
<td>- Two hyperparameters $\varepsilon$ and $minPts$ that are hard to find</td>
</tr>
<tr>
<td></td>
<td>- Local optimal solutions</td>
<td>- Expensive for large datasets</td>
<td>if dataset is continuous</td>
</tr>
<tr>
<td></td>
<td>- Curse of dimensionality</td>
<td>- Number of hyperparameters</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Similar-size clustering</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*incomplete list
Comparison of clustering algorithms for presented application

<table>
<thead>
<tr>
<th><strong>K-means (start first)</strong></th>
<th><strong>Spectral clustering (second approach)</strong></th>
<th><strong>DBSCAN (here not adequate)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>approach</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Iteratively minimize the within-cluster sum of squares</td>
<td>• Construct similarity matrix $A$ of size (nr_of_points)$\times$(nr_of_points)</td>
<td>• Find points in $\varepsilon$-environment of every point</td>
</tr>
<tr>
<td></td>
<td>• Build graph Laplacian matrix $L = D - A$ with diagonal matrix $D_{ii} = \sum_j A_{ij}$</td>
<td>• If environment contains enough $\text{minPts}$ points, a new cluster is started</td>
</tr>
<tr>
<td></td>
<td>• Compute first $K$ eigenvectors of $L$</td>
<td>• Otherwise, it’s noise or belongs to other cluster.</td>
</tr>
<tr>
<td></td>
<td>• Cluster low-dimensional data representation with e.g. K-means</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>pros</strong>*</th>
<th><strong>Spectral clustering</strong></th>
<th><strong>DBSCAN</strong>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Scales to large data sets</td>
<td>• Reduces curse of dimensionality</td>
<td>• Does not require number of clusters $K$</td>
</tr>
<tr>
<td>• Guarantees convergence</td>
<td>• Does not make strong assumptions on clusters (e.g. spherical shape)</td>
<td></td>
</tr>
<tr>
<td>• Very simple</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>cons</strong>*</th>
<th><strong>Spectral clustering</strong></th>
<th><strong>DBSCAN</strong>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Choosing $K$ manually</td>
<td>• Choosing $K$ manually</td>
<td>• Two hyperparameters $\varepsilon$ and $\text{minPts}$ that are hard to find if dataset is continuous</td>
</tr>
<tr>
<td>• Local optimal solutions</td>
<td>• Expensive for large datasets</td>
<td></td>
</tr>
<tr>
<td>• Curse of dimensionality</td>
<td>• Number of hyperparameters</td>
<td></td>
</tr>
<tr>
<td>• Similar-size clustering</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*incomplete list
Outline

1. Rocket engine combustion analysis at DLR

2. Helmholtz Analytics Toolkit (HeAT) for distributed ML

3. Clustering results with HeAT
HeAT

• **HeAT = Helmholtz Analytics Toolkit**

• Python framework for **parallel, distributed** data analytics and machine learning

• Developed within the Helmholtz Analytics Framework Project since 2018

• **Aim:** Bridge data analytics and **high-performance computing**

• Open Source licensed, MIT

  [helmholtz-analytics/heat](https://github.com/helmholtz-analytics/heat)
How we started HeAT:
The Helmholtz Analytics Framework (HAF) Project

• Joint project of all 6 Helmholtz centers

• Goal: foster data analytics methods and tools within Helmholtz federation.

• Scope:
  • Development of domain-specific data analysis techniques
  • Co-design between domain scientists and information experts
Motivation: HAF applications
Scope

Facilitating applications of HAF in their work

Bringing HPC and Machine Learning / Data Analytics closer together

Ease of use

Design

**HeAT**
- k-means
- SVM
- Deep Learning
- And more machine learning algorithms

**PyTorch**
- Tensor Linear Algebra
- Automatic Differentiation
- NumPy-like interface
- GPU support

**mpi4py**
- Distributed Parallelism (MPI)
Data Distribution

Example:

```python
import heat as ht

# construct a range tensor

>>> range_data = ht.arange(6, split=1)

>>> range_data.mean()

2.5

>>> range_data.argmax()

5
```
What has been done so far?

- The core technology has been identified
- Implementation of a distributed parallel tensor core framework
- NumPy-compatible core functionality
- Some linear algebra routines
- Parallel data I/O via HDF 5 and NETCDF
- K-means and spectral clustering algorithms are available
Outline

1. Rocket engine combustion analysis at DLR
2. Helmholtz Analytics Toolkit (HeAT) for distributed ML
3. Clustering results with HeAT
K-means clustering: Strategies to avoid its drawbacks

- **Avoid local optimum solutions**
  - Algorithm is run multiple times (here: 10-times)
  - Take solution with smallest objective function (not a big difference in our case)
  - Implementation of K-Means++*
    - Choose the initial centers less randomly

- **Selection of K in K-means?**
  - Detailed analysis of objective function depending on $K$
    (here: algorithm is used for $K= 2, \ldots, 10$)
  - Runtime of algorithm scales at least linearly in $K$
  - Note that an optimal $K$ is often problem dependent

Clustering allows for quantitative comparison.

Apart from final cluster, all other clusters represent long-running flow phases.

**Fig. 4:** Distribution of frames to their corresponding clusters.

**Fig. 5:** Time length of each cluster [s].
Test 284 with K=7  (Part 1/3)

cluster 1
(1320 / 30000 frames)
ignition phase
(ignition comes from bottom of the chamber)

cluster 2
(2942 / 30000 frames)
burn phase without energy from outside
(ignition valves closed)

cluster 3
(3493 / 30000 frames)
fuel slap burns in the middle
(oxygen mass flow increases)
Test 284 with K=7 (Part 2/3)

cluster 4
(3493 / 30000 frames)
whole surface is burning (brightness decreases due to $\text{CH}^*+\text{O}_2 = \text{CO}+\text{OH}^*$)

cluster 5
(2452 / 30000 frames)
large side flame close to camera

cluster 6
(16980 / 30000 frames)
constant combustion (with low CH* concentration, largest cluster in time)
Test 284 with K=7  (Part 3/3)

cluster 7
(194 / 30000 frames)
flame extinguishing phase (oxygen valve closes, nitrogen purge)

What about short-term irregularities?
Increasing the number of clusters K?

![Chart showing overlapping clusters over time.]

**Test Case:** 284 (20 clusters)

**Solution Strategies:**

- Cluster recombination / data postprocessing
- Different clustering approach (e.g., spectral clustering)
Spectral clustering: Strategies to avoid (some of) its drawbacks

• Expensive for large datasets
  • Usage of HeAT on HPDA-cluster at DLR
    • Distributed algorithm for similarity matrix computation
    • Implementation of distributed Lanczos algorithm for eigenvalue computation
  • Spectral clustering on 150 processes on 3 nodes took about 1 hour / test

• Large number of hyperparameters?
  • First results with HeAT have been achieved
  • Use scikit-learn + scikit-optimize / auto-sklearn / … on simplified problem (i.e. fewer images) to accelerate hyperparameter estimation
Fig. 6: Similarity matrix of all tests using a Gaussian kernel with variance $\sigma = 30000^2$. 

irregularities in similarity matrix
Fig. 7: 20 smallest eigenvalues of the graph Laplacian of all four tests. The number of 0 eigenvalues of the graph Laplacian corresponds to number of connected components.

Hyperparameter optimization of test 284

Fig. 8a: Spectral clustering with K=7 and affinity matrix from Gaussian kernel with $\sigma = 30000^2$

Fig. 8b: Analogous clustering with $\sigma = 28000^2$

real turbulent structures but not resolved in own cluster
Conclusion and outlook

- Clustering of combustion image data with K-means and spectral clustering using HeAT on HPDA-cluster at DLR possible within a reasonable amount of time.

- Analysis of turbulent combustion tests in combustion chamber allows a quantitative test comparison.

- Future work: Focus on anomaly detection and more adequate analysis techniques.

- Further details:

Thank you for your attention!
backup slides
NumPy

Data structure
-ND-Tensor

Operations
- Elementwise operations
- Slicing
- Matrix operations
- Reduction
## PyTorch

<table>
<thead>
<tr>
<th>Runs on</th>
<th>Data structure</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>ND-Tensor</td>
<td>- Elementwise operations</td>
</tr>
<tr>
<td>or GPU</td>
<td>shape: (4, 3, 2)</td>
<td>- Slicing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Matrix operations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Reduction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Automatic differentiation</td>
</tr>
</tbody>
</table>
HeAT

Runs on

MPI

or

Data structure

ND-Tensor

Operations

- Elementwise operations
- Slicing
- Matrix operations
- Reduction
- Automatic differentiation