JPlex

CSRI Workshop on Combinatorial Algebraic Topology
August 29, 2009
Henry Adams
JPlex computes the persistent homology of a filtered simplicial complex.

- **Plex**: Vin de Silva and Patrick Perry (2000-2006)
- **JPlex**: Harlan Sexton and Mikael Vejdemo-Johansson (2008-present)
Input

A filtered simplicial complex is a nested sequence of simplicial complexes.

\[ K^1 \subseteq K^2 \subseteq \ldots \subseteq K^m \]
Input

A filtered simplicial complex is a nested sequence of simplicial complexes.

\[ K^1 \subseteq K^2 \subseteq \ldots \subseteq K^m \]
A Betti barcode is a collection of (possibly semi-infinite) intervals and is one way to describe the persistent homology of a filtered simplicial complex.
Functoriality is more informative than Betti numbers alone. Significant features persist.
How?

- *Computing Persistent Homology* by Afra Zomorodian and Gunnar Carlsson (2005)

- Taking persistent homology over field $\mathbb{F}$ gives a finitely generated graded module over the PID $\mathbb{F}[t]$.

\[
\left( \bigoplus_{i=1}^{n} \sum^{\alpha_i} \mathbb{F}[t] \right) \oplus \left( \bigoplus_{j=1}^{m} \sum^{\gamma_{ij}} \mathbb{F}[t]/(t^{n_{ij}}) \right)
\]

- Compute intervals by putting boundary matrices in Smith normal form - in fact, just column echelon form.
Persistent homology applications

• Topological simplification
  (Edelsbrunner, Letscher, Zomorodian)

• Point cloud data analysis
  – image patches
    (Carlsson, Ishkhanov, de Silva, Zomorodian)
  – neuron firings
    (Singh, Memoli, Ishkhanov, Sapiro, Carlsson, Ringach)

• Shape recognition
  (Carlsson, Collins, Guibas, Zomorodian)

• Edge or corner identification
  (E. Carlsson, G. Carlsson, de Silva)
JPlex capabilities

• Streamlined for
  – Persistence algorithm
  – Rips and Witness complexes

• Does not return homology generators.

• Field of coefficients with prime $p < 256$ (data type char)

• Simplex dimensions $< 8$. 
Scalability

• $O(n^3)$, where $n$ is number of simplices.

• JPlex timings (same for $\mathbb{Z}_2$ through $\mathbb{Z}_{251}$)

<table>
<thead>
<tr>
<th>$n$</th>
<th>length</th>
<th>seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>11,151</td>
<td>$\approx 300$</td>
<td>0.28</td>
</tr>
<tr>
<td>62,180</td>
<td>$\approx 300$</td>
<td>9.81</td>
</tr>
</tbody>
</table>

• Faster $\mathbb{Z}_2$ implementation in *Computing Persistent Homology*

<table>
<thead>
<tr>
<th>$n$</th>
<th>length</th>
<th>seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>12,000</td>
<td>1,020</td>
<td>$&lt; 0.01$</td>
</tr>
<tr>
<td>3,029.383</td>
<td>$\approx 300$</td>
<td>50.23</td>
</tr>
</tbody>
</table>
JPlex and Plex-2.5

• Memory problems fixed
• Easy installation
• MATLAB and BeanShell interface
• Reduced toolset
  – cohomology
  – sequential maxmin landmarks
  – union, intersection
  – what do you miss?
Future plans

• Upcoming release with cohomology
• A better platform for research-focused development of techniques?
JPlex usability

- Available http://comptop.stanford.edu/
### JPlex usability

- Javadoc tree

### Package edu.stanford.math.plex

#### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCPlot</td>
<td>The BCPlot class does simple plotting of PersistenceIntervals.</td>
</tr>
<tr>
<td>Chain</td>
<td>A Chain instance is an element of the module constructed by taking formal sums of ring elements times simplices.</td>
</tr>
<tr>
<td>CRC</td>
<td>The CRC class provides good hash methods for int/long and string data.</td>
</tr>
<tr>
<td>DiscreteSpace</td>
<td>The DiscreteSpace class implements a finite discrete metric space.</td>
</tr>
<tr>
<td>DistanceData</td>
<td>The DistanceData class is the simplest implementation of NSpace.</td>
</tr>
<tr>
<td>EuclideanArrayData</td>
<td>The EuclideanArrayData class is the simplest implementation of NSpace.</td>
</tr>
<tr>
<td>ExplicitStream</td>
<td>A ExplicitStream instance is SimplexStream whose elements are given explicitly, along with associated values of a persistence parameter.</td>
</tr>
<tr>
<td>ExplicitStream.DComplex</td>
<td>A DComplex (for Dimensional Complex) is the set of simplices (and associated persistence parameters) of specified dimension for an ExplicitStream.</td>
</tr>
<tr>
<td>LazyWitnessStream</td>
<td>A LazyWitnessStream is a SimplexStream whose elements are the simplices of the lazy Witness complex of a PointData instance.</td>
</tr>
<tr>
<td>MappedBufferData</td>
<td>The MappedBufferData class is another very simple implementation of NSpace.</td>
</tr>
</tbody>
</table>
JPlex usability

- Available http://comptop.stanford.edu/
- Javadoc tree
- Tutorials: toy examples, real example, exercises

3.3. Subclass ExplicitStream and persistent homology.
Let's build a stream with persistent 9-dimensional cycles. We build a house with the upper triangle at time 0, the top vertex at time 1, the red edges at times 2 and 3, and the red 2-simplex at time 5.

```scala
>>> house = ExplicitStream;
>>> house.add([[1, 2, 3, 4, 5], [2, 0, 0, 0, 1]]);
>>> house.add([[1, 2, 2, 3, 3, 4, 4], [1, 1, 5, 0, 5], [0, 0, 0, 2, 3]]);
>>> house.add([3, 4, 5, 7]);

We compute the 4th homology.

>>> house.close
>>> intervals = Flex.Persistence.computeIntervals(house);
```
Discussion

• Question: what other capabilities should be considered?

  – $O(nm^2)$
  – parallelizable