MATH:7450 (22M:305) Topics in Topology: Scientific and Engineering Applications of Algebraic Topology

Sept 27, 2013: Phom

Fall 2013 course offered through the University of Iowa Division of Continuing Education

Isabel K. Darcy, Department of Mathematics
Applied Mathematical and Computational Sciences,
University of Iowa

http://www.math.uiowa.edu/~idarcy/AppliedTopology.html
IMA Annual Program Year Workshop
Introduction to Statistics and Probability for Topologists
October 2-4, 2013

Wednesday | Thursday | Friday
Wednesday October 02, 2013

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Speaker</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30am-9:00am</td>
<td>Coffee</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:00am-10:15am</td>
<td>Statistical Inference for Topological Data Analysis</td>
<td>Alessandro Rinaldo (Carnegie Mellon University)</td>
<td>Lind Hall 305</td>
</tr>
<tr>
<td>10:15am-10:45am</td>
<td>Break</td>
<td></td>
<td>Lind Hall 400</td>
</tr>
<tr>
<td>10:45am-12:00pm</td>
<td>Lecture 2</td>
<td>Alessandro Rinaldo (Carnegie Mellon University)</td>
<td>Lind Hall 305</td>
</tr>
<tr>
<td>12:00pm-1:30pm</td>
<td>Lunch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30pm-2:45pm</td>
<td>Statistical Learning and Bayesian Inference and Topology</td>
<td>Sayan Mukherjee (Duke University)</td>
<td>Lind Hall 305</td>
</tr>
<tr>
<td>2:45pm-3:15pm</td>
<td>Break</td>
<td></td>
<td>Lind Hall 400</td>
</tr>
<tr>
<td>3:15pm-4:30pm</td>
<td>Lecture 2</td>
<td>Sayan Mukherjee (Duke University)</td>
<td>Lind Hall 305</td>
</tr>
</tbody>
</table>

Thursday October 03, 2013

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Speaker</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30am-9:00am</td>
<td>Coffee</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:00am-10:15am</td>
<td>Probability and Stochastic Processes for Persistent Homologists</td>
<td>Robert Adler (Technion-Israel Institute of Technology)</td>
<td>Lind Hall 305</td>
</tr>
<tr>
<td>10:15am-10:45am</td>
<td>Break</td>
<td></td>
<td>Lind Hall 400</td>
</tr>
<tr>
<td>10:45am-12:00pm</td>
<td>Lecture 2</td>
<td>Robert Adler (Technion-Israel Institute of Technology)</td>
<td>Lind Hall 305</td>
</tr>
<tr>
<td>Time</td>
<td>Event</td>
<td>Speaker</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------</td>
<td>----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>8:00am-8:45am</td>
<td>Registration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:45am-9:00am</td>
<td>Welcome</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:00am-9:50am</td>
<td>Lecture</td>
<td>Gunnar Carlsson (Stanford University)</td>
<td></td>
</tr>
<tr>
<td>9:50am-10:15am</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:15am-11:05am</td>
<td>Geometric Methods for Heterogeneous Data</td>
<td>Susan Holmes (Stanford University)</td>
<td></td>
</tr>
<tr>
<td>11:05am-11:30am</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30am-12:20pm</td>
<td>The Geometry and Topology of the Cosmic Web</td>
<td>Rien van de Weijgaert (Rijksuniversiteit te Groningen)</td>
<td></td>
</tr>
<tr>
<td>12:20pm-2:00pm</td>
<td>Lunch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:00pm-2:50pm</td>
<td>Back to Basics: Merge Trees</td>
<td>Dmitriy Morozov (Lawrence Berkeley National Laboratory)</td>
<td></td>
</tr>
<tr>
<td>2:50pm-3:15pm</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:15pm-4:05pm</td>
<td>Letting Loops Loose</td>
<td>Yuriy Mileyko (University of Hawaii at Manoa)</td>
<td></td>
</tr>
<tr>
<td>4:05pm-5:00pm</td>
<td>Discussion</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scientific Resources

The IMA offers several valuable scientific resources to the talk materials and videos, the IMA volumes, and the IMA press.

Videos
Each lecture that is given at the IMA is recorded and available later archived on our website for future viewing. These videos are accessible to the general public.

Talk Materials
A schedule of events, lists of organizers and participants,
Media Live Streaming (available only during the workshop)

- Live Streaming from Keller 3180
- Live Streaming from Lind Hall 305

IMA Annual Program workshops and tutorials
IMA Hot Topics workshops and special events
IMA Public Lectures
IMA Seminars on Industrial Problems
IMA Postdoc Seminars
phom: Persistent Homology in R

This package computes persistent homology of filtered simplicial complexes, and provides facilities for constructing complexes from geometric data.

Version: 1.0.2
Depends: Rcpp (≥ 0.9.7)
LinkingTo: Rcpp
Published: 2013-04-30
Author: Andrew Tausz
Reference manual: phom.pdf
Vignettes: User's manual
What is R?

Introduction to R

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.
CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: main page, windows release, windows old release.

0-Cloud

http://cran.rstudio.com/

Rstudio, automatic redirection to servers worldwide

Argentina

http://mirror.fcaglp.unlp.edu.ar/CRAN/

http://r.mirror.mendoza-conicet.gob.ar/

Universidad Nacional de La Plata

CONICET Mendoza

Australia

http://cran.csiro.au/

CSIRO
Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for (Mac) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms
<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Directory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>debian/</td>
<td>30-Aug-2013 04:27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>redhat/</td>
<td>25-Nov-2009 12:31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>suse/</td>
<td>16-Feb-2012 09:39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ubuntu/</td>
<td>05-May-2013 09:32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
R for Mac OS X

R-3.0.2.pkg (latest version)
MD5-hash: e693d62ae27a97ac61baf0858ab61346 (ca. 64MB)

R 3.0.2 binary for Mac OS X 10.6 (Snow Leopard) and higher, signed package. Contains R 3.0.2 framework, R.app GUI 1.62 in 64-bit for Intel Macs. The above file is an Installer package which can be installed by double-clicking.
R for Windows

Subdirectories:

- base
  Binaries for base distribution (managed by Duncan Murdoch). This is what you want to install R for the first time.

- contrib
  Binaries of contributed packages (managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.

- Rtools
  Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.
R-3.0.1 for Windows (32/64 bit)

Download R 3.0.1 for Windows (52 megabytes, 32/64 bit)

Installation and other instructions
New features in this version
getwd()
setwd("Documents/CompTOP/Data")
list.files()

points <- read.table("5yrMs.csv",sep="",""
points = read.table("5yrMs.csv",sep="",""
points[1,2]

help()
help(matrix)

install.packages("phom")
phom: Persistent Homology in R

This package computes persistent homology of filtered simplicial complexes, and provides facilities for constructing complexes from geometric data.

Version: 1.0.2
Depends: Rcpp (≥ 0.9.7)
LinkingTo: Rcpp
Published: 2013-04-30
Author: Andrew Tausz
Reference manual: phom.pdf
Vignettes: User's manual
Start R and type:

```r
> install.packages("phom")
also installing the dependency ‘Rcpp’
```

trying URL 'http://streaming.stat.iastate.edu/CRAN/bin/macosx/contrib/3.0/'
Create points on a circle:

\[ t \leftarrow 2 \times \pi \times \text{runif}(100) \]
\[ x \leftarrow \cos(t); \ y \leftarrow \sin(t) \]
\[ X \leftarrow t(\text{as.matrix}(\text{rbind}(x, y)))) \]
\[ \text{plot}(X) \]

\[
\text{library(phom)} \\
\text{max\_dim} \leftarrow 1 \\
\text{max\_f} \leftarrow 0.6 \\
\text{intervals} \leftarrow \text{pHom}(X, \text{max\_dim}, \text{max\_f}) \\
\text{plotPersistenceDiagram(intervals, max\_dim, max\_f, title=\"Random Points on S^1 with Euclidean Norm\")} \\
\text{plotBarcodeDiagram(intervals, max\_dim, max\_f, title='')} \\n\]
pHom(X,
dimension,
max filtration value,
mode = "vr",
metric = "euclidean",
p = 2,
landmark set size = 2* ceiling(sqrt(length(X))),
maxmin samples = min(1000, length(X))
pHom(X, dimension, max filtration value, mode = “vr”, metric = “euclidean”, p = 2, landmark set size = 2* ceiling(sqrt(length(X))), maxmin samples = min(1000, length(X)))

**metric**

This indicates the type of metric that will be used.
metric
This indicates the type of metric that will be used.

Valid choices include the following: "distance_matrix", "euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski".

When "distance_matrix" is specified, the parameter "X" is interpreted as a distance matrix.

Otherwise, "X" is regarded as a set of points, where each row is one point.
$\text{pHom}(X,\newline \text{dimension,}\
\text{max filtration value,}\
\text{mode} = \text{“vr”},\
\text{metric} = \text{“euclidean”},$

$p = 2,$

$\text{landmark set size} = 2^* \text{ceiling}(\sqrt{\text{length}(X)}),$

$\text{maxmin samples} = \min(1000, \text{length}(X))$
\text{pHom}(X, \text{dimension, max filtration value, mode = “vr”, metric = “euclidean”, p = 2, landmark set size = 2}^* \text{ ceiling} (\sqrt{\text{length}(X)})), \text{maxmin samples = min}(1000, \text{length}(X)))

\text{X}

Data matrix consisting of either data points OR distances between data points.
A matrix with 1 of the 2 following interpretations.

If \texttt{metric = "distance\_matrix"}, then
\(X\) is required to be a square matrix whose entries indicate the distance between two points. The number of rows (and columns) equals the number of points in the dataset.

If \texttt{metric is NOT "distance\_matrix"}, then \(X\) is required to be a matrix in which the rows are points in Euclidean space. The number of columns is the dimensionality of the dataset.
pHom(X, dimension, max filtration value, mode = “vr”, metric = “euclidean”, p = 2, landmark set size = 2* ceiling(sqrt(length(X))), maxmin samples = min(1000, length(X)))

dimension
The maximum dimension to compute persistent homology to.
\text{pHom}(X, \\
dimension, \\
max filtration value, \\
mode = "vr", \\
metric = "euclidean", \\
p = 2, \\
landmark set size = 2* \text{ceiling}(\sqrt{\text{length}(X)}), \\
maxmin samples = \min(1000, \text{length}(X)))
pHom(X, dimension, max filtration value, mode = "vr", metric = "euclidean", p = 2, landmark set size = 2* ceiling(sqrt(length(X))), maxmin samples = min(1000, length(X)))

mode
"vr" (default) = Vietoris-Rips
"lw" = lazy-witness
Phom lazy witness complex \( \text{LW}(D, \varepsilon, k) \): 

Let \( D = \) set of point cloud data points. 
Choose \( L \subseteq D, \ L = \) set of landmark points. 

Let \( m_k(v) = \text{dist}(v, d) \) where \( d \) is the \( k^{\text{th}} \) closest point in \( D \) to \( v \). 

\( \{l_1, l_2\} \) is a 1-simplex

iff

\[ \max\{d(v, l_1), d(v, l_1)\} \leq m_k(v) + \varepsilon \]

\( v \) is the witness
mode
This indicates the type of filtration to use.

2 possible choices are
"vr" (default) indicates that the Vietoris-Rips filtration will be used,
"lw" indicates that the lazy-witness construction will be used.

For Vietoris-Rips filtrations, the parameters landmark_set_size and maxmin_samples are ignored.
pHom(X, dimension, max filtration value, mode = “vr”, metric = “euclidean”, p = 2, landmark set size = 2* ceiling(sqrt(length(X))), maxmin samples = min(1000, length(X)))

The number of points to include in the landmark set. A sensible value for this is in between 20 and 100. The default value is taken to be 2p|X|. This parameter is only relevant for the lazy-witness filtration.
pHom(X, dimension, max filtration value, mode = "vr", metric = "euclidean", p = 2, landmark set size = 2* ceiling(sqrt(length(X))), maxmin samples = min(1000, length(X)))

The number of samples to use when performing the maxmin selection. The default value is taken to be min(|X|, 1000). This parameter is only relevant for the lazy-witness filtration.
plotBarcodeDiagram(intervals, dimension, max f, title="Persistence Diagram")

intervals
A matrix with three columns that specifies the persistence intervals. Entries in the first column indicate the dimension of an interval. The entries in the second and third columns indicate the start and end points of the intervals, respectively. The function pHom produces outputs that are in this form.
plotBarcodeDiagram(intervals, dimension, max f, title=“Persistence Diagram”)

dimension
The dimension to plot intervals for. Unlike plotPersistenceDiagram, this function only plots intervals for one dimension, and not all of them.

max f
The maximum filtration value to use in the diagram.

title
The title on the barcode diagram.
plotPersistenceDiagram(intervals, \texttt{max dim}, \texttt{max f},
\texttt{title=”Persistence Diagram”})

\textbf{intervals}
A matrix with three columns that specifies the persistence intervals. Entries in the first column indicate the dimension of an interval. The entries in the second and third columns indicate the start and end points of the intervals, respectively. The function pHom produces outputs that are in this form.

\textbf{max dim}
The maximum dimension to plot.

\textbf{max f}
The maximum filtration value to use in the persistence diagram.

\textbf{title}
The title on the persistence diagram.
persistence diagram = set of points in the plane where x-coordinate = the start of interval, y-coordinate = the ending point.
plotPersistenceDiagram(intervals, max dim, max f, title="Persistence Diagram")

intervals
A matrix with three columns that specifies the persistence intervals. Entries in the first column indicate the dimension of an interval. The entries in the second and third columns indicate the start and end points of the intervals, respectively. The function pHom produces outputs that are in this form.

max dim
The maximum dimension to plot.

max f
The maximum filtration value to use in the persistence diagram.

title
The title on the persistence diagram.
Choosing Landmark points:

A.) Random

B.) Maxmin

1.) choose point $l_1$ randomly

2.) If $\{l_1, ..., l_{k-1}\}$ have been chosen, choose $l_k$ such that $\{l_1, ..., l_{k-1}\}$ is in $D - \{l_1, ..., l_{k-1}\}$ and

$$\min \{d(l_k, l_1), ..., d(l_k, l_{k-1})\} \geq \min \{d(v, l_1), ..., d(v, l_{k-1})\}$$
num_landmark_points = 100;

random_selector = api.Plex4.createRandomSelector(point_cloud, num_landmark_points);

maxmin_selector = api.Plex4.createMaxMinSelector(point_cloud, num_landmark_points);
Javaplex Witness complex $W(D, L, \varepsilon)$:

Let $D =$ set of point cloud data points. Choose $L \subseteq D$, $L =$ set of landmark points.

Let $m_k(v) = \text{dist}(v, l)$ where $l$ is the $k+1$ closest point in $L$ to $v$.

\[
\{l_1, ..., l_{k+1}\} \text{ is a } k\text{-simplex iff } \\
d(v, l_i) \leq m_k(v) + \varepsilon \text{ for all } i
\]

$v$ is the witness
```python
>> num_landmark_points = 50;
>> max_dimension = 3;
>> num_divisions = 100;

>> landmark_selector = api.Plex4.createMaxMinSelector(point_cloud, num_landmark_points);

>> random_selector = api.Plex4.createRandomSelector(point_cloud, num_landmark_points);
```
The next command returns the landmark covering measure $R$ from Section 5.2. Often the value for $t_{\text{max}}$ is chosen in proportion to $R$.

$$>	ext{R} = \text{landmark}\_\text{selector}.\text{getMaxDistanceFromPointsToLandmarks}()$$

$R = 0.7033 \%$ Generally close to 0.7

$$>	ext{max}\_\text{filtration}\_\text{value} = \text{R} / 8;$$
We create the witness stream.

```python
>> stream = api.Plex4.createWitnessStream(
    landmark_selector, max_dimension,
    max_filtration_value, num_divisions);

>> num_simplices = stream.getSize()
num_simplices = 1164 % Generally close to 1200
Topological analysis of population activity in visual cortex

Gurjeet Singh
Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, USA

Facundo Memoli
Department of Mathematics, Stanford University, Stanford, CA, USA

Tigran Ishkhanov
Department of Mathematics, Stanford University, Stanford, CA, USA

Guillermo Sapiro
Department of Electrical and Computer Engineering, University of Minnesota, Minnesota, MN, USA

Gunnar Carlsson
Department of Mathematics, Stanford University, Stanford, CA, USA

Dario L. Ringach
Departments of Neurobiology and Psychology, Jules Stein Eye Institute, David Geffen School of Medicine, University of California, Los Angeles, CA, USA
The primary visual cortex (V1)

1.) V1 is a component in the visual pathway, which begins with
   • the retinal cells in the eye,
   • proceeds through the lateral geniculate locus,
   • then to the primary visual cortex,
   • and then through a number of higher level processing units, such as V2, V3, V4, middle temporal area (MT), and others

2.) V1 performs low level tasks, such as edge and line detection, and its output is then processed for higher level and larger scale properties further along the visual pathway. However, the mechanism by which it carries out these tasks is not understood.
Studied via method 2: embedded electrode arrays

• Record voltages at points in time at each electrode.
• Spike train: lists of firing times for a neuron
  • obtained via spike sorting – i.e. signal processing.
• Data = an array of N spike trains.
• Compared spontaneous (eyes occluded) to evoked (via movie clips).
• 10 second segments broken into 50 ms bins
• The 5 neurons with the highest firing rate in each ten second window were chosen
• For each bin, create a vector in $\mathbb{R}^5$ corresponding to the number of firings of each of the 5 neurons.
• 200 bins = 200 data points in $\mathbb{R}^5$
• Many 10 second segments = many data sets
Topological equivalence in rubber-world.

Singh G et al. J Vis 2008;8:11

©2008 by Association for Research in Vision and Ophthalmology
Betti numbers provide a signature of the underlying topology.

Singh G et al. J Vis 2008;8:11
Figure 4 animation
Experimental recordings in primary visual cortex.

Singh G et al. J Vis 2008;8:11
Estimation of topological structure in driven and spontaneous conditions.

- Record voltages at points in time at each electrode.
- Spike train: lists of firing times for a neuron
  - obtained via spike sorting – i.e. signal processing.
- Data = an array of N spike trains.
- Compared spontaneous (eyes occluded) to evoked (via movie clips).
- 10 second segments broken into 50 ms bins
  - Transition between states about 80 ms
- The 5 neurons with the highest firing rate in each ten second window were chosen
- For each bin, create a vector in \( \mathbb{R}^5 \) corresponding to the number of firings of each of the 5 neurons.
- 200 bins = 200 data points in \( \mathbb{R}^5 \).
  - Used 35 landmark points.
- 20-30 minutes of data = many data sets
- Control: shuffled data 52700 times.
Testing the statistical significance of barcodes.

Singh G et al. J Vis 2008;8:11