Reproducibility and Provenance

CS-GY 9223 Massive Data Analysis Course - Fall 2014
Presenter: Fernando Chirigati
Data Analysis and Mining

• In exploratory tasks, *change is the norm!*
  • Data analysis and mining are iterative processes
  • Many trial-and-error steps, easy to get lost, ...
  • We need the ability to *reproduce* past results!

Figure modified from J. van Wijk, IEEE Vis 2005
Reproducibility...? What? Why?
Reproducibility

• Good science requires reproducibility

“If I have seen further, it is by standing on the shoulders of giants.”
Isaac Newton

• Reproducibility goes beyond verifying the correctness of results
  • It is crucial for science to move forward – science is incremental
  • It helps newcomers
  • It increases impact, visibility [Vandewalle et al., IEEE Signal Processing Magazine 2009] and research quality [Begley and Ellis, Nature 2012]
  • It increases your productivity!

• “Without reproducibility, people die!”
  [John Wilbanks, AMPS Workshop on Reproducibility 2011]
How Bright Promise in Cancer Testing Fell Apart

But the research at Duke turned out to be wrong. Its gene-based tests proved worthless, and the research behind them was discredited. Ms. Jacobs died a few months after treatment, and her husband and other patients’ relatives have retained lawyers.

Instead, as patients and their doctors try to make critical decisions about serious illnesses, they may be getting worthless information that is based on bad science.

Doctors say the heart of the problem is the intricacy of the analyses in this emerging field and the difficulty in finding errors. Even well-respected scientists often “oversee a
<table>
<thead>
<tr>
<th>Country</th>
<th>Coverage</th>
<th>30 or less</th>
<th>30 to 60</th>
<th>60 to 90</th>
<th>90 or above</th>
<th>30 or less</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>1946-2009</td>
<td>3.4</td>
<td>3.3</td>
<td>-2.0</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>UK</td>
<td>1946-2009</td>
<td>2.4</td>
<td>2.5</td>
<td>2.4</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>Sweden</td>
<td>1946-2009</td>
<td>3.6</td>
<td>2.9</td>
<td>2.7</td>
<td>n.a.</td>
<td>6.3</td>
</tr>
<tr>
<td>Spain</td>
<td>1946-2009</td>
<td>1.5</td>
<td>4.2</td>
<td>3.6</td>
<td>13.3</td>
<td></td>
</tr>
<tr>
<td>Portugal</td>
<td>1946-2009</td>
<td>4.8</td>
<td>2.5</td>
<td>0.3</td>
<td>n.a.</td>
<td>7.9</td>
</tr>
<tr>
<td>New Zealand</td>
<td>1946-2009</td>
<td>2.5</td>
<td>2.9</td>
<td>3.9</td>
<td>-7.9</td>
<td>2.6</td>
</tr>
<tr>
<td>Netherlands</td>
<td>1946-2009</td>
<td>4.1</td>
<td>2.7</td>
<td>1.1</td>
<td>n.a.</td>
<td>6.4</td>
</tr>
<tr>
<td>Norway</td>
<td>1946-2009</td>
<td>3.4</td>
<td>5.1</td>
<td>n.a.</td>
<td>5.4</td>
<td></td>
</tr>
<tr>
<td>Japan</td>
<td>1946-2009</td>
<td>7.0</td>
<td>4.0</td>
<td>1.0</td>
<td>0.7</td>
<td>7.0</td>
</tr>
<tr>
<td>Italy</td>
<td>1946-2009</td>
<td>5.4</td>
<td>2.1</td>
<td>1.8</td>
<td>1.0</td>
<td>5.6</td>
</tr>
<tr>
<td>Ireland</td>
<td>1946-2009</td>
<td>4.4</td>
<td>4.5</td>
<td>4.0</td>
<td>2.4</td>
<td>2.9</td>
</tr>
<tr>
<td>Greece</td>
<td>1946-2009</td>
<td>4.2</td>
<td>0.3</td>
<td>2.7</td>
<td>2.9</td>
<td>13.3</td>
</tr>
<tr>
<td>Germany</td>
<td>1946-2009</td>
<td>3.9</td>
<td>0.9</td>
<td>n.a.</td>
<td>3.2</td>
<td>5.4</td>
</tr>
<tr>
<td>France</td>
<td>1946-2009</td>
<td>4.9</td>
<td>2.7</td>
<td>2.5</td>
<td>n.a.</td>
<td>5.2</td>
</tr>
<tr>
<td>Finland</td>
<td>1946-2009</td>
<td>3.8</td>
<td>2.4</td>
<td>5.5</td>
<td>n.a.</td>
<td>7.0</td>
</tr>
<tr>
<td>Denmark</td>
<td>1950-2009</td>
<td>3.5</td>
<td>1.7</td>
<td>2.4</td>
<td>n.a.</td>
<td>5.6</td>
</tr>
<tr>
<td>Canada</td>
<td>1951-2009</td>
<td>1.9</td>
<td>3.6</td>
<td>4.1</td>
<td>2.6</td>
<td>n.a.</td>
</tr>
<tr>
<td>Belgium</td>
<td>1947-2009</td>
<td>5.2</td>
<td>3.3</td>
<td>3.8</td>
<td>5.7</td>
<td></td>
</tr>
<tr>
<td>Austria</td>
<td>1947-2009</td>
<td>4.2</td>
<td>4.2</td>
<td>2.6</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td>Australia</td>
<td>1951-2009</td>
<td>3.2</td>
<td>4.9</td>
<td>4.0</td>
<td>n.a.</td>
<td></td>
</tr>
</tbody>
</table>

In the empirical, historical, and theoretical literature, we find no consistent evidence that high levels of public debt cause higher inflation. This is in contrast to the typical public–private debate in emerging market countries, where high public debt levels coincide with higher inflation.

Our topic would seem to be a timely one. Public debt has been soaring in the wake of the recent global financial crisis, and the economic crisis in the Eurozone and beyond. Have governments and private entities been able to manage their financial obligations, and historic circumstances.
How?
Provenance

- Provenance: a key ingredient for reproducibility

  “The source or origin of an object; its history and pedigree; a record of the ultimate derivation and passage of an item through its various owners.”

  *The Oxford English Dictionary*

- It helps determine the value, accuracy and authorship of an object

- It is used in many fields: works of art and antiques, archives and books, *science*, ...
Provenance is not new...

... but now we have larger volumes of data and more complex analyses!
Computational Provenance
Computational Provenance

How were these images created?
Which data and parameters were used?
Were there any pre and post-processing steps?
What is the difference between these images?
Is the procedure correct?
Who created these images?
Computational Provenance

Provenance to the rescue!

Interpretability
Understanding
Reproducibility

Data Quality
Verification of Procedures
Attribution and Ownership
Provenance is Always a DAG

I1 = readFile(/Users/fchirigati/input)
O1 = vtkStructuredReader(input=I1)
O2 = vtkContourFilter(value=57,input=O1)
O3 = vtkContourFilter(value=90,input=O1)
...

/Users/fchirigati/input

readFile

I1

vtkStructuredReader

57

O1

vtkContourFilter

90

vtkContourFilter

...
Provenance for Reproducibility

- A computational experiment composed by a sequence of steps $S$ that has been developed at time $T$, on environment $E$, and on data $D$ is \textit{reproducible} if it can be executed with a sequence of steps $S' \subseteq S$ at time $T' > T$, on environment $E'$ (different or the same as $E$), and on data $D$ (different or the same as $D$) with consistent results [Freire et al. 2012]

- We need three types of provenance
  - Description of data $D$
  - Description of environment $E$
  - Sequence of steps $S$
Axes of Reproducibility

- **Transparency Level**
  - How *transparent* the available provenance information is?
  - E.g.: source code provides a higher transparency level than binaries since the implementation can be explored and re-used

- **Portability**
  - In which environments the experiment can be reproduced?
  - E.g.: original environment, similar (compatible) environments, or different environments

- **Coverage**
  - How much of the original experiment can be reproduced?
  - E.g.: a step that depends on data from a third-party Web service may not be reproducible if the Web service becomes unavailable
Components for Reproducibility

- Longevity
- Portability
- Replicability and Modifiability
- Experiment Sharing
- Document Linkage
- Representation

**Reproducibility**
Components for Reproducibility

- Replicability and Modifiability
- Capture
- Experiment Sharing
- Document Linkage
- Longevity
- Portability
- Representation

REPRODUCIBILITY
Representation

• Creation of an specification for the experiment
• The specification can be
  • Descriptive only (non-executable)
    • Mostly useful for debugging and documentation
    • E.g.: process graph
  • Executable
    • Allows experiments to be re-executed by running the specification
    • E.g.: ad-hoc commands, makefiles, scripts, scientific workflows
• Different specifications require different provenance capture mechanisms
CFLAGS=-O
LDFLAGS=-s -lm
OBJS=modscag.o \ fileops.o mgsmix.o \ geochecks.o shdnorm.o

all: modscag modsorth

modscag: $(OBJS)  
  $(CC) $(LDFLAGS) \  
  -o modscag $(OBJS)

modsorth: modsorth.o  
  $(CC) $(LDFLAGS) \  
  -o modsorth \  
  modsorth.o
import vtk
data = vtk.vtkStructuredPointsReader()
data.SetFileName("../examples/data/head.120.vtk")
contour = vtk.vtkContourFilter()
contour.SetInput(0, data.GetOutput())
contour.SetValue(0, 67)
mapper = vtk.vtkPolyDataMapper()
mapper.SetInput(contour.GetOutput())
mapper.ScalarVisibilityOff()
actor = vtk.vtkActor()
actor.SetMapper(mapper)
cam = vtk.vtkCamera()
cam.SetViewUp(0, 0, -1)
cam.SetPosition(745, -453, 369)
cam.SetFocalPoint(135, 135, 150)
cam.ComputeViewPlaneNormal()
ren = vtk.vtkRenderer()
ren.AddActor(actor)
ren.SetActiveCamera(cam)
ren.ResetCamera()
renwin = vtk.vtkRenderWindow()
renwin.AddRenderer(ren)
style = vtk.vtkInteractorStyleTrackballCamera()
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renwin)
iren.SetInteractorStyle(style)
iren.Initialize()
iren.Start()
Components for Reproducibility

- Longevity
- Portability
- Replicability and Modifiability
- Capture
- Representation
- Experiment Sharing
- Document Linkage

REPRODUCIBILITY
Capture

• Capture of description of data, environment, and experiment
• Different granularities and techniques
  • **OS-Based Capture – makefiles, no source code**
    • Fine-grained provenance: manipulation of system calls and computational processes
    • Useful when no source code is available, or when there are multiple scripts and tools connected by data dependencies
    • Use of `ptrace`, `strace`, `SystemTap`, ...
  • **Data-Based Capture – source code, scripts, and data**
    • Fine-grained provenance: keeping track of changes over data (versioning)
    • Includes version control systems (for source code) and temporal databases (for tuples in database tables)
Capture

• Different granularities and techniques (cont.)
  • **Code-Instrumented Capture – source code and scripts**
    • Source code is instrumented to capture provenance
    • Granularity depends on how the instrumentation is done
    • It can be *manual* or *automatic*
Capture

- Different granularities and techniques (cont.)
  - Code-Instrumented Capture – source code and scripts
    - Source code is instrumented to capture provenance
    - Granularity depends on how the instrumentation is done
    - It can be manual or automatic

VCR [Gavish and Donoho, 2011]
Capture

- Different granularities and techniques (cont.)
  - **Code-Instrumented Capture – source code and scripts**
    - Source code is instrumented to capture provenance
    - Granularity depends on how the instrumentation is done
    - It can be *manual* or *automatic*

noWorkflow [Murta et al., 2014]
Capture

• Different granularities and techniques (cont.)
  • **Workflow-Based Capture – workflows**
    • Coarse-grained provenance: captured by *workflow management systems*
    • Workflow modeling defines the boundaries for the capture

O1 = vtkStructuredReader(input=I1)
O2 = vtkContourFilter(value=57,input=O1)
O3 = vtkDataSetMapper(input=O2)

[Davidson and Freire, SIGMOD 2008]
Components for Reproducibility

**REPRODUCIBILITY**

- Replicability and Modifiability
- Capture
- Representation
- Experiment Sharing
- Document Linkage
- Portability
- Longevity
Replicability

- Ability to repeat the experiment execution with the *same* parameters and data originally used
  - Data and parameters must be precisely captured
  - Particularly useful for publication reviews
- Replicability may be hampered by
  - Non-deterministic processes
  - Third-party services
  - Library updates
  - Stateful data (e.g., data in a database that is modified independently)
Modifiability

- Ability to *vary* parameters, data, and even the structure of the experiment
  - A detailed and manipulable specification must be provided (high transparency for the description of experiment)
  - Mostly useful to test the experiment under different inputs, and to allow others to build upon the work
  - E.g.: scientific workflows can be easily changed
Parameter Exploration in VisTrails

- isovalue=90
- isovalue=50
- isovalue=120
Components for Reproducibility

- Longevity
- Portability
- Replicability and Modifiability
- Capture
- Representation
- Experiment Sharing
- Document Linkage

**Reproducibility**
Portability

• Ability for the experiment to be reproduced in different environments
• Three levels:
  • *Low Portability*: experiment is only reproducible in the same original environment
    • No capture of description of the environment
    • E.g.: scientific workflows
  • *Medium Portability*: experiment is reproducible in environments similar to the original one
    • Capture of libraries and dependencies
    • E.g.: Linux packing tools (CDE, ReproZip)
  • *High Portability*: experiment is reproducible everywhere
    • Capture of the complete environment – e.g.: virtual machines
    • Remote access through a Web-based interface – e.g.: crowdLabs
Components for Reproducibility

- Longevity
- Portability
- Replicability and Modifiability
- Capture
- Representation
- Experiment Sharing
- Document Linkage

Reproducibility
Longevity

• Ability to reproduce the experiment long after it was created
• Related to the maintenance of reproducibility coverage
• Two main mechanisms
  • Longevity *by archiving*
    • All the provenance components are archived in a self-contained and executable package that can be always reconstituted
    • E.g.: virtual machines, Linux packing tools
  • Longevity *by upgrading*
    • Experiment can be upgraded to cope with newer software libraries
    • A detailed specification of the experiment must be provided
    • E.g.: upgrade mechanism in VisTrails [Koop et al., 2010], Web service replacement in Taverna [Belhajjame et al. 2011]
WorkFlow Upgrades

• Implementation Change

```python
def compute(self):
    fig = pylab.figure()
    ...
    pylab.scatter(x_data, y_data)
    ...
```

• Interface Change

```python
def compute(self):
    fig = pylab.figure()
    pylab.setp(fig, facecolor='w')
    ...
    pylab.scatter(x_data, y_data)
    ...
```

• Deprecation, Addition, or Replacement
Workflow Upgrades

matplotlib 1.0, csv 0.1

matplotlib 1.1, csv 1.0
Workflow Upgrades

• Solution in VisTrails [Koop et al., 2010]
  • Maintain information about library versions
  • Allow developers to define logic for upgrades
    • If they did, use those routines
    • Otherwise, try to perform automatic upgrades
  • On failure, show the incompatible workflow, notify the user, and allow the user to make changes
  • Learn from “fixes”
    • System keeps provenance of workflow evolution
    • Use provenance of manual upgrades to automatically patch other workflows
Components for Reproducibility

- Longevity
- Portability
- Document Linkage
- Experiment Sharing
- Replicability and Modifiability
- Capture
- Representation
Document Linkage

- Ability to connect the results in a document to their corresponding provenance
- Related to the idea of *reproducible papers*
- Two types of linkage
  - *By reference:* data and code are hosted externally and referenced from the document
  - *Inlined:* provenance is included *inside* the document
Galos Conjguates of Topological Phases


Microsoft Research, Station Q, University of California, Santa Barbara, CA 93106, USA
Théorîque Physik, ETH Zurich, 8093 Zurich, Switzerland

(Dated: July 6, 2011)

Galos conjugation relates unitary conformal field theories (CFTs) and topological quantum field theories (TQFTs) to their non-unitary counterparts. Here we investigate Galos conjugates of quantum double models, such as the Levin-Wen model. While these Galos conjugated Hamiltonians are typically non-Hermitian, we find that their ground state wave functions still obey a generalized version of the usual code property (local operators do not act on the ground state manifold) and hence enjoy a generalized topological protection. The key question addressed in this paper is whether such non-unitary topological phases can also appear as the ground states of Hermitian Hamiltonians. Specific attempts at constructing Hermitian Hamiltonians with these ground states lead to a loss of the code property and topological protection of the degenerate ground states. Beyond this we rigorously prove that no local change of basis (IV.5) can transform the ground states of the Galos conjugated doubled Fibonacci theory into the ground states of a topological model whose Hermitian Hamiltonian satisfies Lieb-Robinson bounds. These include all gapped local or quasi-local Hamiltonians. A similar statement holds for other non-unitary TQFTs. One consequence is that the “Gaffnian” wave function cannot be the ground state of a gapped fractional quantum Hall state.

PACS numbers: 05.30.Pp, 73.43.-f

I. INTRODUCTION

Galos conjugation, by definition, replaces a root of a polynomial by another one with identical algebraic properties. For example, $i$ and $-i$ are Galos conjugate (consider $z^2 = 1 = 0$) as are $\frac{1}{2} + \frac{\sqrt{3}}{2}i$ and $\frac{1}{2} - \frac{\sqrt{3}}{2}i$ (consider $z^2 = -1 = 0$), as well as $\sqrt{2} \cos(1/4)$ and $\sqrt{2} \cos(3/4)$ (consider $z^2 = -2 = 0$). In physics Galos conjugation can be used to convert non-unitary conformal field theories (CFTs) to unitary ones, and vice versa. One famous example is the non-unitary Yang-Lee CFT, which is Galos conjugate to the Fibonacci CFT ($C_2$). This can be explained by the non-symmetry of the Fibonacci states.

In statistical mechanics non-unitary conformal field theories have a venerable history. However, it has remained less clear if there exist physical situations in which such theories can provide a useful description of a quantum system. Galos conjugation typically destroys the Hermitian structure of a Hamiltonian. Some non-Hermitian Hamiltonians pryingly have totally realistic spectra, in the study of PT-invariant one-dimensional Hamiltonians. A notable example is the 1D quantum Hall effect in a magnetic field, which is a non-Hermitian system with a gap opening due to the breaking of time-reversal symmetry.

We reach this conclusion quite indirectly. Our main thrust is the investigation of Galos conjugation of fractional quantum Hall states.

Abelian Levin-Wen model. This model, which is also called “DFib”, is a topological quantum field theory (TQFT) whose states are string-nets on a surface labeled by either a trivial or “Fibonacci” anyon. From this starting point, we give a rigorous argument that the “Gaffnian” ground state cannot be locally conjugated to the ground state of any topological phase, which is a Hermitian model satisfying Lieb-Robinson bounds (which includes but is not limited to gapped local and quasi-local Hamiltonians).

The Lieb-Robinson bounds are a technical tool for local lattice models. In relativistically invariant field theories, the speed of light is a strict upper bound to the velocity of propagation. In lattice theories, the LR bounds provide a similar upper bound by a velocity called the LR velocity, but in contrast to the relativistic case there can be some exponentially small “leakage” outside the light cone in the lattice case. The LR bounds provide a rigorous upper bound to the velocity propagation in a lattice model.

FIG. 6. (color online) Ground-state degeneracy splitting of the non-Hermitian doubled Yang-Lee model when perturbed by a string tension ($\theta = 0$).

Reproducible result in paper and on the Web
Components for Reproducibility

- Longevity
- Portability
- Replicability and Modifiability
- Capture
- Representation
- Experiment Sharing
- Document Linkage

**REPRODUCIBILITY**
Experiment Sharing

• Provide an infrastructure to *upload, archive, and share* data related to the experiment
  • Different resources for different kinds of objects (data, source code, papers, ...)
• Related to *data preservation*
• Two types of sharing
  • *Archival:* data and experiment can be downloaded from the server, but cannot be execute at the server
    • E.g.: Github (source code), myExperiment (workflows), arXiv (papers)
  • *Hosted Execution:* experiment can be executed at the server (high portability)
    • E.g.: RunMyCode (source code), crowdLabs (workflows)
Demo

- **VisTrails**
  - Open-source scientific workflow, data visualization and provenance management system
  - It allows users to reproduce past results
    - Representation: workflow
    - Capture: workflow-based
    - Replicability and Modifiability (Parameter Exploration)
    - Portability: low
    - Longevity: by upgrade
    - Document Linkage: external (crowdLabs) and inlined (LaTeX)
    - Experiment Sharing: hosted execution (crowdLabs)
Questions?
Hands-On Assignment

1. Install VisTrails 2.1.4: http://vistrails.org/index.php/Downloads

2. Download the following packages:
   1. tabledata-backport: http://vgc.poly.edu/~fchirigati/mda-class/tabledata-backport.zip
   2. gmaps: http://vgc.poly.edu/~fchirigati/mda-class/gmaps.zip

3. Extract packages under $HOME/.vistrails/userpackages

4. Download MTA workflow: http://vgc.poly.edu/~fchirigati/mda-class/mta-analysis.vt

5. Open the workflow in VisTrails, enabling all the requested packages

6. Disable the tabledata package
Assignment

- Groups of 2 or 3 students
- You will create a public Github repository for your assignment
  - You should be familiar on how to create a Github repository and commit files to it
- After finishing the assignment, add the VisTrails workflow to the repository and use the following form to submit the Github link: http://bit.ly/1qJT1td
- The assignment must be submitted *before* the class ends, unless otherwise noted
Assignment

• Using *mta-analysis.vt* as a starting point, do the following analyses in VisTrails (be creative!):

  1. Plot in a map all the subway stations from the *Queens* line. Tag the corresponding workflow version as *Task 1*.

  2. Using the given fare card usage dataset, compare the total AirTrain full fare (*AIRTRAIN FF*) usage between *Fulton Street, Broadway*, and *Jamaica* lines; you can choose how to show this information (in standard output, in a plot, in tables, ...). Tag the corresponding workflow version as *Task 2*.

  3. Compare the full fare (*FF*) card usage in the *Fulton Street* line between the periods of 10/20/2012-10/26/2012 (http://web.mta.info/developers/data/nyct/fares/fares_121103.csv) and 10/27/2012-11/02/2012 (http://web.mta.info/developers/data/nyct/fares/fares_121110.csv); use a *bar plot*. Tag the corresponding workflow version as *Task 3*. *How are the amounts different? Why do you think the results look like this?*

• Do not forget to send the link to your public Github repository through *http://bit.ly/1qJT1td*