Transients Classification Pipeline

http://sites.google.com/site/dstarr1/tcp
http://tinyurl.com/tcp123

Dan Starr
Josh Bloom (PI)
John Brewer

UC Berkeley
Las Cumbres Observatory, Santa Barbara
Transients Classification Pipeline

- **Goal:**
  - Science classification and assessment without immediate astronomer input

- **Why:**
  - Certain science requires rapid follow-up
  - Too much data for astronomers to assess.

- **How:**
  - TCP will identify and broadcast interesting transient sources to astronomers & telescopes
  - Palomar 48” & Mosaic camera commissioned in November 2008
Telescope to Telescopes

- Palomar 48": 100 Mpix, 7.8 sq-deg detector
- ~120s cadence: 192MB: <100GB/night
- 10-100k single observation objects / night
- Post filtering: ~1000s difference-objects / night

TCP

- PTF consortium
- Palomar 60"
- PAIRITEL, MDM, Wise, ...
Transients Classification Pipeline

Data-Stream
- single observation/epoch
  (Palomar 48°)

Source Matching / Generation
- Cluster single observations with matching sources or create sources

Feature Generation
- for each source

Source Science Classification
- using source features and classification algorithms

Database Storage

Broadcast of interesting sources

Listening Telescopes & Astronomers
Why Python?

• An excellent glue
• Rapid development
• Parallelizable tasks
• Python use:
  • minor planet filter
  • “feature” generators
  • building & applying science classifiers
  • data broadcast
  • flat and structured data storage
  • test suites
Rapid development using Python

- **Useful packages**
  - Scipy, numpy, matplotlib, ...

- **Evolving constraints and architecture**
  - Ex: added surveys/telescopes
  - Ex: evolving light-curve complexities (filters...)
  - Ex: need to cache external data sources (due to throttling, etc...)

- **Allows contribution by less experienced programming community**
  - Ex: “feature” extraction algorithms
A need for parallelization

- Expanding data-stream
  - Palomar 48 inch telescope
  - SASIR, LSST?

- Immediate follow-up
  - GRB, supernova follow-up
  - Follow-up using robotic telescopes, etc...

- "Ipython1" is a branch of Ipython (merged soon) which allows parallelization

- Ipython1 lets us load modules and initialize database connections only once on each node

- Tasks can be quickly delegated to free nodes
Why Python?

- An excellent glue
- Rapid development
- Parallelizable tasks
- Python use:
  - *minor planet filter*
  - “feature” generators
  - building & applying science classifiers
  - data broadcast
  - flat and structured data storage
  - test suites
Minor planet ("asteroid") filter

- Minor planets pollute our stellar transient stream
- **PyEphem**: given orbital parameters, this estimates a MP’s position at an arbitrary time
- For every transient source, TCP checks for nearby MPs, filtering out sources with close matches
- Currently we use 360,000 MPs
- As PAN-STARRs goes on-line, many more MPs will be cataloged
- Calculations are parallelized to handle increasing catalog size
Why Python?

• An excellent glue
• Rapid development
• Parallelizable tasks

Python use:

• minor planet filter
• “feature” generators
• building & applying science classifiers
• data broadcast
• flat and structured data storage
• test suites
Generated “Features”

- “Features”: real number metrics
  - Metrics derived from time-series light-curves:
    - Ex: time-series period, amplitude, frequency components
    - Ex: period folded statistical modes
    - Ex: metrics representing “goodness-of-fit” to various models
  - Intrinsic properties:
    - Ex: distance, color differences
  - Context information:
    - Ex: nearest galaxies, galactic latitude
Science classification

- Need knowledge of most variable science
  - Build warehouse of example light curves
- Generate “priors” / algorithms characterizing each science class
- Apply “priors” / algorithms to new or updated sources, to determine their most probable science class
Building science classifiers

- Berkeley’s variable light-curve repositories
  - TCP-TUTOR: internal, evolving repository
    - ~150 science classes
    - ~14,000 sources from ~87 papers
  - (Future): http://dotastro.org
    - Any astronomer may add light-curves & science classes

- Resample light-curves to better represent
  - data-stream observing cadences
  - each instrument’s capabilities, noise

- Build science priors / classification algorithms using resampled light-curves
Applying science classifiers

- Source classification occurs at multiple points
  - Sources from real-time data-streams
  - Sources identified during ingestion of a survey
  - Sources re-evaluated by autonomous agents
- Classification algorithms implemented using a couple languages
  - R: used by Berkeley statistics collaborators, execute scripts using Python’s Rpy
  - WEKA: Java based, but Jython can be used. Can also wrap via shell.
  - Future ML algorithms (preferably Python/C++)
Data access and broadcast

• **Push:**
  * Broadcast XML via smtplib, socket, jabber
  * Broadcast sources matching astronomer predefined constraints
  * Broadcast to robotic telescopes for immediate follow-up
  * VOEvent XML packets

• **Pull:**
  * Web interface
  * Allow source retrieval using custom queries
Why Python?

- An excellent glue
- Rapid development
- Parallelizable tasks
- Python use:
  - minor planet filter
  - “feature” generators
  - building & applying science classifiers
  - data broadcast
  - flat and structured data storage
  - test suites
Data Storage

- Simple Data
  - MySQL : SDSS-II : 750M row tables
  - SQL retrieval of Palomar 48” object stream

- Structured Data
  - VOEvent XML
    - VOEvent contains different sub-elements for different instruments and follow-up groups
    - Store XML using Berkeley-DB XML, query using dbxml and Xpath / XQuery
Testing

- Self contained pipeline test framework deployable on development machines
- Currently use “unittest” package and a custom testsuite
- Future: try using the “nose” testing framework
Transients Classification Pipeline

http://sites.google.com/site/dstarr1/tcp
http://tinyurl.com/tcp123

Dan Starr : dstarr@astro.berkeley.edu
Josh Bloom (PI) : jbloom@astro.berkeley.edu

Berkeley Astronomy:
    John M. Brewer, Maxime Rischard, Dovi Poznanski, Nat Butler, Chris Klein, Rachel Kennedy

Berkeley Statistics:
    Elizabeth Purdom, Noureddine El Karoui, John Rice

Berkeley CS:
    Martin Wainwright

Los Alamos Nat. Lab. / UC Santa Cruz:
    Damian Eads

Lawrence Berkeley Lab:
    Peter Nugent