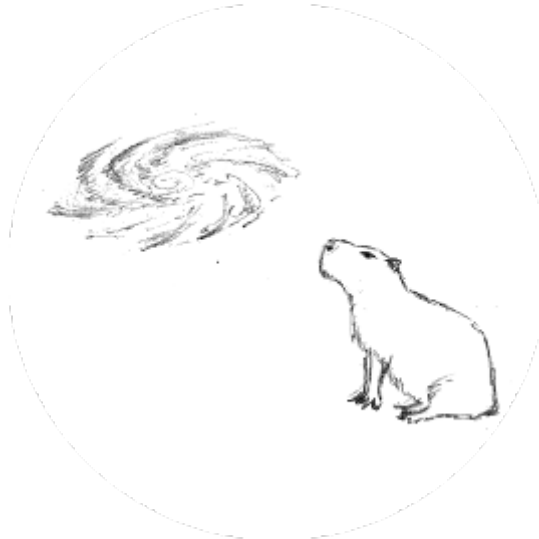


Carpyncho - Utilidades de Machine Learning



Jornada NOVA en Córdoba

21 de Agosto del 2015

About Me

- Ingeniero en Sistemas egresado de la UTN-FRC
- Doctorando en Informática en la *Universidad Nacional de Rosario*
- El título de mi plan de trabajo:

Análisis y diseño de procesos de minería de datos astrofísicos sobre catálogos fotométricos múltiple época

Directores:

Dr. Pablo Granitto (CIFASIS-CONICET)

Dr. Sebastian Gurovich (IATE-OAC-UNC)

Objetivos de Investigación

- Mejorar usos de los datos
- Integrar Aprendizaje automático al proceso de analisis de catálogos astronómicos.
- Hacer "crecer" los catalogos integrando datos de diferentes experimentos.
- Facilitar la realización de experimentos sobre los datos.
- Hacer transparente la distribucion de datos, algoritmos y resultados.

Que es Carpyngo Actualmente

- Es una herramienta en python
- Es una base de datos que alberga 2 tiles del VVV en un formato util para el análisis.
- Tiene la capacidad de consolidar los datos de los pawprints con los tiles por proximidad (matching por proximidad).
- Calcula estadísticas de las consolidaciones.
- Tiene implementado una primera version de *cone-search* (con capacidad de paginacion automatica)
- Calcula estadísticas de magnitud automáticamente.
- Calcula periodos (LS y PDM), componentes de fourier para las fuentes que se deseen.
- Soporta analisis de "estabilidad" de calculos de periodos y componentes fourier a traves de monte-carlo.

A Futuro: Busquedas

```
>>> import carpyncho as pyncho
>>> conn = pyncho.Connection()
>>> sources = conn.cone_search(ra, dec, sr)
>>> list(sources)
[source_1, source_2, ...]
>>> src = sources[0] # la primera
>>> src.clasificaciones
["rrlyrae_a"]
>>> src.ra
0.1
>>> src.magnitude.std
0.32
>>> src.mjd
[0,1,2,3...]
>>> src.magnitudes
[3,4,5,6...]
```

```
>>> sources = conn.cone_search(ra, dec, sr)
>>> sources.filter(magnitude.average > 23).exclude(sources.ra < 32)
```

A Futuro: Clasificaciones

```
>>> sources = conn.filter(magnitude in [1, 2, 3])
>>> test, training = sources.random_split(3)
>>> model = pyncho.RandomForest(training, test)
>>> model.cmatrix # confusion matrix
>>> model.score(test) # clasificacion
```

A Futuro: Réplicas y Pedidos de integración

```
>>> conn.add_replica("zodb:///db_local.zdb")
>>> conn.add_replica("mysql://user:password@localhost:3306/db_local.zdb")
>>> replica = sources.replicate()
```

esto ya no se conecta remoto pero tiene los mismos datos

```
>>> replica.cone_search(...).filter(...)
```

```
>>> sources_raras
>>> estadisticas_de_sources_raras
>>> conn.pull_request(
    sources_raras, estadisticas_de_sources_raras, "la papa",
    "integren esto por que es la verdad de la milanese")
# en un futuro
>>> sources = conn.filter(clasification.name = "la papa")
```

Mas a Futuro: Aun sin planear

- Integrar seleccion de features.
- Integrar visores multidimensionales.
- Integrar

¿Preguntas?

- Charla: <https://goo.gl/ZTJgIX>
- **Contactos:**
 - jbcabral.org
 - Juan B Cabral <jbc.develop@gmail.com>