Light Curves Classifier

Martin Vo

Masaryk University
Department of Theoretical Physics and Astrophysics
Czech Republic

EWASS 2017
Prague, 30 June
Supported by grant COST LD-15113 of Czech Ministry of Education Youth and Sports
Light Curves Classifier

Package for tasks related to classification of light curves
Light Curves Classifier

Package for tasks related to classification of light curves

Using options:

1. Python package (github.com/mavrix93/LightCurvesClassifier)
2. Web Interface (vocloud-dev.asu.cas.cz/lcc)
3. Command line UI
Motivation

Workflow of classification tasks is very similar

1. Get data
2. Get features
3. Train classifiers
4. Evaluate methods
5. Find new objects
How to describe light curves?

![Graph showing light curves with dissimilarity measure]

Template word: opiighkgfhjfm
Inspected word: ihjglfglefmre
Dissimilarity: 13.4

Shifted magnitude [mag]

Shifted time [days]
How to describe light curves?

![Graph showing light curves with dissimilarity measures](image)
Variograms

Variograms

![Variogram Chart]

Dissimilarity: 4.7

- Template variogram
- Inspected variogram
Variograms
Classification

Available descriptors
- PositionDescriptor
- VariogramShapeDescr
- VariogramSlopeDescr
- CurveDescr
- HistShapeDescr
- CurveDensityDescr
- ColorIndexDescr
- PropertyDescr

Selected items
- KurtosisDescr
- SkewnessDescr
- CurvesShapeDescr
- AbbeValueDescr

Available deciders
- NeuronDecider
- ExtraTreesDec
- CustomDecider
- QDADEC
- LDADec
- TreeDec
- AdaBoostDec
- GaussianNBDec

Selected items
- SVCDec
- RandomForestDec
- GradBoostDec
Classification

KurtosisDescr
- bins: 50:150:10
- absolute: False

SkewnessDescr
- bins: 40:160:10
- absolute: False

CurvesShapeDescr
- comp_stars: 10:100:30
- days_per_bin: 5:15:30
- alphabet_size: 0.25
- slide: average
- meth: Load template objects

GradBoostDec
- threhold: 0.8
- loss: deviance
- learning_rate: 0.05:0.3:10
- n_estimators: 100
- subsample: 1.0
- criterion: friedman_mse
- min_samples_split: 2
- min_samples_leaf: 1
Classification

Qso vs non variable stars described by kurtosis and Abbe value
Classification

Members vs non-members of a star cluster described by proper motions

Probability plot

- Searched objects
- Contamination objects

Martin Vo

Light Curves Classifier
Classification

Qso vs Be Stars described by color indexes
Classification

Qso vs periodic variable stars described by dissimilarity of light curves, histograms and variograms in symbolic space
Classification

Qso vs Be Stars described by reduced light curves by using PCA
Systematic searching

Available connectors
- Asas
- OgleII
- FileManager
- CorotBright
- Kepler
- Macho
- CorotFaint

Selected items
- Catalina
- Oglell

Query
- ra: 170.8113
- dec: 34.1737
- delta: 2
- nearest: True

Oglell
- field_num: 5:7
- starid: 15000
- target: lmc

Catalina
- Load query file

ADD >>
<< REMOVE
Getting stars

```python
queries = [{"ra": 297.8399, "dec": 46.57427, "delta": 10}, {"kic_num": 9787239}]

client = StarsProvider.getProvider("Kepler", queries)
stars = client.getStars()
```

Listing 1: Getting stars from Kepler
Filtering

```python
filt = StarsFilter([AbbeValueDescr(bins=150),
                    KurtosisDescr(bins=90)],
                   [TreeDec(threshold=0.8)])
filt.learn(searched_stars, contam_stars)
passed_stars = filt.filterStars(inspected_stars)
```

Listing 2: Filtering
Tuning parameters

```python
estim = ParamsEstimator(searched_stars, contamination_stars, descriptors, deciders, tuned_params)
estim.fit()
```

**Listing 3: Tuning parameters**
Implementing own modules

```python
class StdDesc(BaseDescriptor):
    
def getFeatures(self, star):
        
        """
        Get standart deviation of magnitudes
        """
        return np.std(star.lightCurve.mag)
```

Listing 4: Custom descriptor
Uploading methods

```python
PackageReader.appendModules("descriptors", "some_path/my_descriptors")
```

**Listing 5: Uploading methods**
Uploading methods

Listing 6: Uploading methods

In Web Interface
class MachoDb(VizierTapBase, LightCurvesDb):
    ""
    Client for MACHO database
    
    EXAMPLES:
     
     queries = [{"Field": 1, "Tile": 3441, "Seqn": 25}]
     client = StarsProvider.getProvider(obtain_method="Macho",
         obtain_params=queries)
     
     stars = client.getStars()
     ""

    TABLE = "II/247/machovar"
    LC_URL = "http://cdsarc.u-strasbg.fr/viz-bin/nph-Plot/w/Vgraph/txt?II%2f247%2f%2f{macho_name}&F=b%2
    NAME = "{Field}.{Tile}.{Seqn}"
    LC_FILE = ""

    LC_META = {
        "xlabel": "Time",
        "xlabel_unit": "MJD (JD-2400000.5)",
        "origin": "MACHO"
    }

    IDENT_MAP = {"MachoDb": ["Field", "Tile", "Seqn"]}
    MORE_MAP = collections.OrderedDict([("Class", "var_type"),
        ("Vmag", "v_mag"),
        ("Rmag", "r_mag"),
        ("rPer", "period_r"),
        ("bPer", "period_b")]}
Conclusion

- Many “ready to use” methods
- Connectors to most of the largest databases of light curves
- Possibility to easily implement new methods and connectors
- Executing jobs in cloud (for web service)
- Easy to tune parameters of methods
- Powerful library for development