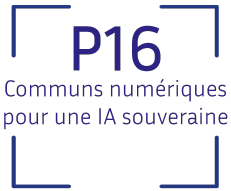
time series + ML + Python = ❤️ (& `tslearn`)



romain.tavenard@univ-rennes2.fr
guillaume.charavel@inria.fr
rtavenar
charavelg

What's `tslearn`?

Open source python library based on scikit learn

Give us a ⭐ at https://github.com/tslearn-team/tslearn

ML toolkit for multivariate variable length timeseries

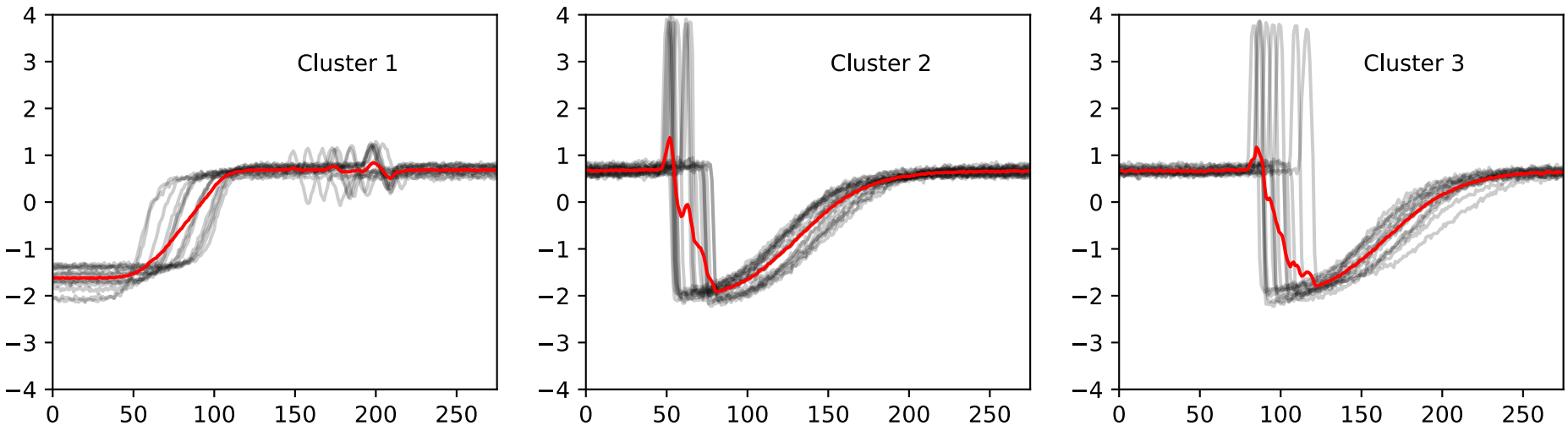
- Inputs formatting and fetching
- Preprocessing
- Metrics
- Clustering, regression and classification algorithms

In [ ]:
```
pip install tslearn[all_features]
```

Why use a dedicated library for time series?

Because if you don't...

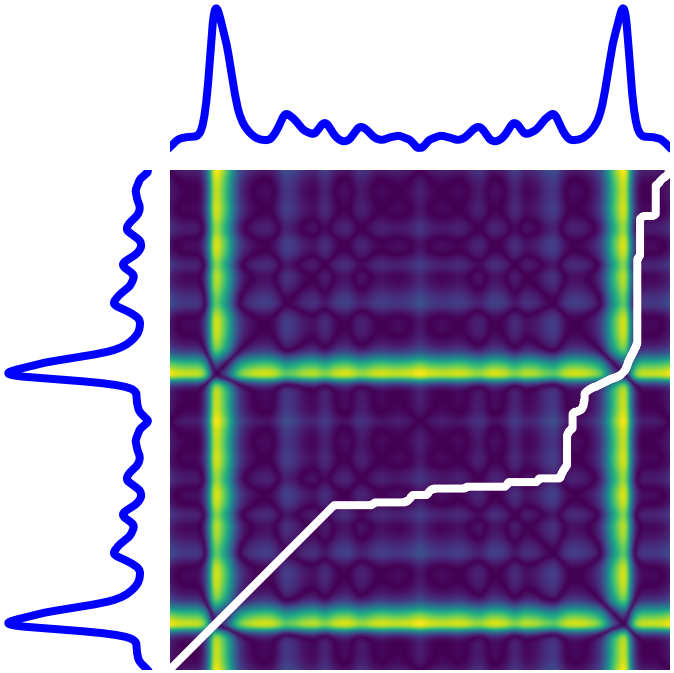
In [3]:
```
from sklearn.cluster import KMeans
from tslearn.utils import to_sklearn_dataset

X_sklearn = to_sklearn_dataset(X_train)
model = KMeans(n_clusters=3, max_iter=10, random_state=0)
model.fit(X_sklearn)
plot_clustering(model, X_sklearn)
```



Use time series metrics

eg. Dynamic Time Warping (DTW [Sakoe, 1978])

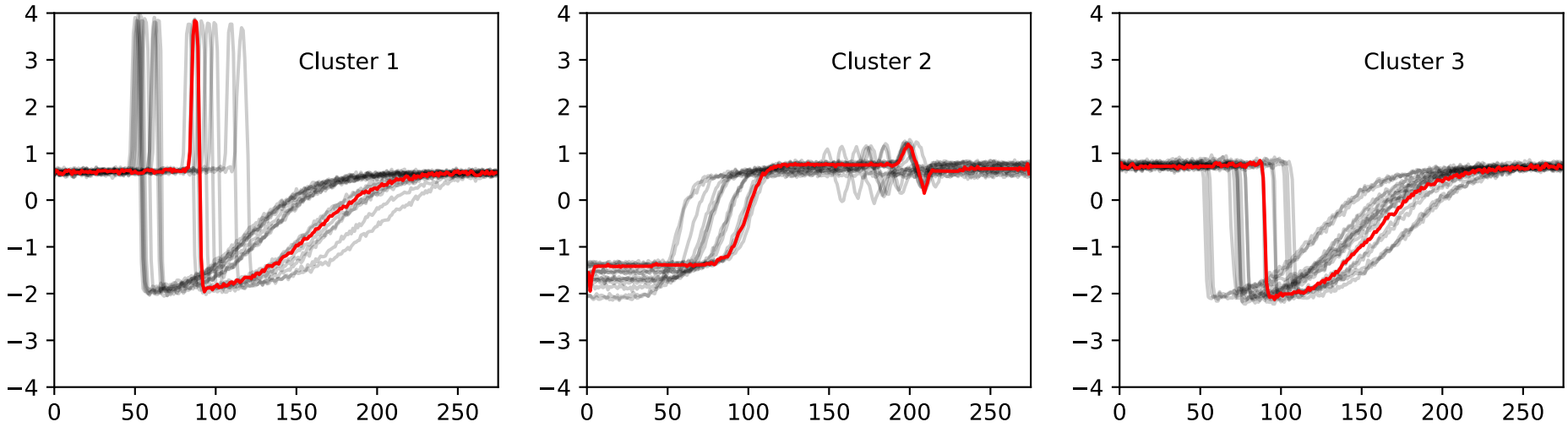
In [6]:
```
from tslearn.metrics import dtw_path

path, sim = dtw_path(s_y1, s_y2)
plot_dtw(s_y1, s_y2, path)
```



k-means + DTW

DTW Barycenter Averaging [Petitjean, 2011] to compute barycenters

In [7]:
```
from tslearn.clustering import TimeSeriesKMeans

model = TimeSeriesKMeans(n_clusters=3, metric="dtw",
                         max_iter=10, random_state=0)
model.fit(X_train)
plot_clustering(model, X_train)
```

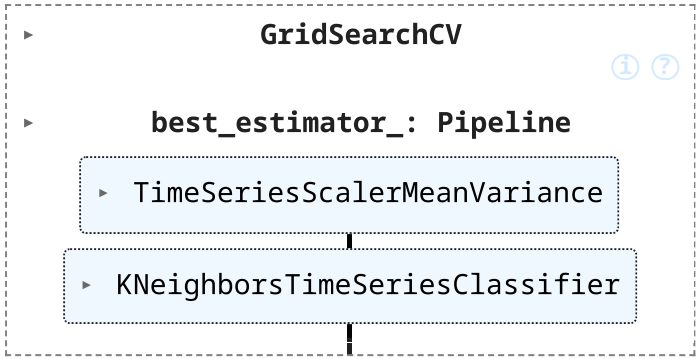


Benefit from great `scikit-learn` features

In [8]:
```
X_train, y_train, _, _ = CachedDatasets().load_dataset("Trace")

# Keep only the first 50 timeseries of both train and
# retain only a small amount of each of the timeseries
X_train, y_train = X_train[:50, 50:150], y_train[:50]
```

In [9]:
```
from sklearn.model_selection import GridSearchCV, StratifiedKFold
from sklearn.pipeline import Pipeline
from tslearn.neighbors import KNeighborsTimeSeriesClassifier
from tslearn.preprocessing import TimeSeriesScalerMeanVariance

n_splits=3
pipeline = GridSearchCV(
    Pipeline([
            ('normalize', TimeSeriesScalerMeanVariance()),
            ('knn', KNeighborsTimeSeriesClassifier())
    ]),
    {'knn__n_neighbors': [2, 5, 25],
     'knn__weights': ['uniform', 'distance']},
    cv=StratifiedKFold(n_splits=n_splits,
                       shuffle=True,
                       random_state=42)
)

pipeline.fit(X_train, y_train)
```

Out[9]:



In [11…

```
import pandas as pd
pd.DataFrame(
    pipeline.cv_results_,
    columns=['param_knn__n_neighbors',
             'param_knn__weights',
             'mean_test_score']
)
```

Out[11]:

	param_knn__n_neighbors	param_knn__weights	mean_test_score
0	2	uniform	0.640931
1	2	distance	0.740196
2	5	uniform	0.681373
3	5	distance	0.762255
4	25	uniform	0.617647
5	25	distance	0.738971

In [12…

```
pipeline.best_params_
```

Out[12]:

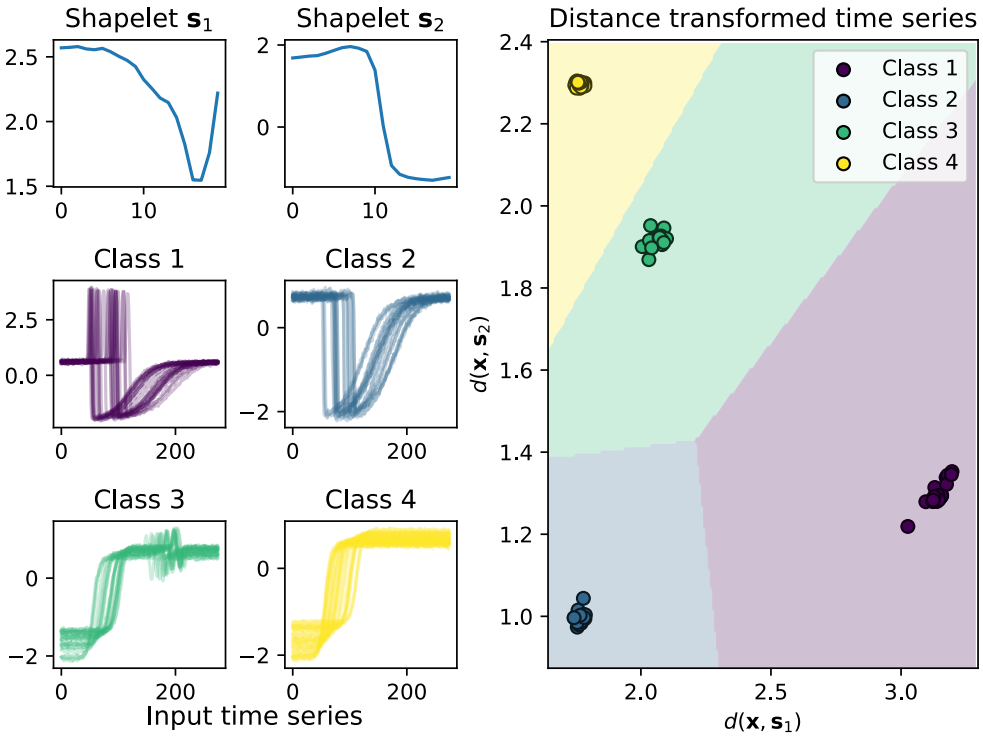
```
{'knn__n_neighbors': 5, 'knn__weights': 'distance'}
```

And now for something very similar

J. Grabocka et al. Learning Time-Series Shapelets. SIGKDD 2014.

In [17…

```
# We will extract 2 shapelets and align them with the time series
shapelet_sizes = {20: 2}

# Define the model and fit it using the training data
shp_clf = LearningShapelets(n_shapelets_per_size=shapelet_sizes,
                            weight_regularizer=0.0001,
                            optimizer=Adam(0.01),
                            max_iter=300,
                            verbose=0,
                            scale=True,
                            random_state=42)
shp_clf.fit(X_train, y_train)
```



What's next

- join the array API trend
- more estimators !!!
- Forecasting
- GPU computation optimization
- refactoring
- documentation improvements
- ...
- welcome newcomers

If you use `tslearn`, please cite us!

```
@misc{tslearn,
      title={tslearn: A machine learning toolkit dedicated to time-series
             data},
      author={Romain Tavenard and Johann Faouzi and Gilles Vandewiele and
              Felix Divo and Guillaume Androz and Chester Holtz and Marie
              Payne and Roman Yurchak and Marc Ru{\ss}wurm and Kushal
              Kolar and Eli Woods},
      year={2017},
      note={\url{https://github.com/tslearn-team/tslearn}}
}
```