

Merlion:2. Models (forecast) 应用

1. 导入包

导入 models & configs

所有模型初始化时，都需要使用模型的配置对象ModelClass(config)以及预处理模块transforms

```
from merlion.models.forecast.arma import Arima, ArimaConfig
from merlion.models.forecast.prophet import Prophet, ProphetConfig
from merlion.models.forecast.smoother import MSES, MSESConfig
```

导入Import data pre-processing transforms

```
from merlion.transform.base import Identity
from merlion.transform.resample import TemporalResample
```

2. 初始化预测模型

e.g:

(1) ARIMA假设输入数据是以一个固定的时间间隔进行采样的，所以我们将transform设置为以该间隔重新取样，我们还必须指定一个最大的预测范围。

```
config1 = ArimaConfig(max_forecast_steps=100, order=(20, 1, 5), transform=TemporalResample(granularity="1h"))
model1 = Arima(config1)
```

(2) Prophet对输入数据没有真正的假设（也不需要最大预测范围），所以我们通过使用Identity跳过数据预处理。

```
config2 = ProphetConfig(max_forecast_steps=None, transform=Identity())
model2 = Prophet(config2)
```

(3) MSES假定输入数据是以固定的时间间隔采样的，并要求我们指定一个最大的预测范围，并且这里还指定其look-back的超参数为60

```
config3 = MSESConfig(max_forecast_steps=100, max_backstep=60, transform=TemporalResample(granularity="1h"))
model3 = MSES(config3)
```

3. 模型结合

对比两个不同的组合：（1）ensemble是取每个单独模型的平均预测值；（2）selector是根据sMAPE（对称平均精度误差）选择最佳单独模型。sMAPE是一个用于评估连续预测质量的指标。

```
from merlion.evaluate.forecast import ForecastMetric
from merlion.models.ensemble.combine import Mean, ModelSelector
from merlion.models.ensemble.forecast import ForecasterEnsemble,
ForecasterEnsembleConfig
```

ForecasterEnsemble是一个预测器，它的配置需要一个组合器对象，指定你想在ensemble模型中组合的单一模型。有两种方法来指定ensemble中的实际模型。

(1) 在初始化ForecasterEnsembleConfig时提供它们各自的配置，并同时提供模型类的名称。

e.g.

```
ensemble_config = ForecasterEnsembleConfig(
    combiner=Mean(),
    model_configs=[(type(model1).__name__, config1),
                   (type(model2).__name__, config2),
                   (type(model3).__name__, config3)])
ensemble = ForecasterEnsemble(config=ensemble_config)
```

(2) 可以跳过给ForecasterEnsembleConfig的单个模型配置，而在初始化ForecasterEnsemble本身时直接指定模型。

```
selector_config = ForecasterEnsembleConfig(
    combiner=ModelSelector(metric=ForecastMetric.SMAPE))
selector = ForecasterEnsemble(
    config=selector_config, models=[model1, model2, model3])
```

4. Model Training

```
forecast1, stderr1 = model1.train(train_data)
forecast2, stderr2 = model2.train(train_data)
forecast3, stderr3 = model3.train(train_data)
```

5. Model Inference

获得每个模型的预测和标准差

```
forecast1, stderr1 = model1.forecast(time_stamps=time_stamps)
forecast2, stderr2 = model2.forecast(time_stamps=time_stamps)
forecast3, stderr3 = model3.forecast(time_stamps=time_stamps)
```

同样，也可以用于ensembles，但是没有标准差

```
forecast_e, stderr_e = ensemble.forecast(time_stamps=time_stamps)
forecast_s, stderr_s = selector.forecast(time_stamps=time_stamps,
time_series_prev=train_data)
```

(待补充) 6. Model Visualization and Quantitative Evaluation

```
from merlion.evaluate.forecast import ForecastMetric
```