M4metalearning Framework

choosing subset of M4 ts (there is 100_000 ts in the dataset)

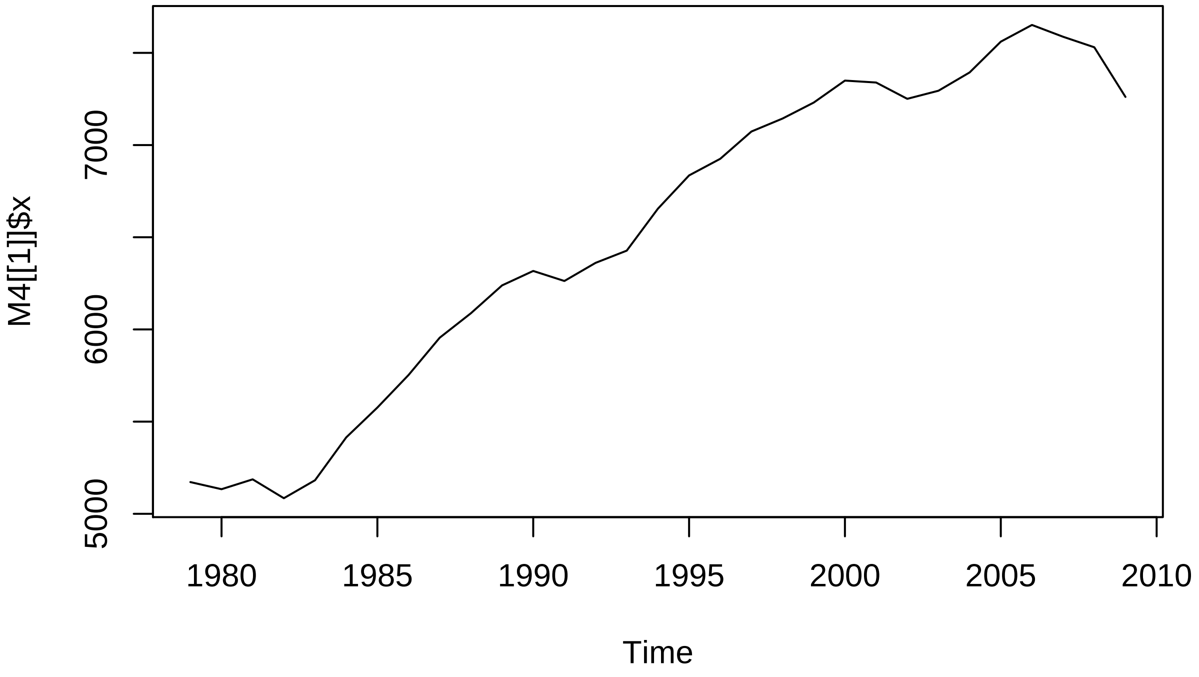
```
library(M4metalearning)
library(M4comp2018)
set.seed(31-05-2018)
indices <- sample(length(M4))

M4_train <- M4[ indices[1:15]]
M4_test <- M4[indices[16:25]]
M4_train <- temp_holdout(M4_train)
M4_test <- temp_holdout(M4_test)
```

typical time series

```
print(M4[[1]]$x)
```

```
## Time Series:
## Start = 1979
## End = 2009
## Frequency = 1
##  [1] 5172.1 5133.5 5186.9 5084.6 5182.0 5414.3 5576.2 5752.9 5955.2 6087.8
## [11] 6238.9 6317.2 6262.7 6361.0 6427.4 6654.9 6835.4 6925.5 7073.5 7144.0
## [21] 7230.6 7349.6 7339.2 7250.8 7294.6 7393.9 7560.9 7651.4 7587.3 7530.5
## [31] 7261.1
```

```
plot(M4[[1]]$x)
```



temp_holdout method

```
print(temp_holdout(M4[1])[[1]]$x)
```

```
## Time Series:
## Start = 1
## End = 25
## Frequency = 1
##  [1] 5172.1 5133.5 5186.9 5084.6 5182.0 5414.3 5576.2 5752.9 5955.2 6087.8
## [11] 6238.9 6317.2 6262.7 6361.0 6427.4 6654.9 6835.4 6925.5 7073.5 7144.0
## [21] 7230.6 7349.6 7339.2 7250.8 7294.6
```

```
print(temp_holdout(M4[1])[[1]]$xx)
```

```
## [1] 7393.9 7560.9 7651.4 7587.3 7530.5 7261.1
```

Making forecast and calculating errors for metaalgorithm training

```
M4_train <- calc_forecasts(M4_train, forec_methods(), n.cores=3)
M4_train <- calc_errors(M4_train)
```

features for metaalgorithm aka ts-classifier

```
library(tsfeatures)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
M4_train <- THA_features(M4_train)
```

```
train_data <- create_feat_classif_problem(M4_train)
```

```
head(train_data$data)
```

```
##         x_acf1  x_acf10 diff1_acf1 diff1_acf10 diff2_acf1 diff2_acf10 seas_acf1
## [1,] 0.8832748 4.232452 -0.2389881   0.1131093 -0.5941547   0.4543368 0.6966372
## [2,] 0.9904175 9.023754 -0.1865819   0.3136358 -0.6317607   1.0733833 0.8858861
## [3,] 0.8375349 3.124303 -0.3583538   0.2639291 -0.6170912   0.5811634 0.6474853
## [4,] 0.8556140 4.836155 -0.2242352   0.2329850 -0.5197771   0.5412065 0.8649517
## [5,] 0.9909209 8.848528  0.3673279   0.6335500 -0.4052513   0.2403692 0.8573676
## [6,] 0.9527165 4.764242  0.5018335   0.4929890  0.2371428   0.6361506 0.2781817
##        ARCH.LM crossing_points   entropy flat_spots   arch_acf  garch_acf
## [1,] 0.7647674               6 0.7243900         18 0.21242225 0.18230288
## [2,] 0.9815247              13 0.4111315         32 0.09079794 0.09084010
## [3,] 0.5183356              14 0.6791853         12 0.16759323 0.15602179
## [4,] 0.6985168              57 0.6268651          6 0.06490069 0.06696787
## [5,] 0.9943604               3 0.4311862         29 0.09092470 0.06106224
## [6,] 0.9332083               2 0.6257218          7 0.75737066 0.22241800
##         arch_r2   garch_r2     alpha         beta     hurst    lumpiness
## [1,] 0.22223382 0.18542345 0.7329870 0.0001000145 0.9916251 0.1065689196
## [2,] 0.08895485 0.08976024 0.7791506 0.0424917963 0.9998193 0.0001770260
## [3,] 0.28366721 0.14866902 0.2892736 0.1572207011 0.9904831 0.0161239935
## [4,] 0.06124382 0.06129501 0.6435463 0.0001000068 0.9954262 0.0292641538
## [5,] 0.11856311 0.08493449 0.9999000 0.1659826949 0.9997280 0.0002172079
## [6,] 0.34586642 0.45171193 0.9867997 0.0001000496 0.9936615 0.0248819450
##      nonlinearity   x_pacf5 diff1x_pacf5 diff2x_pacf5    seas_pacf nperiods
## [1,]  0.042662373 0.8124524   0.07269682    0.6698911  0.035003878        1
## [2,]  0.155571473 0.9969312   0.11745704    0.7045026  0.029829404        1
## [3,]  1.294375922 0.8389343   0.17977610    0.5851097 -0.115686278        1
## [4,]  0.081267255 0.8265863   0.15214459    0.6562691  0.413530471        1
## [5,]  0.004854705 0.9888423   0.21960019    0.4009188 -0.002398549        1
## [6,]  1.111609862 1.0990767   0.71935553    0.6808228  0.001903144        1
##      seasonal_period     trend        spike linearity curvature     e_acf1
## [1,]               4 0.9490610 5.666859e-06 -6.354113 -4.185679 -0.3448834
## [2,]              12 0.9980721 9.477205e-11 13.105771 -9.365355  0.4584335
## [3,]               4 0.9523771 8.566045e-07 -2.754124  6.540502 -0.4148839
## [4,]              12 0.9431769 3.300466e-08 13.167398 -6.604768  0.3190897
## [5,]              12 0.9978034 2.369111e-10 11.011219 -8.629369  0.6574255
## [6,]              12 0.9287671 7.487234e-06 -5.819744 -1.464688  0.6801245
##        e_acf10 seasonal_strength peak trough stability  hw_alpha      hw_beta
## [1,] 0.2463158        0.22267079    3      1 0.8809953 0.7437214 0.0001942303
## [2,] 0.3493179        0.52759336    2      5 1.0286614 0.9451373 0.0472874124
## [3,] 0.4648802        0.31647172    3      2 0.9444379 0.2098668 0.1981108137
## [4,] 0.3051092        0.81163073    9      5 0.7858571 0.5700414 0.0001002529
## [5,] 0.7193193        0.17663042    9      1 1.0739942 0.9643506 0.1240326909
## [6,] 0.6906730        0.08030603    3      1 0.9310720 0.9939837 0.1353067004
##          hw_gamma unitroot_kpss unitroot_pp series_length
## [1,] 0.0001089199     1.4587350   -6.697437            95
## [2,] 0.0004526029     3.4616880   -2.374209           288
## [3,] 0.0001000035     0.5308268   -7.531413            62
## [4,] 0.1877809818     4.2129081  -37.139587           334
## [5,] 0.0356037038     2.6172976   -2.376961           251
## [6,] 0.0001004050     1.1147884   -1.091533            51
```

training metalearner classifier for minimise error of ensemble on test period

```
meta_model <- train_selection_ensemble(train_data$data, train_data$errors)
```

test dataset forecasts

```
M4_test <- calc_forecasts(M4_test, forec_methods(), n.cores=1)
```

features to predict weights

```
M4_test <- THA_features(M4_test, n.cores=1)
```

```
test_data <- create_feat_classif_problem(M4_test)
```

predictions of weights for every ts and forecast method:

```
preds <- predict_selection_ensemble(meta_model, test_data$data)
```

```
preds
```

```
##            [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
##  [1,] 0.3511662 0.014302875 0.010769316 0.018170343 0.010509997 0.015961486
##  [2,] 0.6884198 0.008741332 0.006581765 0.011104970 0.006423280 0.009755007
##  [3,] 0.9299401 0.002205345 0.001660510 0.002801666 0.001620526 0.002461085
##  [4,] 0.8527997 0.004485823 0.003377589 0.005698780 0.003296259 0.005006014
##  [5,] 0.1749905 0.017243101 0.012983152 0.021905601 0.012670525 0.019242671
##  [6,] 0.9289814 0.002314442 0.001742654 0.002940262 0.001700692 0.002582833
##  [7,] 0.7183602 0.008865462 0.006675228 0.011262664 0.006514493 0.009893531
##  [8,] 0.9244711 0.002226784 0.001676652 0.002828902 0.001636279 0.002485009
##  [9,] 0.8987735 0.002879848 0.002168375 0.003658553 0.002116161 0.003213805
## [10,] 0.8924775 0.003384594 0.002548422 0.004299781 0.002487057 0.003777083
##               [,7]       [,8]        [,9]
##  [1,] 0.019322973 0.54768627 0.012110530
##  [2,] 0.011809411 0.24976302 0.007401460
##  [3,] 0.002979389 0.05446403 0.001867310
##  [4,] 0.006060280 0.11547736 0.003798236
##  [5,] 0.023295176 0.70306921 0.014600079
##  [6,] 0.003126777 0.05465123 0.001959684
##  [7,] 0.011977108 0.21894473 0.007506563
##  [8,] 0.003008352 0.05978145 0.001885462
##  [9,] 0.003890632 0.08086069 0.002438425
## [10,] 0.004572536 0.08358718 0.002865803
```

forecasting

```
M4_test <- ensemble_forecast(preds, M4_test)
```

```
print(M4_test[[1]]$xx)
```

```
## [1] 4910 4800 4620 4720 4750 4820 4870 4690
```

```
print(M4_test[[1]]$y_hat)
```

```
## [1] 4742.220 4755.686 4761.642 4764.938 4771.393 4768.596 4772.393 4785.596
```