地理空間データの交差検証、

正しくできていますか？ (short ver.)

Shinya Uryu (u_ribo)

July 1, 2019(updated) November 13, 2018 (first)

Tokyo.R#74LT @CyberAgent

データの自己相関

「近縁」なデータの類似性

ここでいう「近縁」

- 空間…
  都市部から次第に郊外に景観が変わっていく
- 時間… 一日の気温は徐々に変化していく
- 系統… 近縁な種ほど同じ生態的特性をもつ

… 性質が近いデータは値が類似しやすい

通常の交差検証

モデルの精度検証に用いるためのデータ分割の手法

k分割交差検証

- データをk個に分割
    - 分割はデータの並びに応じて行われる
- kのうち一つをテストデータ、k-1個の群を訓練データとして学習

repeated cross-validation

- k分割交差検証を繰り返す
    - テストデータの一部にfold間での重複を許す

そのデータで交差検証しても大丈夫?

空間データにk分割交差検証を適用する際の問題

ざっくりいうと

1. モデルエラー
    - 独立同分布の仮定に違反してしまう
2. 過学習しやすい
    - データの空間自己相関を考慮せずにランダムサンプリング

打開策

1. Spatial Cross-Validation
2. Target-oriented cross-validation

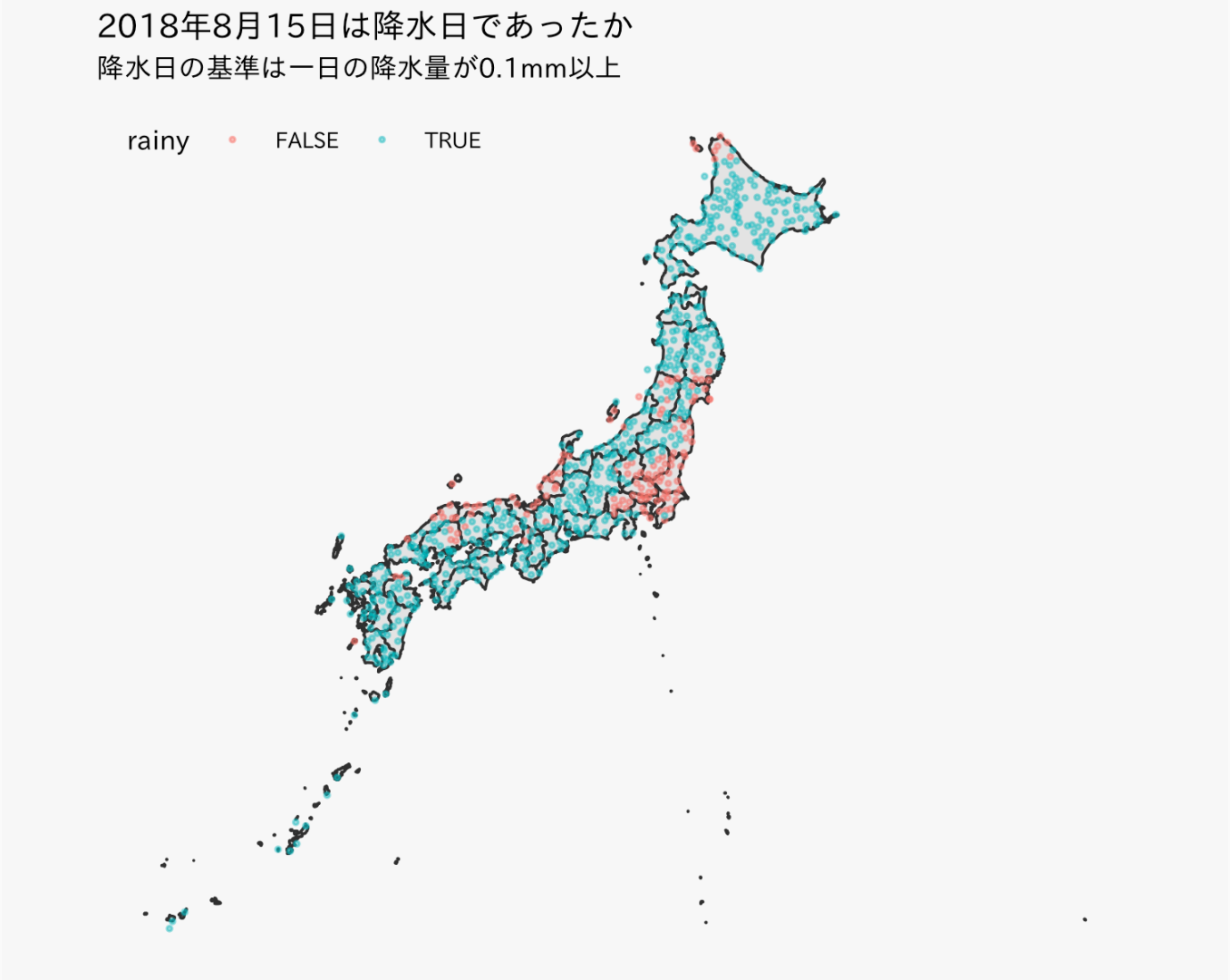
```
library(sf)
library(tidyverse)
```

気象庁のデータを用いたデモ

降水量を予測するモデルを考える

- 有線ロボット気象計
  - 降水量、気温、風向、風速、日照時間
  - 今回は降水量、気温だけを利用
- 全国668地点
- 2018年8月15日
- 降水日...
  一日の降水量が0.1mm以上あったかの判定

「雨の日」だったかを予測するモデルを考える



1/2 Spatial Cross-Validation

Step

1. 空間データの座標と他の変数を切り分ける
2. `makeClassifTask(coordinate =`)に座標データを与えたタスクを作る
3. `makeResampleDesc(method =` `"SpRepCV")`でデータ生成

データの用意

```
df_train <-
  df_weather_20180815 %>%
  select(elevation, temperature_mean, rainy) %
  st_set_geometry(NULL)

coords <-
  df_weather_20180815 %>%
  st_coordinates()
```

mlr package

```
library(mlr)

spatial_task <-
  makeClassifTask(target = "rainy",
                  data = as.data.frame(df_train),
                  #
                  coordinates = as.data.frame(coords),
                  positive = "TRUE")

learner_rf <-
  makeLearner("classif.ranger",
              predict.type = "prob")
```

Conventionally CV

データがランダムに記録されていることを想定し、RepCV

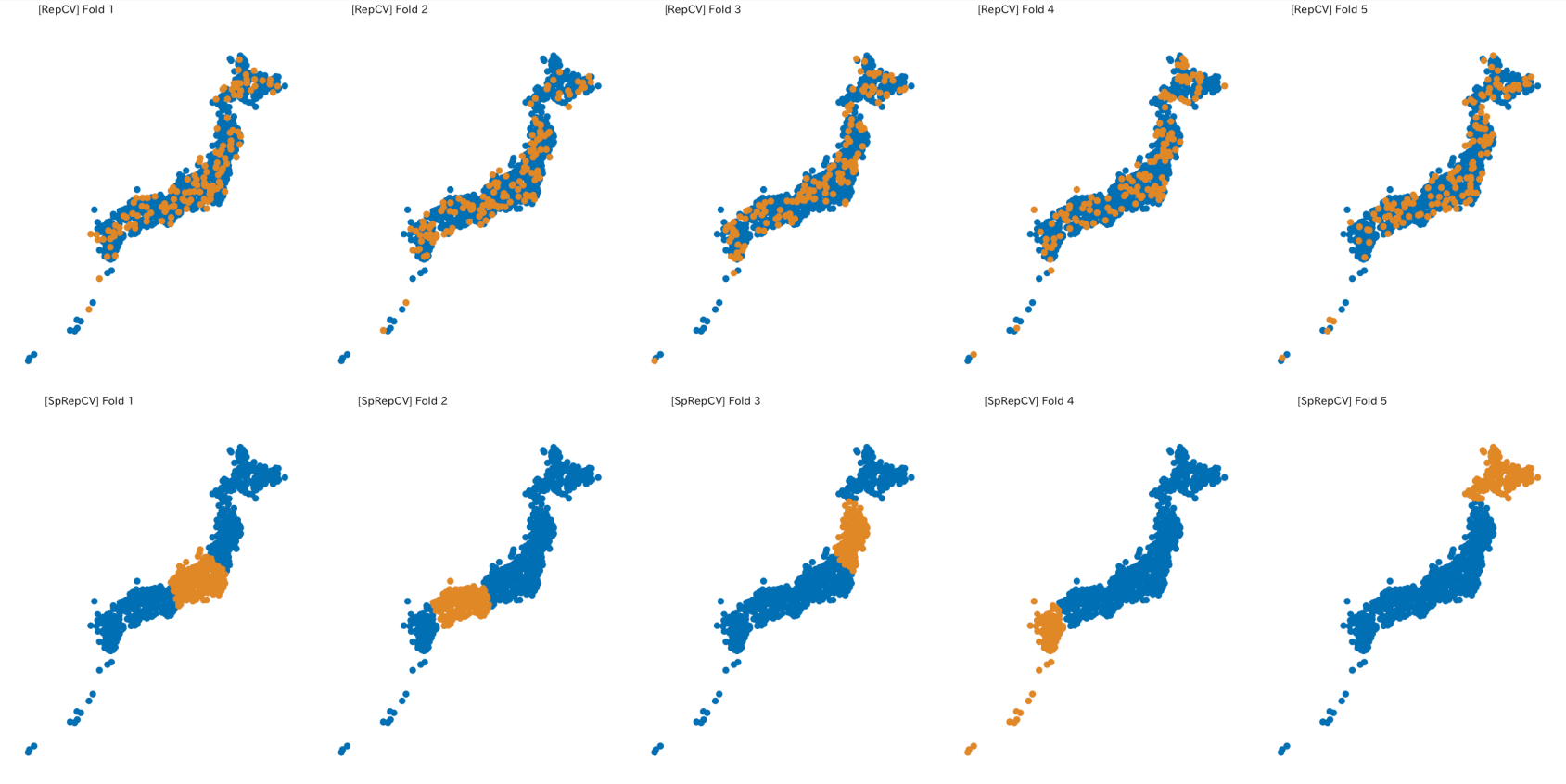
```
resampling_cv <- makeResampleDesc(method = "RepCV",
                                  folds = 5, reps = 5)
set.seed(123)
cv_out <-
  resample(learner    = learner_rf,
           task       = spatial_task,
           resampling = resampling_cv,
           measures   = list(auc))

mean(cv_out$measures.test$auc, na.rm = TRUE)
# [1] 0.8544815
```

Spatial CV

```
resampling_sp <-
  makeResampleDesc("SpRepCV", folds = 5, reps = 5)
set.seed(123)
sp_cv_out <-
  resample(learner    = learner_rf,
           task       = spatial_task,
           resampling = resampling_sp,
           measures   = list(auc))
mean(sp_cv_out$measures.test$auc, na.rm = TRUE)
# [1] 0.7891348
```

プロット

上段: Rep k-fold CV 下段: Spatial CV



Repeat k-fold CV vs Spatial CV

- Repeat k-fold
  CVではテストデータが地理的にランダムに散ってしまう
  - データ漏洩に繋がってしまう恐れも
- Spatial
  CVではデータの空間配置を考慮したspatial
  partitioningが行われる
  - 地理的に近いデータをtestデータとして使う

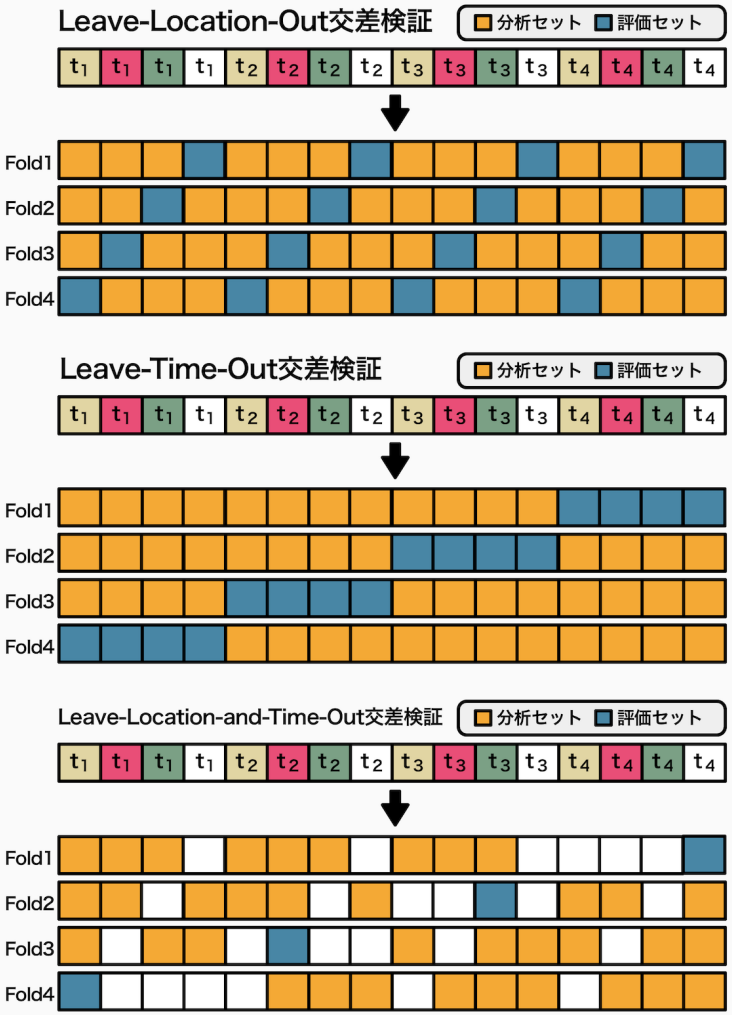
2.2 Target-oriented cross-validation

Target-oriented cross-validation

ざっくりいうと

- 空間 + 時間データの自己相関にも対応可能なCV
- サンプリングのデータの配置戦略を考慮する
  - LLO-CV… 特定の地点 (Location)のみをテストに
  - LTO-CV… 特定の時点 (Time) のみをテストに
  - LLTO-CV… 特定の地点および時点のみをテストに
    - 訓練データからは同一時点・地点のデータも除外

Target-oriented cross-validation



時空間データ分割の方法

1. LLO
同一地点を評価セットに含めない

2. LTO
同一時点を評価セットに含めない

3. LLTO
同一時点・場所を評価セットに含めず
分析セットからも除外

github.com/uribo/dpp-cookbook

Mayer et al. (2018) を元に作成

CAST package

```
library(CAST)
library(caret)
```

Step

1. `CreateSpacetimeFolds()`で
   割り当てデータのインデックスを操作
2. `trainControl(index = )`に
   生成したインデックスを指定

降水量を予測するモデルを作成

データの用意

```
df_train <-
  df_weather_20180815 %>%
  select(station_no,
                  elevation,
                  temperature_mean,
                  precipitation_sum) %>%
  st_set_geometry(NULL)
```

caretでtrain()

```
set.seed(123)
model <-
  train(df_train[, c("elevation", "temperature_mean")]
        df_train$precipitation_sum,
      method     = "rf",
      tuneLength = 1,
      importance = TRUE,
      trControl  = trainControl(method = "cv",
                                number = 5))
```

caret で cv

```
model$results
```

mtry	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
1	24.9597	0.2386014	15.37561	3.43792	0.0580465	1.339328

Target-oriented CV

考慮すべき変数（空間、時間）を
CreateSpacetimeFolds(spacevar =) に指定

```
indices <-
  CreateSpacetimeFolds(
    df_train,
    spacevar = "station_no",
    k        = 10,
    seed     = 123)
```

Target-oriented CV

```
set.seed(123)
model_LLO <-
  train(
    df_train[, c("elevation", "temperature_mean")],
    df_train$precipitation_sum,
    method     = "rf",
    tuneLength = 1,
    importance = TRUE,
    trControl  = trainControl(method = "cv",
                              index = indices$index))
```

Target-oriented CV

LLOの方がConventional CVよりもRMSEが低くなった

```
mean(model$results$RMSE)
```

```
## [1] 24.9597
```

```
mean(model_LLO$results$RMSE)
```

```
## [1] 24.61368
```

References

- Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Arroita, G. G., et al. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography, 40(8), 913–929.
- Meyer, H., Reudenbach, C., Hengl, T., Katurji, M., & Nauss, T. (2018). Improving performance of spatio-temporal machine learning models using forward feature selection and target-oriented validation. Environmental Modelling & Software, 101, 1–9.
- The importance of spatial cross-validation in predictive modeling
- Visualization of spatial cross-validation partitioning