Arrest Classification

18/01/2021

Imports

```r
library(tidyverse)
library(data.table)
library(lubridate)
library(caret)
library(e1071)

devtools::load_all("../chigcrim/")

set.seed(1)
```

Introduction

Predicting if a suspect is arrested is a binary classification problem. To assess performance, three metrics will be used, the overall accuracy, the sensitivity and the specificity. These are defined as follows:

- Overall accuracy: The proportion of correct predictions.
- Specificity: Proportion of negative observations correctly predicted.
- Sensitivity: Proportion of positive observations correctly predicted.

Here, two classifiers are considered, logistic regression and a support vector machine classifier. Note that as logistic regression is a probabilistic classifier, the results are rounded to yield predictions that can be assessed with the metrics above to facilitate the comparison with the support vector machine results.

Feature selection

Due to computational limitations, only the 2019 data will be considered. Below, the feature selection choices are outlined:

- Encode the date as the day in the year (1-365).
- Encode time of day as a float 0-24 (this looks OK from a plot of proportion arrested over time).
- Drop `year` as these are all 2019.
- We will drop the `ids`, as it contains unique values.
- We drop `case_number`, as it contains almost all unique values.
- Drop `primary_type`, `description` and `iucr` code. Keep `fbi_code` as an indicator of crime type.
- Assume `updated_on` is not informative (see EDA).
- `latitude` and `longitude` are dropped. `x_coordinates` and `y_coordinates` are kept. Note that these coordinates likely not particularly useful for linear classifiers, but should be useful for non-linear methods.

- Community areas will be kept, but other areas (`district`, `beat`, `ward` and `block`) are dropped.
- NAs will be dropped (see EDA).
- Particularly rare factors will be grouped into a variable other (see `?otherise`).

```
df <- load_data(year = 2019, strings_as_factors = FALSE)

df$fbi_code <- otherise(df$fbi_code, 500)
```

```
## [1] "7 out of 26 categories were converted to OTHER corresponding to 0.512086233171979 % of observati
```

```
df$location_description <- otherise(df$location_description, 1000)
```

```
## [1] "126 out of 155 categories were converted to OTHER corresponding to 10.8171498984657 % of observa
```

```
remove_features <- c("id", "year", "case_number", "primary_type", "description",
                     "iucr", "updated_on", "latitude", "longitude",
                     "date", "district", "beat", "ward", "block")

df <- df %>%
  mutate(day = yday(df$date),
         time = hour(df$date) + minute(df$date)/60) %>%
  select(-all_of(remove_features)) %>%
  filter(complete.cases(df))

# Convert to factors
df <- df %>%
  mutate(location_description = as.factor(location_description),
         fbi_code = as.factor(fbi_code),
         community_area = as.factor(community_area))

head(df)
```

```
## # A tibble: 6 x 9
##   location_descri~ arrest domestic community_area fbi_code x_coordinate
##   <fct>            <lgl>  <lgl>    <fct>          <fct>           <int>
## 1 RESTAURANT       FALSE  FALSE    14             14            1153943
## 2 RESIDENCE        FALSE  FALSE    44             14            1182085
## 3 HOTEL/MOTEL      FALSE  FALSE    8              04B           1175159
## 4 ALLEY            TRUE   FALSE    23             15            1151958
## 5 APARTMENT        FALSE  TRUE     29             08B           1152589
## 6 RESIDENCE        FALSE  FALSE    30             14            1155950
## # ... with 3 more variables: y_coordinate <int>, day <dbl>, time <dbl>
```

Logistic Regression

We will first consider a logistic regression classifier. Due to the size of the data set (258150 × 9), running several folds would be time consuming, so only a single fold is used. The factor variables are one-hot-encoded internally.

```
X <- df %>% select(-arrest)
y <- df$arrest

n_test <- round(0.2*nrow(X))
test_idxs <- sample(1:nrow(X), n_test)

X_train <- X[-test_idxs, ]
```

```r
X_test <- X[test_idxs, ]
y_train <- y[-test_idxs]
y_test <- y[test_idxs]

lr <- LogisticRegression$new(solver = "BFGS")
lr$fit(X_train, y_train, control = list(maxit=1000, reltol = 1e-4))

y_hat <- lr$predict(X_test)
y_hat <-  as.logical(round(y_hat))
lr_results <- confusionMatrix(as.factor(y_hat), as.factor(y_test))
```

Support Vector Machine

As the support vector machine allows use of a kernel function, it should be able to capture non-linear relationships in space (given an appropriate kernel). Here, a radial basis function kernel is used. Unfortunately, support vector machines do not scale well to large data sets, so here we will only use 20000 rows for training.

```r
# Make train and test data (normally want train > test
# but SVM won't compute with large number of rows)
train_idxs <- sample(1:nrow(df), 20000)
train <- df[train_idxs, ]
test <- df[-train_idxs, ]

# Create svm using e1071 package
svm_ <- svm(arrest ~ ., data = train, type = "C-classification", kernel = "radial")

y_hat <- predict(svm_, newdata = test)
svm_results <- confusionMatrix(y_hat, as.factor(test$arrest))
```

Comparison of models

The results for the models are shown below:

Logistic regression:

- Overall accuracy: 0.8620376
- Sensitivity: 0.9785241
- Specificity: 0.4399069

Support Vector Machine

- Overall accuracy: 0.8607054
- Sensitivity: 0.9871373
- Specificity: 0.4032077

Surprisingly, despite the ability of kernel ridge regression to utilise non-linear relationships (in the original feature space), it did not perform better than logistic regression (and was much more computationally costly to train). The support vector machine did have higher sensitivity. This reflects the fact that the logistic regression classifier gave more balanced predictions between the classes, inevitably leading to more false negatives.

Perhaps these results suggest that much of the spatial relationship in the data could be captured using the `community_area` and `location_description` factor variables, rather than the specific x and y coordinates of the crime. Hence, the logistic regression classifier could perform well, despite the data being spatial, and likely infringing on the independent and identically distributed assumption of logistic regression.