SVM by category, CV for SVM, and Outlier Detection

Sameerah Helal

12/11/2020

Background

Customer acquisition and retention are one of most important goals of any business. This is particularly true of banks, which rely on long-term connections, like loans or accounts, in order to stay in business. We are given a data set related to a marketing campaign performed by a Portuguese bank; including client details and social/economic context variables; with the variable of interest being campaign success, defined by whether or not a client subscribed to a long-term deposit with the bank.
Given this data, our primary question of interest is to identify the best model to predict whether or not the a client will subscribe to a long term deposit. We will attempt to find the most important individual features as well as the most useful combinations and linear combinations. We will then test different prediction models, tuning to get the best parameters, then comparing the models to identify the one best suited to predict our variable of interest.

SVM by Variable Category

In the context of the business problem of predicting the success of telemarketing in getting a client to sign on to a long-term deposit based on a limited number of variables, we may encounter the case where the bank is not able to collect or access the full range of variables. In addition to feature selection using other methods, we may also attempt to predict y based on different categories of information the bank may have. We consider life information (age, job, marital status, and education level), bank information (whether or not the client has a housing loan, personal loan, or credit in default), last contact information (number of times contacted during this campaign, number of days between the most recent two campaigns, number of times previously contacted, and outcome of the previous campaign for a particular client), and social and economic context at the time (employment variation rate, consumer price index, euribor 3 month rate, and number of employees). We first load and encode the data.

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Warning: package 'factoextra' was built under R version 4.0.3

## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

## Warning: package 'e1071' was built under R version 4.0.3
```

We then train four SVM models by separating the variables by category.

We test our models using the test set containing 30% of the data, which we feed into the model and use the results to compute accuracy.

```
## [1] 0.888727
```

```
## [1] 0.8884843
```

```
## [1] 0.896415
```

```
## [1] 0.8901837
```

Our results are that life information and bank information have approximately 88.4% accuracy, while last contact information has 89.95% and socio-economic context has 88.9%. So if the bank were able to use only one of these categories, it would have the best, if only marginally better, accuracy by using last contact information as their predictor. This is consistent with our previous findings that duration of phone call had the highest relative importance as a predictor, since the highest accuracy among the categories belongs to that of last contact information.

Cross Validation for SVM

To get the best possible parameters for SVM, we tune the model. To accommodate the function requirements, we use a different encoding in one aspect: we encode the output y as factor levels instead of numeric binary or continuous. We also separate the train and test data.

Our data is appropriately formatted for our needs, and we may perform cross validation. In this case, we use k-fold cross validation with $k = 10$.

```
##
## Error estimation of 'svm' using 10-fold cross validation: 0.09090907
```

```
## [1] 0.9092822
```

The result of performing 10-fold cross validation for SVM on this data set is a tuned model that has an accuracy of 90.92% when tested on the test set. This is not an increase from our model that did not include cross validation. However, this may be accounted for by a possible decrease in over fitting, meaning that, while the accuracy for this particular test set is not better, the cross-validated model will generalize better.

Outlier Detection (Literature Review)

Given a data set, outliers are data points that deviate from the rest of the samples, often to the point of skewing any models trained using them. There are many methods for computing outliers, including, but not limited to z-score, local outlier factor, one-class SVM, and isolation forest. For our data, we use isolation forest because it is suited for data with many, sometimes irrelevant attributes, and is robust despite needing few parameters.
https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf?q=isolation-forest
Isolation forest is a nonparametric, unsupervised outlier detection algorithm that is principled upon outliers being few in number, and far from the rest of the data. As a method that uses binary trees, the idea of isolation forest is that anomalies would be more easily "isolated" into leaf nodes than the other data points. The measure of ease of isolation in this case is height, or path-length from the root. The algorithm randomly selects a feature and a split value within the min-max range, and as it compares observations to the split value during prediction, "path lengths" is recorded. Those points that are isolated faster, outliers, will have shorter path lengths. For each observation, an outlier score in the unit range $[0, 1]$ is computed, correpsonding the the sample's "outlierness", with 1 being more like an outlier and 0 being less like an outlier.
Using isolation forest, we compute the number of outliers in the data set. Since the output is not a logical value, we use a threshold of 0.5, which corresponds to average outlierness, to identify outliers. If a sample has greater than 0.5, or greater than average outlierness, then we mark it as an outlier.

```
## Warning: package 'isotree' was built under R version 4.0.3
```

```
## [1] 637
```

```
## [1] 0.01546567
```

We find that, in our data set with over 40,000 entries, we have 615 outliers, or that 1.49% of the data has greater than average likelihood of being an outlier.