title: Deduplication of contact information

author: Ignacio Arroyo Fernandez

date: June 28, 2018

Data exploration

The data I downloaded has a main file whose content is in CSV format. Each row containing results and labeling provided by a baseline system based on string distance metrics and manual annotation of each sample pair of clients. The first field of the file is a kind of index of the associated sample. The next fields are labeled according to the name of the distance metric and the client information fields. These fields indicating distance metrics for each pair of samples constitute a feature vector representing each sample pair. The last field is the manual annotation of whether a sample pair corresponds to the same client (1) or not (-1).

The dataset is imbalanced. That is, there are 18% of samples labeled as -1 (no match), while the 82% remaining is labeled as 1 (matching). This condition implies additional difficulties in the case of proposing a Machine Learning approach. In addition, there are lots of missing values in the feature vectors.

Modeling approach

In this test I have a Record Linkage (RL) problem which is currently well studied in the state of the art of Machine Learning. The first to take into account in this scenario is that the feature vectors are scores given by string similarity metrics. Although this approach of representing record similarities is quite comprehensive, it is very effective in this application.

For modeling this problem I decided to use supervised learning. This is because it is possible taking advantage of the manually labeled data provided. Also, by simple inspection, I can saw that in fact several string similarity scores correlate with the manual labels. This is a good indicator of that the distributions of the features are deviated accordingly with the correct class each sample belongs to. Likely the samples are linearly separable regardless their 26 dimensions. Furthermore, it can be more time-consuming trying to determine the importance of features in advance (an unsupervised approach can also be effective in the case of doing feature engineering beforehand). Those things made me to think that a supervised learning algorithm can determine such importance by itself.

The supervised learning algorithm I used is a Support Vector Machine (SVM). Its performance in imbalanced feature scenarios is good. Class imbalance scenarios are also well handled by SVMs as this algorithm is mainly based on geometrical intuitions.

Implementation

The implementation of the modeling of this RL problem is in Python 3.6. Also a GitHub repo was created.

Dependencies

A third party Machine Learning library called scikit-learn (v >=0.18) was used, which must be installed in any Linux system to reproduce the results reported in next section.

Performance analysis

In order to verify the performance of my modeling approach I used the evaluation metric suggested in the task statement. That is, a mixture of specificity and sensibility i.e. metric = (sensibility + 2 * specificity)/3.

The modeling approach adopted in this test reached an evaluation score of: **0.917**. As I used a linear SVM, I think the classes of this dataset are linearly separable, so the classifier reached good generalization. This also can confirm the hypothesis of that several string similarity scores correlate with the manual labels, so the means of these scores are statistically independent.

It is needed to say that the samples were randomly splitted, which let us to be confident with the result obtained. An additional test for gaining confidence on the modeling approach is to perform grid search on the model parameters, which can be combined with cross-validation.

Conclusions

The test performed this time was a RL problem and it was correctly modeled by a binary classifier. The results showed that missing values and class imbalance were not meaningful perturbations to the SVM algorithm used. Thus, relatively high performance was reached (0.917).

Potential improvements

Additional time is needed to perform a formal Exploratory Data Analysis, which can be helpful in determining the data distribution, as well as in assessing the actual impact the missing values have in modeling the learning problem.

The first potential improvement can be adding Fizzy String Matching measures to the feature vectors. Feature Selection methods are also recommended for machine learning approaches. In this case, we have a highly imbalanced dataset, and in addition we face highly imbalanced features, so Information-Theoretic feature selection methods are recommended to weigh the importance of features. Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used, simple and effective method of this kind.