Feature Generation Using Machine Learning from Large Sparse Financial Data

*Emergent Research Forum (ERF)*

# 1. Introduction

Financial institutions have been using predictive analysis for a long time. The recent availability of computation resources and tremendous customer data has made it possible to apply novel methods and techniques that have not been tried earlier. While many financial operations require human-directed decisions based on pre-defined rules, for instance, loan eligibility evaluation, the introduction of automated analysis techniques is altering the landscape of classic financial analytics. Especially, machine learning technologies, which learn the features and the relations among them in a high-dimensional space without necessarily being specifically directed, have proved useful in finance domain (López de Prado 2018).

Coupled with digitization, the modeling of customer features has become a core component in modern financial analytics. For example, in credit card businesses, marketing offers tailored to each customer’s needs has reached a more granular level ever before. The tractable customer savings and spending have implications in interest rates and personal wealth growth estimation. An analytical limitation is that, however, it is not feasible to adopt the known machine learning methodologies directly to the financial data due to its unique characteristics. First, the usual set of predictors *X*, such as earnings or income, has non-normality. With a highly skewed distribution, the empirical econometrics studies have resolved such an issue with complex mathematics and specific parametric assumptions. Granted, the extrapolation and generalization of such data may not be reliable across other analytical tasks.

A second caveat in financial data is the variables with low levels. Most of the business problems in finance related to customer are to predict the target *y*,but the dependent variable has low levels in classification. For binary labels, logistic regression has widely been employed, but it cannot address class imbalance, i.e., when true classes are scarce, and problems such as perfect separation may occur (King and Zeng 2001). Imbalanced classes are quintessential in financial analysis, for example, economic events in market, insurance claims from clients, frauds among normal transactions, and lending defaults in banking, to name a few. A highly imbalanced class may hinder adequate fitting in parameter space, so adjustments should be made taking the data properties into consideration.

Another characteristic of the financial data set, prevalent both in the predictor *X* and the target *y*, is that whereas rich in volume, most of the records are zeros, if vectorized. Unlike behavioral data tracked online, such as click streams, the financial records are stacked comparably slow, due to the low frequency of the transactions. The adequate transformation of features can be a potential solution, but the heterogeneity of the observations, e.g., continues versus categorical, count versus nominal, stock versus flow, or sequential versus stable, constitutes a significant barrier in the application of machine learning. For example, supervised learning algorithms would require stationary and i.i.d. features but we cannot generate a large number of new features from a known sample of observations. Hence, it is critical to resolve the sparsity; otherwise (local) optimization may not occur properly.

In this study, we try to address the above difficulties in adopting machine learning methodologies. We test feature processing using multiple machine learning approaches in combination with established methods. We evaluate separate feature selection results as part of a prediction pipeline, and show how they differ across models. The empirical implications of the feature transformation and selection on the prediction outcomes are discussed.

# 2. Related Work

The problem of feature selection is defined as follows: given a set of candidate features, select a subset that performs the best under some classification system. This procedure can reduce not only the cost of extraction by reducing the number of features that need to be collected, but in some cases it can also provide a better classification accuracy (Jain and Zongker 1997).

Feature selection composes a well-established line of research in information retrieval and data mining. However, there is still yet scare research on learning problems in financial applications. Ban and Rudin (2018) is a recent example that attempted using machine learning in feature extraction from large data in the areas of management science.

For feature selection, linear models such as such as support vector machines (SVM), unsupervised mechanism such as clustering, and matrix factorization have proved well in the literature. Another line of the literature has focused on the induction algorithms that scale well to find, among a large amount of correlated variables, comparably irrelevant ones to be eliminated to better predict. In the statistics literature, penalized regression such as Lasso (Tibshirani 1996), Elastic Net (Zou and Hastie 2005) which combines *L*1-regularization and *L*2-constraints have been proposed.

Bootstrap aggregation (bagging) is an effective way of reducing the variance of forecasts. The classifier first generates *N* training datasets by random sampling with replacement. Then fit *N* estimator, one on each training set. These estimators are fit independently from each other. Hence, the models can be fit in parallel. Third, the ensemble forecast is the simple average of the individual forecasts from the *N* models. In the case of categorical variables, the probability that an observation belongs to a class is given by the proportion of estimators that classify that observation as a member of the class (majority voting).

Kearns and Valiant (1989) were among the first to find to ask whether one could combine weak estimators to achieve one with high accuracy. Schapire (1990) demonstrated that using the procedure we call boosting. The main distinction from bagging is that it uses sequential fitting.

In this paper we use unsupervised training with principal component analysis (PCA) to transform the feature space. Then, we use learning models to extract features from large sparse data, adopted from statistics and engineering. Recently, several algorithms have been developed, which are able to learn features from datasets in different domains (Hinton 2002, Hinton et al. 2006). In applications to computer vision and engineering, the state-of-the-art feature learning showed similar or better performance compared to non-learning algorithms.

# Methods, Models, Experiments

## Data

The experiment uses customer profile and consumption records from one of the major credit card companies in Korea. Transactions from August 2018 through October 2018 were aggregated to the card holder level. The data set was limited to 100,000 randomly chosen individuals who have at least one credit card usage in selected businesses during the window. First, about 644 measurements were chosen based on heuristics. The initial list of variables include customer’s age, sex, occupation codes, address area codes, internally managed customer grades, etc., most of which are multi-class variables. A few more binary variables were introduced, for instance, indicating whether the customer holds a specific kind of card or not. Additionally, count variables were considered, e.g., the occurrence of specific types of transactions. The remaining involves numerical metrics derived from credit line and amount used, divided into whether it was spent on weekdays or weekends, per merchant category. The encompassing categories are thus *N* (number of merchant business types) × *M* (number of months) × 2 (weekends or weekdays). Finally, behavioral proxies to capture were added.

Then, categorical variables were transformed through one-hot encoding and continuous variables were scaled using standard normalization (i.e., *z*-score, see below). Missing values were either encoded as a separate class or substituted by column mean.

After preprocessing, a total of 774 features were produced. Cross-correlation was checked and no pair showed strong collinearity. The total observation is therefore 77.4 million, which majorly contain zeros.

Our targets constitute only 2% of the population, a good example of imbalanced class problem. Among the 100,000 customers, only about 2,200 individuals are the targets. To learn the features from the target, the data was split into training, test, and validation samples, with a ratio of 5:2.5:2.5.

## Resolving Class Imbalance

Additionally, sample weights were considered in the implementation of the validation models. Class weights are the subsampling weights that correct for underrepresented labels. This is particularly critical in classification problems where the most importance classes have rare occurrence such as our case. We tried utilizing random undersampling (undersampling of the majority), random oversampling (oversampling of the minority), SMOTE, TOMEK, and SMOTE+TOMEK algorithms, and random undersampling have shown the best performance. Therefore, the validation models used additional balanced models.

## Feature Space Transformation

In this experiment we evaluate how the performance of classification models is affected by using different feature transformation and selection schemes. We choose logistic regression (LR) penalized using the L1-norm, linear support vector machines (LSVM) with a L1-penalty, random forests (RF), and extra tress (XT) models for selecting features. A penalized logistic regression can be effectively used as a feature selector, shrinking irrelevant features’ coefficients to zero. A linear SVM creates a hyperplane that uses support vectors to maximize the distance between the two classes. Thus, when finding the vector coordinates orthogonal to the hyperplane, the absolute size of the coefficients in relation to each other can be used to determine feature importance. Analogous to penalized logistic regression, the hyperparameter C controls for the sparsity of the regularized coefficients.

Both random forests and extra trees produce randomized decision trees to circumvent overfitting. Every node in the trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure based on the (locally) optimal condition is called impurity. At each node of each decision tree, the selected feature splits the subset in a way that impurity is decreased. Therefore, we can derive how much the overall impurity decreases for each tree. Given that we have a forest of trees, we can average those values across all estimators and rank the features accordingly.

The parameters for the feature training models were explored and fixed after being compared using a 5-fold cross-validation. Table 1 shows a comparison of features importance resulting from the four selection algorithms. The feature importance score was calculated using coefficients in linear models, and mean decrease impurity (MDI) in tree-based models. Note that the linear models used the extended one-hot encoded features, so each label in the multi-class category shows different importance scores. To make the measures commensurate, the same score was assigned per each label in XT and RF as well.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | LR | LSVM | XT | RF |
| LR | 1.000 | 0.986 | 0.281 | 0.225 |
| LSVM |  | 1.000 | 0.156 | 0.116 |
| XT |  |  | 1.000 | 0.873 |
| RF |  |  |  | 1.000 |
| Table 1. Cross-correlations of Feature Importance | | | | |

While the linear model group and tree-based group respectively seem to produce similar results therein, they show a striking difference against each other. Whether a particular feature was selected or not is another important issue. Table 2 confirms that the two groups differentiate survived features, as a correlation measure of the binary indicators for whether each feature was selected or not.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | LR | LSVM | XT | RF |
| LR | 1.000 | 0.663 | 0.055 | 0.063 |
| LSVM |  | 1.000 | 0.170 | 0.148 |
| XT |  |  | 1.000 | 0.796 |
| RF |  |  |  | 1.000 |
| Table 2. Cross-correlations of Feature Selection | | | | |

In addition to feature subset extraction, dimensionality reduction was adopted using principal component analysis, compressing the features to 2-dimensional vectors. The validation models used such an input and were compared as well.

## Evaluation

Table 3 summarizes the predictive performance of a selected feature set fitted on different models. The bold numbers represent the strongest performance in a given classification system. Two metrics are used for evaluation. Following the classic literature, accuracy (*acc*) is defined as a weighted arithmetic mean of Precision (fraction of the true positives among the total positives) and Inverse Precision. Given the imbalanced data set, balanced accuracy (*bal*) was also computed, defined as the average of Recall (fraction of the predictions that are successfully predicted) obtained on each class.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Linear Regression | | | | Random Forests | | | |
| LR | | BalancedLR | | RF | | BalnacedRF | |
| Features Processed | | acc | bal | acc | bal | acc | bal | acc | bal |
|  | # Features |  |  |  |  |  |  |  |  |
| ALL (baseline) |  | 0.9771 | 0.5001 | 0.4659 | 0.4970 | 0.9777 | 0.5004 | 0.9772 | 0.5059 |
| PCA | 2 | 0.9770 | 0.5023 | **0.7779** | **0.9763** | 0.9763 | 0.5032 | 0.7264 | 0.6754 |
| LR\* | 268 | 0.9774 | 0.4998 | 0.5366 | 0.9776 | 0.9776 | 0.5000 | 0.4779 | 0.4979 |
| LSVM\* | 151 | 0.9774 | 0.4998 | 0.4966 | 0.9777 | 0.9777 | 0.5000 | 0.4987 | 0.5156 |
| RF | 221 | 0.9728 | 0.5039 | 0.6682 | 0.9622 | 0.9622 | 0.5017 | **0.7212** | **0.7535** |
| XT | 280 | 0.9776 | 0.5000 | 0.7573 | 0.9765 | 0.9765 | 0.5033 | **0.7212** | **0.7535** |

Table 3. Classification Performance of Chosen Predictors (cont’d)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Bagging | | | | Boosting | | | |
| Bagging | | BalancedBagging | | AdaBoost | | BalancedAdaBoost | |
| Features Processed | | acc | bal | acc | bal | acc | bal | acc | Bal |
|  | # Features |  |  |  |  |  |  |  |  |
| ALL (baseline) |  | 0.8241 | 0.5059 | **0.8241** | **0.7036** | 0.9775 | 0.5266 | **0.7533** | **0.7746** |
| PCA | 2 | 0.7856 | 0.5030 | 0.7856 | 0.6747 | 0.9777 | 0.5000 | 0.7167 | 0.7361 |
| LR\* | 268 | 0.8030 | 0.5007 | 0.8030 | 0.5124 | 0.9777 | 0.5000 | 0.4708 | 0.5144 |
| LSVM\* | 151 | 0.8039 | 0.4998 | 0.8039 | 0.5164 | 0.9777 | 0.5000 | 0.4869 | 0.5060 |
| RF | 221 | 0.8212 | 0.5047 | 0.8212 | 0.7017 | 0.9743 | 0.5264 | 0.7465 | 0.7647 |
| XT | 280 | 0.8212 | 0.5047 | 0.8212 | 0.7017 | 0.9743 | 0.5264 | 0.7465 | 0.7647 |

Table 3. Predictive Performance of Chosen Predictors

\* Note: LR and LSVM used an expanded set of 774 encoded features while the other models used the original set of 664 features.

# 5. Discussion

The effects of feature selection are heterogeneous depending on the predictive performance of the classification algorithms. Overall, the tree-based selection performs better than the *L*1-based selection, both in the cost (the final number of selected features) and the outcome (evaluation metrics). With a smaller set, the tree-based filtering achieves better balanced accuracy. Especially, although random forests models do not consider constitution effects among the features when pruning the trees, the selection was successful in better prediction using the same scheme.

It is noted that using all the features sometime performs better than the selected subset in bagging and adapted boosting classifiers. This is partly because the baseline features are already disjoint, and substitution effects in the presence of correlated features were weak. Further, the contrast between linear and tree-based model does not only demonstrate the difference in the degree of conservativeness (to which the variables are shrunk to zero) but also illustrate the potential impact of mathematical properties of variables on selection results.

The unsupervised training of the principal component analysis does not directly outperform the other feature selection mechanisms, as it does not rely on the similarity ground truth data. However, the dimensionality reduction using PCA have been shown to help in other tasks by transforming the feature space in a way that makes machine learning easier. The transformations change the space of functions that can be modeled by parameterizing simple models, e.g. by finding the maximal variance when projecting the vectors. A transformation into a more suitable representation, determined by unsupervised training, can thus lead to better adaptation of the prediction model, given the model is relatively simple. However, the PCA results appear to have contrasting impacts on tree-based models. The dimensionality reduction can reduce noise by throwing away less useful components, but, the construction of a composite feature space does not necessarily elevate the performance, i.e., making the predictor set too simple.

# 6. Conclusion

The study makes several contributions. First, the application of machine learning approaches to feature processing where very sparse. Second, show potential applicability of statistical techniques in rich financial data where interpolation is not a feasible solution. Third, the models proposed from grid search prove to be conducive to better fitting of optimization parameters, which especially when combined with consistent models features are most useful.

Future research is encouraged for different shapes of data. While traditional financial data sets are mostly structured, recent advancements in algorithms have shown that unstructured data can effectively identify previously unknown patterns and relationships. A few examples financial institutions can employ include speech-to-text (STT) data gleaned from customer calls, customer location information (GIS) tracked from mobile apps as well as off-line purchases, and 2D image data generated from customers’ social media accounts. The increasing heterogeneity of input data will pose a challenge in computation and model interpretation. Further studies on constructing and evaluating features would help efficient discovery of knowledge without bold assumptions on variable distributions.

# REFERENCES

Ban, G.-Y., and Rudin, C. 2018. “The Big Data Newsvendor: Practical Insights from Machine Learning,” *Operations Research* (67:1), pp.

Hinton, G. E., Osindero, S., and Teh, Y.-W. 2006. “A Fast Learning Algorithm for Deep Belief Net,” *Neural Computation* (18), pp. 1527–1554.

Hinton, Geoffrey E. 2002. “Training Products of Experts by Minimizing Contrastive Divergence,” *Neural Computing*, 14(8), pp. 1771–1800.

Jain, A. and Zongker, D. (1998) “Feature Selection: Evaluation, Application, and Small Sample Performance,” IEEE Transactions on Pattern Analysis and Machine Inelligence (19:2), pp. 153-158.

King, G. and Zeng, L. (2001) “Logistic Regression in Rare Events Data,” Working paper, Harvard University.

Kearns, M. and Valiant, L. (1989) “Cryptographic Limitations on Learning Boolean Formulas and finite Automata,” in Proceedings of the 21st Annual ACM Symposium on Theory of Computing, pp. 433-444. New York.

López de Prado, M. (2018) *Advances in Financial Machine Learning*, Wiley.

Ranzato, M., Boureau, Y., and LeCun, Y. 2007. “Sparse Feature Learning for Deep Belief Networks,” in *Proceedings of Neural Information Processing Systems*, pp. 1-8.

Schapire, R. (1990) “The Strength of Weak Learnability,” *Machine Learning* (5:2), pp. 197-227.

Tibshirani, R. 1996. “Regression Shrinkage and Selection via the Lasso,” *Journal of the Royal Statistical Society: Series* *B* (58:1), pp. 267-288.

Zou, H., and Hastie, T. (2005) “Regularization and Variable Selection via the Elastic Net,” *Journal of the Royal Statistical Society: Series* *B* (67:2), pp. 301-320.