Feature Generation Using Machine Learning from Large Spare Financial Data

Emergent Research Forum

Minjung Choi
Shinhan Digital Innovation Institute (SDII)
Shinhan Financial Group
mchoi@shinhan.com

Abstract

The modeling of customer features has become a core component in modern financial analytics. There are several difficulties in adopting conventional machine learning (ML) methodologies to finance domain: distributional asymmetry in the observations, class imbalance in the training labels, and data sparsity resulting from infrequent occurrence. In this study, we try to address the statistical challenges of financial data. Then, 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.

Keywords

Feature generation, big data, machine learning, financial data analytics

Introduction

The modeling of customer features has become a core component in financial analytics. For example, in credit card business, 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 ML methodologies directly to financial data due to its unique characteristics. First, the usual set of predictors X, such as earnings or income, shows non-normality. To resolve skewness and asymmetrical distribution, the empirical econometrics studies have resorted to complex mathematics and specific parametric assumptions. Granted, the extrapolation 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: 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, 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, or sequential versus stable, constitutes a significant barrier in the application of machine learning. For example, some 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 sparsity; otherwise (local) optimization may not be achieved properly.

In this study, we try to address the above difficulties in adopting ML methodologies to financial data sets. We test feature processing using multiple 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.

Related Work

The traditional 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 (Jain and Zongker 1997). This procedure can not only reduce the cost by decreasing the number of features that need to be collected, but also provide better classification accuracy. For the first purpose, linear models such as support vector machines (SVM) have been explored in the statistics literature. However, the drawback of using support vectors is its limited application to big data. Later on, the literature has focused on the induction algorithms that scale well to find, among a large number of correlated variables, comparably irrelevant ones to be eliminated. With the introduction of the Lasso (Tibshirani 1996) regularization, sparse modeling in the high-dimensional predictor case with good performance has proved to be possible. In the subsequent research, Elastic Net (Zou and Hastie 2005) which combines L_1- and L_2-penalties has shown even better performance.[1]

While statistical models have focused on finding relevant features, tree-based methods were developed to increase the probability for better prediction in the ML community. On top of decision trees, some ensemble suggestions include bagging, booting, extra trees, and random forests. Bootstrap aggregation or bootstrap averaging (bagging) is known to be 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. Third, the ensemble forecast is the simple average of the individual forecasts from the *N* models. Later, Schapire (1990) demonstrated that it is possible to combine such weak estimators to achieve one with high accuracy using the procedure we now call boosting. The main distinction from bagging is that it uses sequential fitting. Individual classifiers are fit sequentially, and the poor-performing classifiers are dismissed. The main advantage of the boosting algorithm is that it reduces both variance and bias in forecasts. Subsequently, adaptive boosting (AdaBoost) (Freund et al. 1997) and EasyEnsemble (Liu et al. 2009) were popularized as an ensemble of learners trained on different balanced bootstrap samples.

As an enhancement of bagging, new tree induction algorithms were introduced. Random forests that significantly randomize the standard tree growing algorithm are a well-known example (Bremian 2001). In a similar fashion, Geurts et al. (2006) suggest extremely randomized tress (extra-trees) that select split, both attributes and cut-points, totally or partially at random. It is reported to show better performance than bagging and random forests, but the generalizability to finance domains should be checked.

When fitting decision trees, a rotation of the feature space in a direction that aligns with the axes typically reduces the number of levels needed by the tree. Principle component analysis (PCA) is one way to transform the features to fit ML algorithms. Thus, we extend the concept of feature selection to the tasks of feature generation to address the needs of transforming raw data into a more adequate fitting space. We use unsupervised training with principal component analysis (PCA) to transform the feature space. Then, we use multiple known learning models to extract features from large sparse data, including both linear and tree-based. We compare their performance in the prediction of the targets using the real-world financial customer data.

[1] Simply put, Lasso uses $\text{cost} = \sum e_i^2 + \lambda \sum |w_i|$, which incorporates an additional L_1-based constraint to the least squares regression. Elastic Net uses $\text{cost} = \sum e_i^2 + \lambda_1 \sum |w_i| + \lambda_2 \sum w_i^2$, which is regularized through the sum of second (L_1) and third (L_2) terms. Given the formulae, the coefficients that are judged to be irrelevant can be shrunk or set to zero in Lasso, as its original name "absolute shrinkage" indicates, while adjacent coefficients still can survive in Elastic Net depending on the proportions of λ_1 and λ_2. For a complete discussion, refer to Ridge regression (or Tikhonov regression) as well.

Methods

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 employ logistic regression (LR), linear support vector machines (LSVM) regularized using the L_1-penalty, random forests (RF), and extra tress (XT) classifiers regularized via maximum features and depths. L_1-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.

In addition to feature subset selection, dimensionality reduction was adopted using principal components analysis (PCA), compressing the features to a 2-dimensional vector. Principal components provide best linear approximations of the data that are mutually uncorrelated and ordered in variance. The validation models used such an input and were compared as well.

Resampling

Additionally, sample weights are considered in the processing of predictor variables to resolve the class imbalance problem. Class weights are the subsampling weights that correct for underrepresented labels. This is particularly critical where the most importance classes have rare occurrence such as our case. We tried and compared random undersampling, random oversampling, Smote (Chawla et al. 2002), Tomek links (Tomek 1976), and the combinations of both. The attempted algorithms demonstrated that random undersampling shows the most reliable results. Therefore, the validation involved incorporating additional balanced models with random undersampling. In the case of bagging, out-of-bag (oob) subsamples were used.

Experiments

Data

The experiment uses customer profile and consumption records from a learning credit card company 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). Missing values were either encoded as a separate class or substituted by column mean. After preprocessing, a total of 774 features were produced. 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 50, 25, 25%.

Results

The hyperparameters for the feature selection 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

different algorithms. The feature importance score was calculated based on the 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. While the linear model group and tree-based group respectively seem to produce similar results therein, they show a striking difference against each other.

	LR (L_1)	LSVM (L_1)	XT	RF
LR (L_1)	1.000	0.986	0.281	0.225
LSVM (L_1)		1.000	0.156	0.116
XT			1.000	0.873
RF				1.000

Table 1. Cross-correlations of Feature Importance

For validation, four classification models are used: logistic regression (LR), random forests (RF), bagging (Bag), and adaptive boosting (AdaB) classifiers. Analogous to bagging, random forests also average bootstrapped results, known to be more reliable and less likely to end up overfitting. Thus, both bagging and random forests are investigated for comparison. For each classifier, grid search was conducted to find a potential optimal set of tuning parameters. (Among others, the support vector classifier was removed from the final model candidates due to its limitation in computational scalability, and neural network models were also excluded because of the inability to handle mixed types of data.)

Two metrics are used to evaluate each subset of predictors' performance. First, accuracy (*acc*) is defined as a weighted arithmetic mean of Precision (fraction of the true positives among the total positives) and Inverse Precision. However, the measure can be easily inflated when the presence of one class is dominant over the other. Therefore, balanced accuracy (*bal*) was also computed, defined as the average of Recall (fraction of the predictions that are successfully predicted) obtained on each class. 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.

		Logistic Regression				Random Forests			
		LR *		Balanced + LR *		RF		Balanced + RF	
Feature Processing		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* (L_1)	268	0.9774	0.4998	0.5366	0.9776	0.9776	0.5000	0.4779	0.4979
LSVM* (L_1)	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**
		Bagging				Boosting			
		Bag		Balanced + Bag		AdaB		Balanced + AdaB	
Feature Processing		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* (L_1)	268	0.8030	0.5007	0.8030	0.5124	0.9777	0.5000	0.4708	0.5144
LSVM* (L_1)	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. Classification Performance of Predictor Subsets

* Note: Linear models such as LR and LSVM used one-hot encoded (expanded) variables while the others used original variables.

Discussion & Conclusion

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).

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 RF- and XT-based subsets tend to produce analogous results. In reference to the confusion matrix, they have the same prediction outcomes in this experiment. We suspect two reasons behind the results. First, both random forests and extra trees produce randomized decision trees to circumvent overfitting. The algorithms average the values across all estimators and rank the features accordingly. Thus, their ranked classes may have been the same, especially given the imbalance of the data set.

We could find that the use of ensemble learners, the combination of preferably independent weak learners improve accuracy. López de Prado (2018) argues that in financial applications, bagging is preferable to boosting because bagging addresses overfitting while boosting addresses underfitting. As indicated by our experiment, bagging outperforms both in the original and balanced data set. In the case of financial data where signal-to-noise ratio is low, it is important to grapple with overfitting. It is also noted that bagging is a superior choice to support vector machines because it is scalable. In our experiment, we could find that the SVM algorithm consumed a sizable amount of time and resource until it converges, or did not even converge due to the sparsity in the data. Furthermore, once it has been converged, there is no guarantee that the solution is a global optimum.

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. 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

Breiman, L. 2001. "Random Forests," *Machine Learning* (45), pp. 5-32.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002) "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research* (16), pp. 321-357.

Freund, Y., and Schapire, R. E. 1997. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences* (55:1), pp. 119-139.

Geurts, P., Ernst, D., and Wehenkel, L. 2006. "Extremely Randomized Trees," *Machine Learning* (63:1), pp. 3-42.

Jain, A., and Zongker, D. 1997. "Feature Selection: Evaluation, Application, and Small Sample Performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (19:2), pp. 153-158.

King, G., and Zeng, L. 2001. "Logistic Regression in Rare Events Data," *Political Analysis* (9:2), pp. 137-163.

Liu, X. Y., Wu, J., and Zhou, Z. H. 2009. "Exploratory Undersampling for Class-Imbalance Learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* (39:2), pp. 539-550.

López de Prado, M. 2018. *Advances in Financial Machine Learning*, Hoboken, NJ: Wiley.

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.

Tomek, I. 1976. "Two Modifications of CNN," *IEEE Transactions on Systems, Man, and Cybernetics* (6:11), pp. 769-772.

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.