Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/32/20230188

Research on feature coding theory and typical application analysis in machine learning algorithms

Pengxiang Wang [1], Kailiang Xiao[2,4] and Lihao Zhou [3]

[1]JSNU-SPbPU Institute of Engineering, Jiangsu Normal University, Xuzhou, 221116, China
[2]School of Electronic and Information Engineering, South China University of Technology, Guangzhou, 510641, China
[3]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[4]202030242188@mail.scut.edu.cn

Abstract. Nowadays, the world is still in the environment of economic depression. In order to promote economic recovery, improve Relations of production and production efficiency, stimulate consumption expansion and upgrading, and accelerate industrial transformation and upgrading, problems such as industrial upgrading need to be solved urgently. Solving the above problems requires more useful tools, and artificial intelligence is one of them. Machine learning is the key to distinguishing artificial intelligence from ordinary program code. Unlike people learning knowledge, machine learning has its own unique language algorithms and behavioral logic. Machine learning, as a technology active in the field of artificial intelligence in recent years, specializes in studying how computers learn, simulate and realize part of human learning behavior, so as to provide data mining and behavior prediction for humans, to obtain new knowledge or skills, or to strengthen the original basic ability of machines. In this study, a variety of common coding algorithms and learning strategies in machine learning are discussed, supervised learning algorithms are selected as examples in the learning strategies, models are further selected and evaluated for a variety of algorithms, and parameters are adjusted and performance is analyzed. As for the theoretical analysis in the research, the paper makes a tentative application in the three fields of housing price, physical store sales and digital recognition, explores and selects the corresponding application method in the appropriate scenario, and expands the application field of machine learning.

Keywords: artificial intelligence, machine learning, feature encoding.

1. Introduction

Machine learning, is specialized in learning how computers learn, simulate, and use human behavior, in order to provide data mining and human behavior to obtain new knowledge or skills. Machine learning, as the basis of artificial intelligence, is the way to make a computer intelligent.

The strategy of machine learning is the inference strategy adopted by the system during the learning process, which involves selecting appropriate machine learning algorithms throughout the entire process. The algorithm mainly includes Supervised learning and Unsupervised learning; The latter discovers and

DOI: 10.54254/2755-2721/32/20230188

summarizes the relationship between input and output based on unlabeled datasets, and makes predictions for new inputs [1].

Focusing on supervised learning, this study conducted in-depth analysis and exploration of linear models under supervised learning, compared the advantages and disadvantages of different supervised learning algorithms, trained, selected and evaluated the models through cross-validation, and analyzed the applied data and parameters through different scenario application and model practice. In order to extend its application range.

2. Related theory

2.1. Seven common coding algorithms in machine learning

2.1.1. Label coding. Coding labels simply means giving different categories and different numerical labels. It belongs to hard coding, that is, it directly maps a large number of category features, and how many category values represent how many. This hard coding method is simple and rude, convenient and fast. In the encoding, sklearn. Preprocessing. labelencoder is used to encode the target label, and its value is between 0 and n classess-1.

2.1.2. Serial number coding. For ordinal variables, assign values from 1 to n to this n-class ordinal variable in sequence. But in fact, the method of characterizing interval variables is applied to ordinal variables, which is similar to label coding, but the categories are coded according to the number order specified in advance. Sklearn. preprocessing.ordinal encoder can encode in two ways: first, load the data categorical_df into the encoder, and then encode the loaded data; second, directly load the encoder and encode it [2].

2.1.3. Binary coding. Binary coding usually refers to linear block coding, a [n, k] linear Block code, eighteen pieces of data are divided into k characters as a section (called group data), and through the encoder, it becomes a group of n long characters, according to the codeword of [n, k] linear Block code.

2.1.4. Frequency coding. Replace the category features with the counts in the training set (generally counting is based on the training set, which belongs to a kind of statistical coding. Statistical coding is to replace the original category with the statistical features of the category, for example, the code is 100 when the category A appears in the training set for 100 times). This method is very sensitive to outliers, so the results can be normalized or transformed (for example, using logarithmic transformation). Unknown categories can be replaced with 1. There is no special encoder for frequency coding, but it can be realized by CountEncoder in the categorical_encodings package.

2.1.5. Mean coding. In machine learning and data mining, whether it is a classification problem or a regression problem, the collected database often includes categorical feature. Because categorical feature indicate that some data belongs to a specific category, numerically, categorical features are usually discrete integers from 0 to n. Generally speaking, for categorical feature, we only need to use tag coding and one-shot coding mentioned in 2.1.1 and 2.1.2. The former can receive irregular feature columns and convert them into integer values from 0 to n-1 (n is n different categories); The latter can make a sparse matrix of m*n through one-hot coding (assuming that the data has a total of m rows, whether the specific output matrix format can be controlled by sparse parameters) [3].

2.1.6. Helmert coding. Helmert coding is usually used in econometrics. After Helmert coding (each value in the classification feature corresponds to a line in the Helmert matrix), the coded variable coefficient in the linear model will show that the difference between the average value of the dependent variable given a certain value of the category variable and the average value of the dependent variable

DOI: 10.54254/2755-2721/32/20230188

given other values of the category. Helmert coding used in category_encoders package is reverse Helmert coding.

2.2. Supervised learning

2.2.1. Linear model return. Minimizing the sum of squares of a residual with penalty by the coefficient of ridge regression:

$$min_{\omega}||X_{\omega} - y||_2^2+\alpha||\omega||_2^2 \tag{1}$$

Ridge regression has changes in the classifier. RidgeClassified, this classifier is sometimes called a vector machine supporting least squares with a linear core. Similar cross validation scores can be achieved by combining recall, accuracy, or accuracy/recall. RidgeClassifier employs different penalty least squares loss methods to provide different digital solvers with different computational performance summaries, based on their respective numerical performance. The logistic function is used to describe the probability of the output results of a single experiment. Logistic regression is realized in LogisticRegression, where binary, one-to-many classification (One-vs-Rest) and polynomial logistic regression are realized, and the regularization options are ℓ1、ℓ2or elastic net. An IsotonicRegression class that fits a non-decreasing function of real numbers to one-dimensional data.

$$\text{minimise} \sum_{i} \omega_i (y_i - \hat{y}_i)^2 \, subject \, \hat{y}_i \le \hat{y}_j \, whenever X_i \le X_j \tag{2}$$

Among them, the weight is strictly positive, and both x and y are arbitrary real numbers. The parameter increasing can change the constraint to even if. Setting it to' auto' will automatically select the constraint according to Spearman's rank correlation coefficient [4]. In terms of mean square error, IsotonicRegression produces a series of predictions for training data, which is the closest to the target. These predictions are interpolated to predict unknown data. Naive Bayes. Given the importance of categorical variables, the independence of each pair of traits is assumed to be biased. Bayes' theorem gives the relationship between class variables and their associated eigenvectors:

$$\mathrm{P}(y \mid x_1,..., x_n) = \frac{P(y)P(x_1,..., x_n \mid y)}{P(x_1,..., x_n)} \tag{3}$$

Conditional independence assumption using Naive Bayes:

$$\mathrm{P}\left(x_i \mid y, x_1,..., x_{i-1}, x_{i+1},..., x_n\right) = P(x_i \mid y) \tag{4}$$

For all, this relationship can be simplified as:

$$\mathrm{P}\left(y \mid x_1,..., x_n\right) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1,..., x_n)} \tag{5}$$

Because it is an input constant:

$$\mathrm{P}\left(y \mid x_1,..., x_n\right) \propto P(y)\prod_{i=1}^{n} P(x_i \mid y) \Rightarrow \hat{y} = \arg\max_{y} P(y)\prod_{i=1}^{n} P(x_i \mid y) \tag{6}$$

Quantities can be assessed using the method of maximum a posteriori prediction (MAP), which correlates with the classes observed during the training phase. Although this method may appear simplistic, don't be misled by its seeming simplicity. Indeed, it requires minimal information to predict necessary parameters accurately [5]. Moreover, the decoupling of conditional feature distribution classes

DOI: 10.54254/2755-2721/32/20230188

facilitates the independent estimation of each distribution as a single entity. This process can effectively mitigate the dilemmas presented by the curse of dimensionality. However, while Naive Bayes is generally recognized as a proficient classifier, it falls short as a predictor. Consequently, the results generated by the 'proba' prediction should be taken with a grain of salt.

The Decision Tree (DT) is a non-parametric technique utilized for both classification and regression in supervised learning. Some of the key advantages of the decision tree include its interpretability and simplicity. Decision trees can be visually inspected, offering an intuitive understanding of the model. Furthermore, decision trees usually require little to no data preprocessing, unlike other algorithms which often necessitate data normalization or scaling. Decision trees also have the capability to handle both numerical and categorical data, and are capable of solving multi-output problems. Despite these advantages, decision trees also have their share of drawbacks. To circumvent some of these issues, strategies such as tree pruning or setting a minimum number of samples required at a leaf node can be implemented.

2.3. Model selection and evaluation

2.3.1. Feature selection and extraction. Training and testing the parameters of the predictor function on the same data set is a bad approach. It is important to note that "experimentation" is not only used in academia, as in commercial spheres, machine learning usually begins with experiments. Scikit learn Test can be used the split auxiliary function to randomly split dataset into the training set and test set [6]. If the evaluator evaluates various parameters, if parameter C is the super parameter of the support vector machine, the selection of the parameter determines the optimal performance of the model, and the risk of fitting remains before fitting the model. The above problem can be resolved by splitting several known data sets into a "validation set". Training the model with training data to evaluate the model in the validation set. If the "test" score is good, you can finally evaluate the model on the test set.

Dividing the dataset into three or more files reduces the amount of data available for modeling. Therefore, the results of model evaluation depend on the random distribution of training sets and validation sets. One way to solve the above problems is corss-validation (CV). When the cross-validation method is applied. The most basic cross-validation method is k-fold CV, which refers to dividing the training set into k smallest subsets (other methods will be introduced below, and all have the same principle). The following procedure applies one of the k "folds": K-1 subsets are used for model training; *Cross-validation iterators for independent co-distributed data.* It is assumed that some data are independent and identically distributed, that all samples come from the same generation process, and that the generation process is not based on the memory of past samples.

2.3.2. Model persistence, security and maintainability restrictions. Reuse without reconfiguring the model. Examples will be provided later to explain how to use pickle to start the model to be more persistent. When using pickle serialization, some security and maintainability issues need to be reviewed.

Another option for pickle is to output the model in another form. See Related Projects for details by using the model output tool. Unlike pickle, once output, you can't restore the complete Scikit-learn estimator object, but you can use models to make predictions, usually using tools that support open model exchange formats Pickle (and through extended joblib), there are some examples about maintainability and security [7]. Because, Do not use untrusted data that has not been pickle, because it may lead to the execution of malicious files during operation. When the model is loaded in another version, it is totally unsupported and not recommended. It should always be remembered that performing operations on such data may result in different and unexpected results.

2.3.3. Model evaluation. Bias and variance are characteristics of estimators, and learning algorithms and hyperparameters are often chosen to reduce them. Way to cut the standard deviation is to use more informative data. However, if the Performance of the real difference is too difficult to estimate the low difference index, then only more informative data can be collected. It is difficult to visualize models in

DOI: 10.54254/2755-2721/32/20230188

high-dimensional space. Therefore, using tools for description is very useful. In order to verify the accuracy of models such as classifiers, it is necessary to evaluate the function. It should be noted that when optimizing hyperparameters based on validation scores, the validation scores are biased, which is not a good generalization estimate. To get a good estimate of generalizability, you can use a different set of tests to calculate the score. The estimation fails if the training and validation results are low. Lower training scores and higher validation scores are generally not possible. The learning curve shows the results of the validation set and the estimator training set with different numbers of training samples. For naive Bayes, with the increase of the training set size, the score decreases very low. Because of this, it will not benefit a lot from a larger data set. In contrast, the training score of support vector machine is much higher than the verification score for small data.

3. Application

3.1. House price

3.1.1. Project background. House price prediction is a classic machine learning problem and a common item in data analysis. Kaggle provides a concrete example: according to 79 features given, the corresponding house price is predicted, including the type of house, the width of street, the area of each floor, etc., and the evaluation index given is the common mean square error (RMSE) of C in regression problems:

$$RMSE = \sqrt{\frac{1}{\mathrm{n}}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{7}$$

3.1.2. Data processing. Looking at the data in the first five lines, we will find that the' ID' in line 0 cannot participate in training. In the data set, we have a total of 79 related features, which are all kinds of data types, including numerical features and classification features. The size and scale of numerical features are different, and some features also contain missing values. Next, the classification features are processed and replaced by one-hot coding. One-hot coding is to transform different categories into unique features [8]. For example, "MSZoning" contains the values "RL" and "Rm". Finally, the missing value of the data feature is added to 0. For the category features with obvious sequential relationship, such as garage quality' GarageQual ', the bigger the better, LabelEncode is used, and for those without sequential relationship, single heat encoding is used. At the same time, we can artificially add new features.

3.1.3. Training. After data initialization, we can start training. Here we first construct a simplest linear model with MLP. Obviously, the linear model is difficult for us to win the competition, but the linear model provides a reliability check to see if there is meaningful information in the data. If all goes well, the linear model will be used as a base model so that we can intuitively know how much the best model outperforms the simple model.

Firstly, we define our loss function. For house prices, we're interested in the relative price fluctuation, so we're more interested in the relative error (y-y')/y than the absolute error y-y'. The solution is very simple. Just take the logarithm of y, and the division of relative error will be converted into the subtraction of ordinary loss function. The root mean square error can be obtained as follows:

$$\sqrt{\frac{1}{\mathrm{n}}\sum_{i=1}^{n}(\log y_i - \log \hat{y}_i)^2} \tag{8}$$

3.1.4. K-fold cross training and model parameter adjustment. K-fold cross-validation means that the training data set is divided into training data and validation data. That is, the training data set is divided

DOI: 10.54254/2755-2721/32/20230188

into k folds, one of which is taken as verification data, and the others are used as training data for training [9].

Generally speaking, when training a model, it is not that the smaller the training loss, the better. When the model is large enough, the training is easy to over-fit, and the noise of the training data is all learned, and the generalization ability of the model is very poor at this time. As shown in the figure, we intuitively describe the relationship between the model under-fitting and over-fitting. Sometimes the error of K-fold cross-validation is much higher, which indicates that the model is over-fitted (as shown in the previous figure). In the whole training process, we hope to monitor the training error and verification error. Less over-fitting may indicate that the existing data can support a stronger model, while greater over-fitting may mean that we can benefit from regularization technology.

3.2. Store sales - time series forecasting

In order to verify the accuracy of models such as classifiers, it is necessary to evaluate the function. A good method to choose more Rothman of course operates more than 3000 pharmacies in seven European countries. Root mean square percentage error (RMSPE) given in the question:

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{y_i - \hat{y}_i}{y_i})^2} \tag{9}$$

Note that there is a phenomenon in the data of the day that it opened on the same day, but the sales volume is 0, so it will not be scored at this time.

3.2.1. Data description. Store.csv (provides additional details about each store). 'Store': Represents the unique ID of each store, totalling to 1115. 'StoreType': Indicates one of the four different store models: A, B, C, and D. The distribution is as follows: A(54%), D(31%) and others (15%). 'Assortment': Classifies the level of variety available: a = basic, b = extra, c = extended. 'CompetitionOpenSinceYear': Approximate year when the closest competitor store opened. 'Promo2': Indicates ongoing promotional activities (0= stores not participating; 1= stores participating). The data is equally split between the two. 'Promo2SinceWeek': The calendar week when the store began participating in the promotional activities. 'Promo2SinceYear': The year when the store started the promotional activities. 'PromoInterval': Refers to continuous promo2 activities, specifying the months when the promotions are restarted. For example, "February, May, August, November" means each round of promotions starts in February, May, August, and November each year [10].

Train.csv (includes historical sales data). 'Store': The unique ID for each store. 'DayOfWeek': Specifies the day of the week. 'Date': The specific date. 'Sales': The store's turnover for the day. 'Customers': The count of customers on that day (note: this information is not available in the test.csv). 'Open': Indicates whether the store was open that day (0 signifies 'No'; 1 signifies 'Yes'). 'Promo': Shows if a promotion was running in the store that day. 'StateHoliday': Identifies if the day was a state holiday (generally, with a few exceptions). 'a' represents a public holiday, 'b' is for Easter holiday, 'c' denotes Christmas, and '0' implies no holiday.

3.2.2. Data analysis. After processing all the data one by one, it is found that there are three missing values in the CompetitionDistance, and the CompetitionDistance of each store is quite different (the maximum is 20, and the minimum is 75860), so the median is chosen to fill in, and because the data of the CompetitionDistance of the store is inclined, it is transformed logarithmically [2]. There are a lot of data missing in the competition opening month & competition opening eye, so it is difficult to determine the filling rules for the approximate year and month of the establishment of competitive stores, and the impact on sales after making its data visualization is not obvious. Promo2SinceWeek, Promo2SinceYear and PromoInterval all have 544 missing values. Looking at the data distribution of promo2, it is found that 544 stores have no long-term promotion activities, so the above three characteristics are not data

DOI: 10.54254/2755-2721/32/20230188

missing. All discrete features are processed in two parts: First numeritization, followed by one-hot encoding.

4. Conclusion

Optimization of Models: Model optimization is paramount. We've already seen the development of various optimization methods such as Adam, RMSprop, Adagrad, and others. As we move forward, we will undoubtedly see the rise of new methods designed to enhance the performance and efficiency of deep learning models. Neural Network Structure: As deep learning applications continue to expand, the need for more versatile neural network structures grows. To adapt to different types of data and tasks, innovative network structures such as residual networks, attention mechanisms, and convolutional neural networks are being proposed. We anticipate the creation of even more novel network structures in the future. Model Interpretability: The 'black box' nature of machine learning and deep learning models has resulted in a bottleneck when it comes to model interpretation and interpretability. The need to explain the decision-making process of these models necessitates the development of more transparent models and interpretation methods. Furthermore, it's vital to strengthen the standardization and normalization of model interpretation. Model Generalization and Transfer Learning: The generalization and transfer learning capabilities of machine learning and deep learning models are of utmost importance. Generalization refers to a model's ability to perform on unseen data, while transfer learning relates to a model's adaptability to different tasks and scenarios. To improve these capabilities, we need to develop more robust and transferable models and algorithms, as well as reinforce the standardization and normalization of model evaluation and comparison. Computing Resources and Energy Consumption: Machine learning often requires significant computing resources and energy, which poses a challenge in certain application scenarios. Therefore, the development of more efficient models and algorithms is required, along with a focus on collaborative optimization of hardware and software. Predicted Future Developments: Advancements in Automation and Intelligence: As machine learning and deep learning technologies advance, we are likely to see increased automation and intelligence. This could lead to more intelligent products and services, including voice assistants, smart furniture, intelligent transportation systems, and smart healthcare solutions. Emergence of Federated Learning and Edge Computing: Federated learning and edge computing will become increasingly prevalent. Federated learning allows model training across multiple devices without data being sent to a central server, thereby ensuring data privacy. Edge computing shifts computational power closer to the device, reducing reliance on cloud computing, and improving computational efficiency and data privacy.

In conclusion, machine learning will continue to be pivotal in driving innovation and progress in the future. However, continuous technological advancements and the exploration of applications are necessary to overcome current challenges. By enhancing our understanding of machine learning coding theory and learning strategy, we can compare the merits and demerits of traditional coding theory, apply different models to various application scenarios, and expand the potential of machine learning through adjustment of model outputs and parameters. This should lead to more comprehensive and profound applications of artificial intelligence, promising a brighter future for humanity.

Authors contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] Sanni-Anibire M O, Zin R M, Olatunji S O. Developing a preliminary cost estimation model for tall buildings based on machine learning[M]//Big Data and Information Theory. Routledge, 2022: 94-102.

[2] Canese L, Cardarilli G C, Di Nunzio L, et al. Multi-agent reinforcement learning: A review of challenges and applications[J]. Applied Sciences, 2021, 11(11): 4948.

DOI: 10.54254/2755-2721/32/20230188

[3] Sarker I H. Machine learning: Algorithms, real-world applications and research directions[J]. SN computer science, 2021, 2(3): 160.
[4] Li Y. Research and application of deep learning in image recognition[C]//2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA). IEEE, 2022: 994-999.
[5] Sun Q, Ge Z. A survey on deep learning for data-driven soft sensors[J]. IEEE Transactions on Industrial Informatics, 2021, 17(9): 5853-5866.
[6] Wang P, Fan E, Wang P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning[J]. Pattern Recognition Letters, 2021, 141: 61-67.
[7] Ginart A A, Naumov M, Mudigere D, et al. Mixed dimension embeddings with application to memory-efficient recommendation systems[C]//2021 IEEE International Symposium on Information Theory (ISIT). IEEE, 2021: 2786-2791.
[8] Zhang W, Li H, Li Y, et al. Application of deep learning algorithms in geotechnical engineering: a short critical review[J]. Artificial Intelligence Review, 2021: 1-41.
[9] Jia W, Sun M, Lian J, et al. Feature dimensionality reduction: a review[J]. Complex & Intelligent Systems, 2022, 8(3): 2663-2693.
[10] Zhang Y, Shi X, Zhang H, et al. Review on deep learning applications in frequency analysis and control of modern power system[J]. International Journal of Electrical Power & Energy Systems, 2022, 136: 107744.